

TEAS Working Group
Internet Draft
Intended status: Informational
Expires: September 2017

Italo Busi (Ed.)
Huawei
Sergio Belotti (Ed.)
Nokia
Victor Lopez
Oscar Gonzalez de Dios
Telefonica
Anurag Sharma
Infinera
Yan Shi
China Unicom
Ricard Vilalta
CTTC
Karthik Sethuraman
NEC

March 3, 2017

Yang model for requesting Path Computation
draft-busibel-teas-yang-path-computation-02.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on September 3, 2016.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

This document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

This document also proposes a yang model for a stateless RPC which complements the stateful solution defined in [TE-TUNNEL].

Table of Contents

1. Introduction.....	3
2. Use Cases.....	4

2.1. IP-Optical integration.....	5
2.1.1. Inter-layer path computation.....	6
2.1.2. Route Diverse IP Services.....	8
2.2. Multi-domain TE Networks.....	8
2.3. Data center interconnections.....	9
3. Interactions with TE Topology.....	11
3.1. TE Topology Aggregation using the "virtual link model"...	11
3.2. TE Topology Abstraction.....	19
3.3. Complementary use of TE topology and path computation...	20
4. Motivation for a YANG Model.....	22
4.1. Benefits of common data models.....	22
4.2. Benefits of a single interface.....	23
4.3. Extensibility.....	23
5. Path Computation for multiple LSPs.....	24
6. YANG Model for requesting Path Computation.....	25
6.1. Modeling Considerations.....	25
6.1.1. Stateless and Stateful Path Computation.....	25
6.1.2. Reduction of Path Computation Requests.....	25
6.2. YANG model for stateless TE path computation.....	25
6.2.1. YANG Tree.....	25
6.2.2. YANG Module.....	36
7. Security Considerations.....	45
8. IANA Considerations.....	45
9. References.....	45
9.1. Normative References.....	45
9.2. Informative References.....	46
10. Acknowledgments.....	47

1. Introduction

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

When we are thinking to this type of scenarios we have in mind specific level of interfaces on which this request can be applied.

We can reference ABNO Control Interface [RFC7491] in which an Application Service Coordinator can request ABNO controller to take in charge path calculation (see Figure 1 in the RFC) and/or ACTN [ACTN-frame], where controller hierarchy is defined, the need for path computation arises on both interfaces CMI (interface between Customer Network Controller(CNC) and Multi Domain Service

Coordinator (MDSC)) and/or MPI (interface between MSDC-PNC).[ACTN-Info] describes an information model for the Path Computation request.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

Path Computation Elements, Controllers and Orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG Data Model for TE Topologies [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [L1-TOPO] for Layer-1 ODU technologies).

The availability of such topology models allows providing the TED using YANG-based protocols (e.g., NETCONF or RESTCONF). Furthermore, it enables a PCE/Controller performing the necessary abstractions or modifications and offering this customized topology to another PCE/Controller or high level orchestrator.

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via YANG-based protocols (e.g., NETCONF or RESTCONF)using the TE-Tunnel Yang model [TE-TUNNEL].

This document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

This document also proposes a yang model for a stateless RPC which complements the stateful solution defined in [TE-TUNNEL].

2. Use Cases

This section presents different use cases, where an orchestrator needs to request underlying SDN controllers for path computation.

The presented uses cases have been grouped, depending on the different underlying topologies: a) IP-Optical integration; b) Multi-domain Traffic Engineered (TE) Networks; and c) Data center interconnections.

2.1. IP-Optical integration

In these use cases, an Optical domain is used to provide connectivity between IP routers which are connected with the Optical domains using access links (see Figure 1).

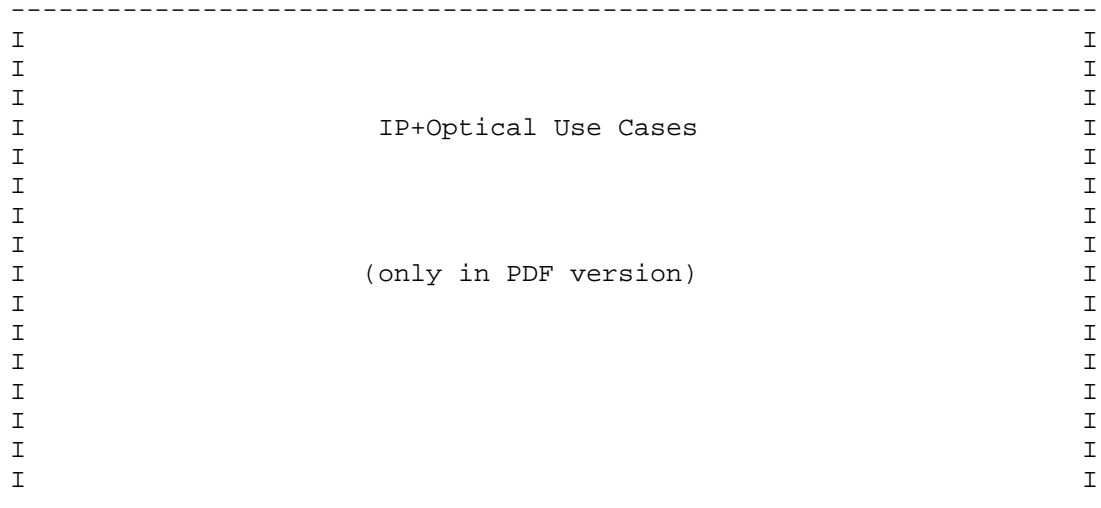


Figure 1 - IP+Optical Use Cases

It is assumed that the Optical domain controller provides to the orchestrator an abstracted view of the Optical network. A possible abstraction shall be representing the optical domain as one "virtual node" with "virtual ports" connected to the access links.

The path computation request helps the orchestrator to know which are the real connections that can be provided at the optical domain.

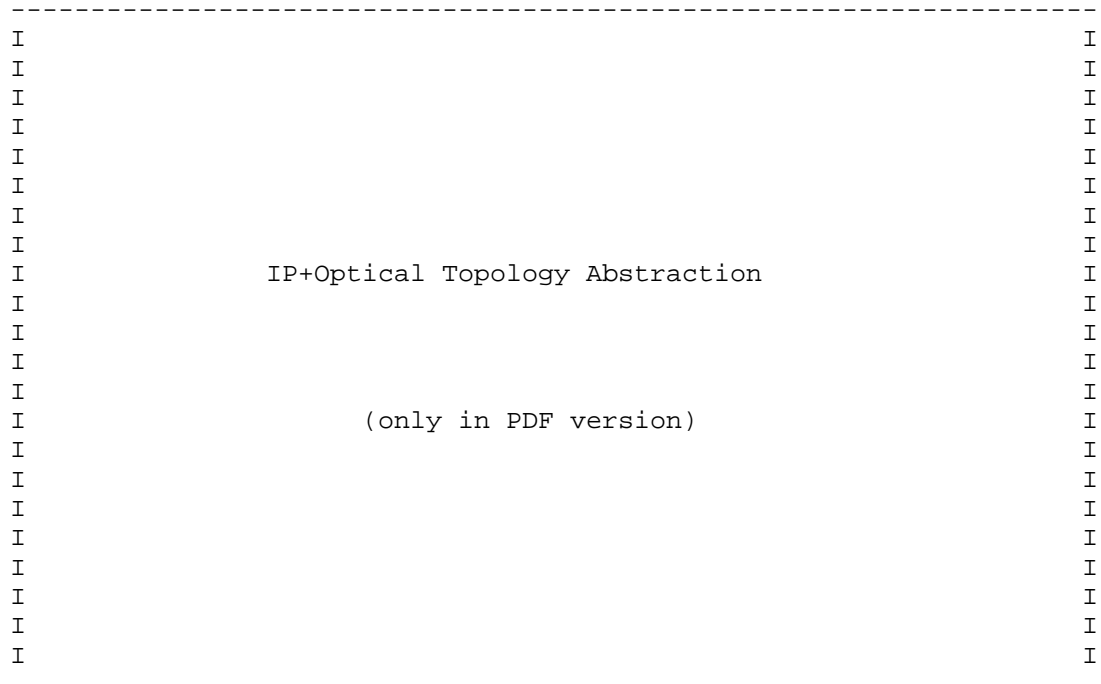


Figure 2 - IP+Optical Topology Abstraction

2.1.1.1. Inter-layer path computation

In this use case, the orchestrator needs to setup an optimal path between two IP routers R1 and R2.

As depicted in Figure 2, the Orchestrator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The orchestrator can request the underlying Optical domain controller to compute a set of potential optimal paths, taking into account optical constraints. Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose

which one of these potential paths to use to setup the optimal e2e path crossing optical network.

```

-----
I                                     I
I           IP+Optical Path Computation Example           I
I                                                         I
I                                                         I
I           (only in PDF version)                         I
I                                                         I
I                                                         I
-----

```

Figure 3 - IP+Optical Path Computation Example

For example, in Figure 3, the Orchestrator can request the Optical domain controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path using the VP2-VP5 Optical path even this is not the optimal path from the Optical domain perspective.

Considering the dynamicity of the connectivity constraints of an Optical domain, it is possible that a path computed by the Optical domain controller when requested by the Orchestrator is no longer valid when the Orchestrator requests it to be setup up.

It is worth noting that with the approach proposed in this document, the likelihood for this issue to happen can be quite small since the time window between the path computation request and the path setup request should be quite short (especially if compared with the time that would be needed to update the information of a very detailed abstract connectivity matrix).

If this risk is still not acceptable, the Orchestrator may also optionally request the Optical domain controller not only to compute the path but also to keep track of its resources (e.g., these resources can be reserved to avoid being used by any other connection). In this case, some mechanism (e.g., a timeout) needs to be defined to avoid having stranded resources within the Optical domain.

These issues and solutions can be fine-tuned during the design of the YANG model for requesting Path Computation.

2.1.2. Route Diverse IP Services

This is for further study.

2.2. Multi-domain TE Networks

In this use case there are two TE domains which are interconnected together by multiple inter-domains links.

A possible example could be a multi-domain optical network.

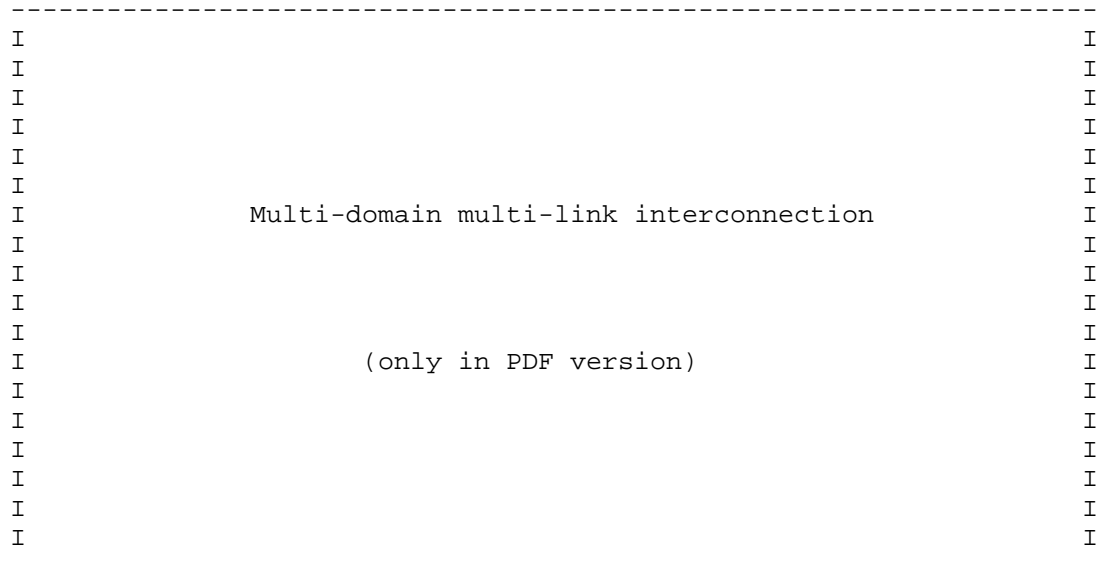


Figure 4 - Multi-domain multi-link interconnection

In order to setup an end-to-end multi-domain TEpath (e.g., between nodes A and H), the orchestrator needs to know the feasibility or the cost of the possible TE paths within the two TE domains, which depend from the current status of the physical resources within each TE network. This is more challenging in case of optical networks because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

In order to setup a multi-domain TE path (e.g., between nodes A and H), Orchestrator can request the TE domain controllers to compute a set of intra-domain optimal paths and take decisions based on the information received. For example:

- o The Orchestrator asks TE domain controllers to provide set of paths between A-C, A-D, E-H and F-H
- o TE domain controllers return a set of feasible paths with the associated costs: the path A-C is not part of this set (in optical networks, it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller)
- o The Orchestrator will select the path A- D-F- H since it is the only feasible multi-domain path and then request the TE domain controllers to setup the A-D and F-H intra-domain paths
- o If there are multiple feasible paths, the Orchestrator can select the optimal path knowing the cost of the intra-domain paths (provided by the TE domain controllers) and the cost of the inter-domain links (known by the Orchestrator)

This approach may have some scalability issues when the number of TE domains is quite big (e.g. 20).

In this case, it would be worthwhile using the abstract TE topology information provided by the domain controllers to limit the number of potential optimal end-to-end paths and then request path computation to fewer domain controllers in order to decide what the optimal path within this limited set is.

For more details, see section 3.3.

2.3. Data center interconnections

In these use case, there is an TE domain which is used to provide connectivity between data centers which are connected with the TE domain using access links.

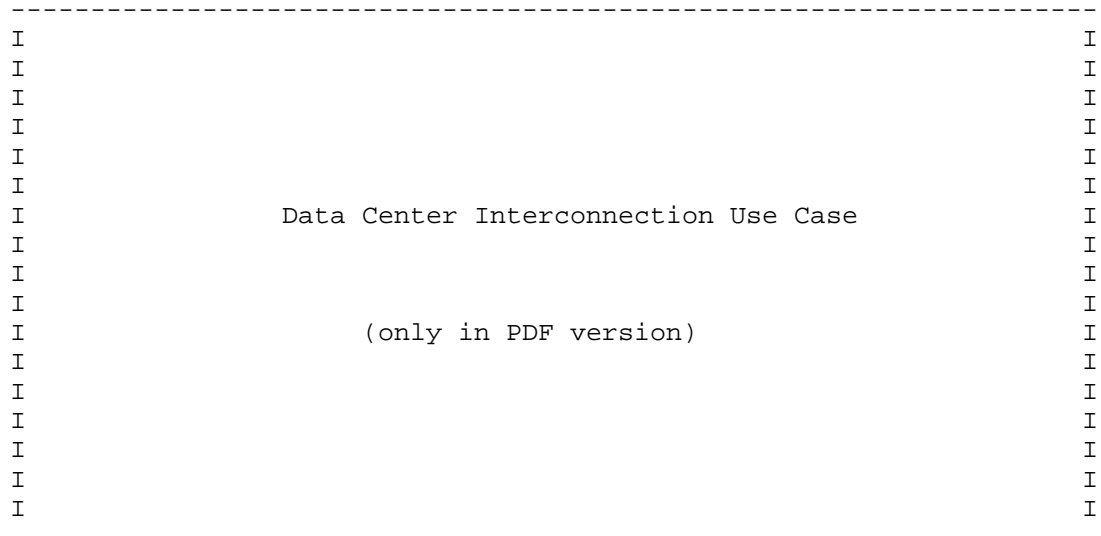


Figure 5 - Data Center Interconnection Use Case

In this use case, a virtual machine within Data Center 1 (DC1) needs to transfer data to another virtual machine that can reside either in DC2 or in DC3.

The optimal decision depends both on the cost of the TE path (DC1-DC2 or DC1-DC3) and of the computing power (data center resources) within DC2 or DC3.

The Cloud Orchestrator may not be able to make this decision because it has only an abstract view of the TE network (as in use case in 2.1).

The cloud orchestrator can request to the TE domain controller to compute the cost of the possible TE paths (e.g., DC1-DC2 and DC1-DC3) and to the DC controller to compute the cost of the computing power (DC resources) within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

3. Interactions with TE Topology

The use cases described in section 2 have been described assuming that the topology view exported by each underlying SDN controller to the orchestrator is aggregated using the "virtual node model", defined in [RFC7926].

TE Topology information, e.g., as provided by [TE-TOPO], could in theory be used by an underlying SDN controllers to provide TE information to the orchestrator thus allowing the Path Computation Element (PCE) within the Orchestrator to perform multi-domain path computation by its own, without requesting path computations to the underlying SDN controllers.

This section analyzes the need for an orchestrator to request underlying SDN controllers for path computation even in these scenarios as well as how the TE Topology information and the path computation can be complementary.

In nutshell, there is a scalability trade-off between providing all the TE information needed by the Orchestrator's PCE to take optimal path computation decisions by its own versus requesting the Orchestrator to ask to too many underlying SDN Domain Controllers a set of feasible optimal intra-domain TE paths.

3.1. TE Topology Aggregation using the "virtual link model"

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export the whole TE domain as a single abstract TE node with a "detailed connectivity matrix", which extends the "connectivity matrix", defined in [RFC7446], with specific TE attributes (e.g., delay, SRLGs and summary TE metrics).

The information provided by the "detailed abstract connectivity matrix" would be equivalent to the information that should be provided by "virtual link model" as defined in [RFC7926].

For example, in the IP-Optical integration use case, described in section 2.1, the Optical domain controller can make the information shown in Figure 3 available to the Orchestrator as part of the TE Topology information and the Orchestrator could use this information to calculate by its own the optimal path between routers R1 and R2, without requesting any additional information to the Optical Domain Controller.

However, there is a tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator's PCE) and scalability to be considered when designing the amount of information to provide within the "detailed abstract connectivity matrix".

Figure 6 below shows another example, similar to Figure 3, where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).

```

-----
I                                     I
I           IP+Optical Path Computation Example           I
I               with multiple choices                       I
I                                                         I
I                                                         I
I                                                         I
I               (only in PDF version)                       I
I                                                         I
I                                                         I
-----

```

Figure 6 - IP+Optical Path Computation Example with multiple choices

Reporting all the information, as in Figure 6, using the "detailed abstract connectivity matrix", is quite challenging from a scalability perspective. The amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user constraints / policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min OSNR margin, max preFEC BER etc. All these constraints could be different based on connectivity requirements.

In the following table, a list of the possible constraints, associated with their potential cardinality, is reported.

The maximum number of potential connections to be computed and reported is, in first approximation, the multiplication of all of them.

Constraint Cardinality

End points $N(N-1)/2$ if connections are bidirectional (OTN and WDM),
 $N(N-1)$ for unidirectional connections.

Bandwidth In WDM networks, bandwidth values are expressed in GHz.

On fixed-grid WDM networks, the central frequencies are on a 50GHz grid and the channel width of the transmitters are typically 50GHz such that each central frequency can be used, i.e., adjacent channels can be placed next to each other in terms of central frequencies.

On flex-grid WDM networks, the central frequencies are on a 6.25GHz grid and the channel width of the transmitters can be multiples of 12.5GHz.

For fixed-grid WDM networks typically there is only one possible bandwidth value (i.e., 50GHz) while for flex-grid WDM networks typically there are 4 possible bandwidth values (e.g., 37.5GHz, 50GHz, 62.5GHz, 75GHz).

In OTN (ODU) networks, bandwidth values are expressed as pairs of ODU type and, in case of ODUFlex, ODU rate in bytes/sec as described in section 5 of [RFC7139].

For "fixed" ODUk types, 6 possible bandwidth values are possible (i.e., ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4).

For ODUFlex(GFP), up to 80 different bandwidth values can be specified, as defined in Table 7-8 of [ITU-T G.709-2016].

For other ODUFlex types, like ODUFlex(CBR), the number of possible bandwidth values depends on the rates of the clients that could be mapped over these ODUFlex types, as shown in Table 7.2 of [ITU-T G.709-2016], which in theory could be a continuum of values. However, since different ODUFlex bandwidths that use the same number of TSs on each link along the path are equivalent for path computation purposes, up to 120 different bandwidth ranges can be specified.

Ideas to reduce the number of ODUFlex bandwidth values in the detailed connectivity matrix, to less than 100, are for further study.

Bandwidth specification for ODUCn is currently for further study but it is expected that other bandwidth values can be specified as integer multiples of 100Gb/s.

In IP we have bandwidth values in bytes/sec. In principle, this is a countinuum of values, but in practice we can identify a set of bandwidth ranges, where any bandwidth value inside the same range produces the same path.

The number of such ranges is the cardinality, which depends on the topology, available bandwidth and status of the network. Simulations (Note: reference paper submitted for publication) show that values for medium size topologies (around 50-150 nodes) are in the range 4-7 (5 on average) for each end points couple.

Metrics IGP, TE and hop number are the basic objective metrics defined so far. There are also the 2 objective functions defined in [RFC5541]: Minimum Load Path (MLP) and Maximum Residual Bandwidth Path (MBP). Assuming that one only metric or objective function can be optimized at once, the total cardinality here is 5.

With [PCEP-Service-Aware], a number of additional metrics are defined, including Path Delay metric, Path Delay Variation metric and Path Loss metric, both for point-to-point and point-to-multipoint paths. This increases the cardinality to 8.

Bounds Each metric can be associated with a bound in order to find a path having a total value of that metric lower than the given bound. This has a potentially very high cardinality (as any value for the bound is allowed). In practice there is a maximum value of the bound (the one with the maximum value of the associated metric) which results always in the same path, and a range approach like for bandwidth in IP should produce also in this case the cardinality. Assuming to have a cardinality similar to the one of the bandwidth (let say 5 on average) we should have 6 (IGP, TE, hop, path delay, path delay variation and path loss; we don't consider here the two

objective functions of [RFC5541] as they are conceived only for optimization)*5 = 30 cardinality.

Priority We have 8 values for setup priority, which is used in path computation to route a path using free resources and, where no free resources are available, resources used by LSPs having a lower holding priority.

Local prot It's possible to ask for a local protected service, where all the links used by the path are protected with fast reroute (this is only for IP networks, but line protection schemas are available on the other technologies as well). This adds an alternative path computation, so the cardinality of this constraint is 2.

Administrative

Colors Administrative colors (aka affinities) are typically assigned to links but when topology abstraction is used affinity information can also appear in the detailed connectivity matrix.

There are 32 bits available for the affinities. Links can be tagged with any combination of these bits, and path computation can be constrained to include or exclude any or all of them. The relevant cardinality is 3 (include-any, exclude-any, include-all) times 2^{32} possible values. However, the number of possible values used in real networks is quite small.

Included Resources

A path computation request can be associated to an ordered set of network resources (links, nodes) to be included along the computed path. This constraint would have a huge cardinality as in principle any combination of network resources is possible. However, as far as the Orchestrator doesn't know details about the internal topology of the domain, it shouldn't include this type of constraint at all (see more details below).

Excluded Resources

A path computation request can be associated to a set of network resources (links, nodes, SRLGs) to be excluded from the computed path. Like for included resources,

this constraint has a potentially very high cardinality,
 but, once again, it can't be actually used by the
 Orchestrator, if it's not aware of the domain topology
 (see more details below).

As discussed above, the Orchestrator can specify include or exclude resources depending on the abstract topology information that the domain controller exposes:

- o In case the domain controller exposes the entire domain as a single abstract TE node with his own external terminations and connectivity matrix (whose size we are estimating), no other topological details are available, therefore the size of the connectivity matrix only depends on the combination of the constraints that the Orchestrator can use in a path computation request to the domain controller. These constraints cannot refer to any details of the internal topology of the domain, as those details are not known to the Orchestrator and so they do not impact size of connectivity matrix exported.
- o Instead in case the domain controller exposes a topology including more than one abstract TE nodes and TE links, and their attributes (e.g. SRLGs, affinities for the links), the Orchestrator knows these details and therefore could compute a path across the domain referring to them in the constraints. The connectivity matrixes to be estimated here are the ones relevant to the abstract TE nodes exported to the Orchestrator. These connectivity matrixes and therefore theirs sizes, while cannot depend on the other abstract TE nodes and TE links, which are external to the given abstract node, could depend to SRLGs (and other attributes, like affinities) which could be present also in the portion of the topology represented by the abstract nodes, and therefore contribute to the size of the related connectivity matrix.

We also don't consider here the possibility to ask for more than one path in diversity or for point-to-multi-point paths, which are for further study.

Considering for example an IP domain without considering SRLG and affinities, we have an estimated number of paths depending on these estimated cardinalities:

Endpoints = $N*(N-1)$, Bandwidth = 5, Metrics = 6, Bounds = 20,
Priority = 8, Local prot = 2

The number of paths to be pre-computed by each IP domain is therefore $24960 * N(N-1)$ where N is the number of domain access points.

This means that with just 4 access points we have nearly 300000 paths to compute, advertise and maintain (if a change happens in the domain, due to a fault, or just the deployment of new traffic, a substantial number of paths need to be recomputed and the relevant changes advertised to the upper controller).

This seems quite challenging. In fact, if we assume a mean length of 1K for the json describing a path (a quite conservative estimate), reporting 300000 paths means transferring and then parsing more than 300 Mbytes for each domain. If we assume that 20% (to be checked) of this paths change when a new deployment of traffic occurs, we have 60 Mbytes of transfer for each domain traversed by a new end-to-end path. If a network has, let say, 20 domains (we want to estimate the load for a non-trivial domain setup) in the beginning a total initial transfer of 6Gigs is needed, and eventually, assuming 4-5 domains are involved in mean during a path deployment we could have 240-300 Mbytes of changes advertised to the higher order controller.

Further bare-bone solutions can be investigated, removing some more options, if this is considered not acceptable; in conclusion, it seems that an approach based only on connectivity matrix is hardly feasible, and could be applicable only to small networks with a limited meshing degree between domains and renouncing to a number of path computation features.

It is also worth noting that the "connectivity matrix" has been originally defined in WSON, [RFC7446] to report the connectivity constraints of a physical node within the WDM network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

Using the "connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the connectivity constraints of an Optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established. The information in the "detailed abstract connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or

available bandwidth) in the "detailed abstract connectivity matrix" while not changing the feasibility of the path.

"Connectivity matrix" is sometimes confused with optical reach table that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools, and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from src to dest, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though "optical reachability table" is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

There is therefore the need to keep the information in the "connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator's PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the Orchestrator's PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed abstract connectivity matrix" that provides accurate, scalable and updated information to allow the Orchestrator's PCE to take optimal decisions by its own.

If the information in the "detailed abstract connectivity matrix" is not complete/accurate, we can have the following drawbacks considering for example the case in Figure 6:

- o If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the Orchestrator's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;

- o If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the Orchestrator's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the Optical domain while the optimal path would actually be the one going through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Instead, using the approach proposed in this document, the Orchestrator, when it needs to setup an end-to-end path, it can request the Optical domain controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- o When setting up a 5 Gb/s path between routers R1 and R2, the Optical domain controller may report only the VP1-VP4 path as the only feasible path: the Orchestrator can successfully setup the end-to-end path passing through this Optical path;
- o When setting up a 1 Gb/s path between routers R1 and R2, the Optical domain controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Orchestrator can then compute the optimal path which is passing through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

3.2. TE Topology Abstraction

Using the TE Topology model, as defined in [TE-TOP0], the underlying SDN controller can export an abstract TE Topology, composed by a set of TE nodes and TE links, which are abstracting the topology controlled by each domain controller.

Considering the example in Figure 4, the TE domain controller 1 can export a TE Topology encompassing the TE nodes A, B, C and D and the TE Link interconnecting them. In a similar way, TE domain controller 2 can export a TE Topology encompassing the TE nodes E, F, G and H and the TE Link interconnecting them.

In this example, for simplicity reasons, each abstract TE node maps with each physical node, but this is not necessary.

In order to setup a multi-domain TE path (e.g., between nodes A and H), the Orchestrator can compute by its own an optimal end-to-end path based on the abstract TE topology information provided by the domain controllers. For example:

- o Orchestrator's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H, and then request the TE domain controllers to setup the A-B-C and E-G-H intra-domain paths
- o But, during path setup, the domain controller may find out that A-B-C intra-domain path is not feasible (as discussed in section 2.2, in optical networks it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller), while only the path A-B-D is feasible
- o So what the hierarchical controller computed is not good and need to re-start the path computation from scratch

As discussed in section 3.1, providing more extensive abstract information from the TE domain controllers to the multi-domain Orchestrator may lead to scalability problems.

In a sense this is similar to the problem of routing and wavelength assignment within an Optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for Optical path setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (step 1) and then compute an intra-domain path within each Optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path that performing per-domain path computation and then stitch them.

3.3. Complementary use of TE topology and path computation

As discussed in section 2.2, there are some scalability issues with path computation requests in a multi-domain TE network with many TE domains, in terms of the number of requests to send to the TE domain controllers. It would therefore be worthwhile using the TE topology information provided by the domain controllers to limit the number of requests.

An example can be described considering the multi-domain abstract topology shown in Figure 7. In this example, an end-to-end TE path between domains A and F needs to be setup. The transit domain should be selected between domains B, C, D and E.

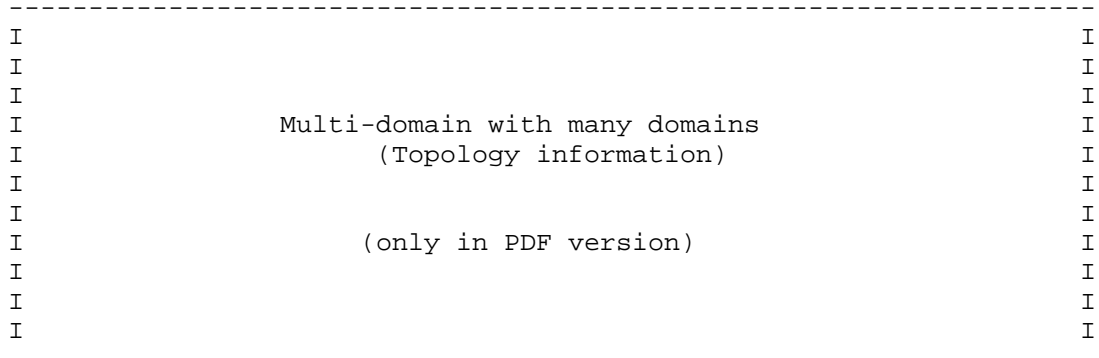


Figure 7 - Multi-domain with many domains (Topology information)

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Orchestrator only knows, from the TE topology provided by the underlying domain controllers, the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Orchestrator can understand by its own that:

- o Domain B cannot be selected as the path connecting domains A and E is not feasible;
- o Domain E cannot be selected as a transit domain since it is known from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-E-F (which is 80, in the worst case)

Therefore, the Orchestrator can understand by its own that the optimal multi-domain path could be either A-D-F or A-E-F but it cannot know which one of the two possible options actually provides the optimal end-to-end path.

The Orchestrator can therefore request path computation only to the TE domain controllers A, D, E and F (and not to all the possible TE domain controllers).

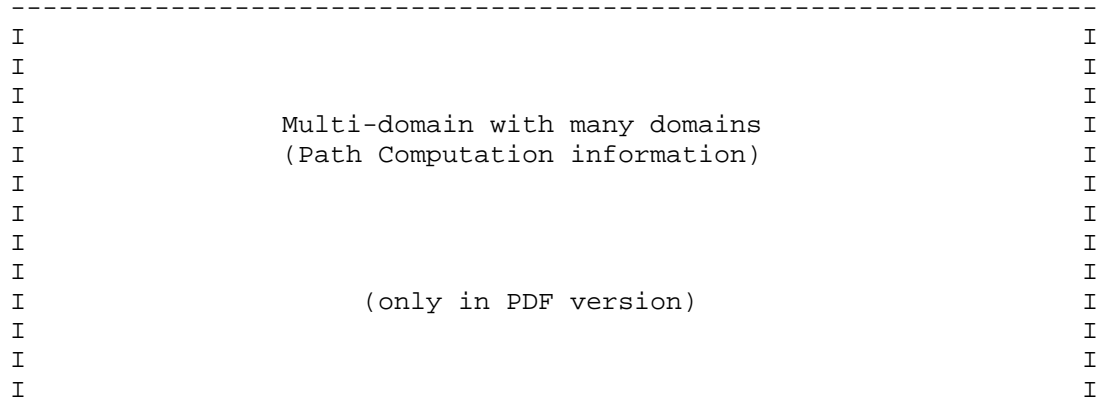


Figure 8 - Multi-domain with many domains (Path Computation information)

Based on these requests, the Orchestrator can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 8, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

4. Motivation for a YANG Model

4.1. Benefits of common data models

Path computation requests should be closely aligned with the YANG data models that provide (abstract) TE topology information, i.e., [TE-TOPO] as well as that are used to configure and manage TE Tunnels, i.e., [TE-TUNNEL]. Otherwise, an error-prone mapping or correlation of information would be required. For instance, there is benefit in using the same endpoint identifiers in path computation requests and in the topology modeling. Also, the attributes used in path computation constraints could use the same or similar data models. As a result, there are many benefits in aligning path computation requests with YANG models for TE topology information and TE Tunnels configuration and management.

4.2. Benefits of a single interface

A typical use case for path computation requests is the interface between an orchestrator and a domain controller. The system integration effort is typically lower if a single, consistent interface is used between such systems, i.e., one data modeling language (i.e., YANG) and a common protocol (e.g., NETCONF or RESTCONF).

Practical benefits of using a single, consistent interface include:

1. Simple authentication and authorization: The interface between different components has to be secured. If different protocols have different security mechanisms, ensuring a common access control model may result in overhead. For instance, there may be a need to deal with different security mechanisms, e.g., different credentials or keys. This can result in increased integration effort.
2. Consistency: Keeping data consistent over multiple different interfaces or protocols is not trivial. For instance, the sequence of actions can matter in certain use cases, or transaction semantics could be desired. While ensuring consistency within one protocol can already be challenging, it is typically cumbersome to achieve that across different protocols.
3. Testing: System integration requires comprehensive testing, including corner cases. The more different technologies are involved, the more difficult it is to run comprehensive test cases and ensure proper integration.
4. Middle-box friendliness: Provider and consumer of path computation requests may be located in different networks, and middle-boxes such as firewalls, NATs, or load balancers may be deployed. In such environments it is simpler to deploy a single protocol. Also, it may be easier to debug connectivity problems.
5. Tooling reuse: Implementers may want to implement path computation requests with tools and libraries that already exist in controllers and/or orchestrators, e.g., leveraging the rapidly growing eco-system for YANG tooling.

4.3. Extensibility

Path computation is only a subset of the typical functionality of a controller. In many use cases, issuing path computation requests comes along with the need to access other functionality on the same

system. In addition to obtaining TE topology, for instance also configuration of services (setup/modification/deletion) may be required, as well as:

1. Receiving notifications for topology changes as well as integration with fault management
2. Performance management such as retrieving monitoring and telemetry data
3. Service assurance, e.g., by triggering OAM functionality
4. Other fulfilment and provisioning actions beyond tunnels and services, such as changing QoS configurations

YANG is a very extensible and flexible data modeling language that can be used for all these use cases.

Adding support for path computation requests to YANG models would seamlessly complement with [TE-TOPO] and [TE-TUNNEL] in the use cases where YANG-based protocols (e.g., NETCONF or RESTCONF) are used.

5. Path Computation for multiple LSPs

There are use cases, where path computation is required for multiple Traffic Engineering Label Switched Paths (TE LSPs) through a network or through a network domain. It may be advantageous to request the new paths for a set of LSPs in one single path computation request [RFC5440] that also includes information regarding the desired objective function, see [RFC5541].

In the context of abstraction and control of TE networks (ACTN), as defined in [ACTN-Frame], when a MDSC receives a virtual network (VN) request from a CNC, the MDSC needs to perform path computation for multiple LSPs as a typical VN is constructed by a set of multiple paths also called end-to-end tunnels. The MDSC may send a single path computation request to the PNC for multiple LSPs, i.e. between the VN end points (access points in ACTN terminology).

In a more general context, when a MDSC needs to send multiple path provisioning requests to the PNC, the MDSC may also group these path provisioning requests together and send them in a single message to the PNC instead of sending separate requests for each path.

6. YANG Model for requesting Path Computation

The TE Tunnel YANG model has been extended to support the need to request path computation.

It is possible to request path computation by configuring a "compute-only" TE tunnel and retrieving the computed path(s) in the LSP(s) Record-Route Object (RRO) list as described in section 3.3.1 of [TE-TUNNEL].

This is a stateful solution since the state of each created "compute-only" TE tunnel needs to be maintained and updated, when underlying network conditions change.

The need also for a stateless solution, based on an RPC, has been recognized, as outlined in section 6.1.1.

A proposal for a stateless RPC to request path computation is provided in section 6.2.

This is intended as an input for further evaluation and discussion with the authors of [TE-TUNNEL] Internet-Draft and TEAS WG participants, about the technical solution as well as whether this RPC should be merged with the YANG model defined in [TE-TUNNEL].

6.1. Modeling Considerations

6.1.1. Stateless and Stateful Path Computation

For further study.

6.1.2. Reduction of Path Computation Requests

For further study.

6.2. YANG model for stateless TE path computation

6.2.1. YANG Tree

Figure 9 below shows the tree diagram of the YANG model defined in module ietf-te-path-computation.yang.

```
module: ietf-te-path-computation
  +--rw paths
```

```

    +--ro path* [path-id]
      +--ro _telink* [link-ref]
        +--ro link-ref ->
/nd:networks/network[nd:network-id=current()/../network-
ref]/lnk:link/link-id
    +--ro network-ref? -> /nd:networks/network/network-id
  +--ro _routingConstraint
    +--ro requestedCapacity? tet:te-bandwidth
  +--ro pathConstraints
    +--ro path-constraints
      +--ro topology-id? te-types:te-topology-id
      +--ro cost-limit? uint32
      +--ro hop-limit? uint8
      +--ro metric-type? identityref
      +--ro tiebreaker-type? identityref
      +--ro ignore-overload? boolean
      +--ro path-affinities {named-path-affinities}?
        +--ro (style)?
          +--:(values)
            +--ro value? uint32
            +--ro mask? uint32
          +--:(named)
            +--ro constraints* [usage]
              +--ro usage identityref
              +--ro constraint
                +--ro affinity-names* [name]
                  +--ro name string
      +--ro path-srlgs
        +--ro (style)?
          +--:(values)
            +--ro usage? identityref
            +--ro values* te-types:srlg
          +--:(named)
            +--ro constraints* [usage]
              +--ro usage identityref
              +--ro constraint
                +--ro srlg-names* [name]
                  +--ro name string
    +--ro bidirectional

```

```

| | | +--ro association
| | |   +--ro id?                uint16
| | |   +--ro source?           inet:ip-address
| | |   +--ro global-source?    inet:ip-address
| | |   +--ro type?             identityref
| | |   +--ro provisioing?      identityref
| | | +--ro _avoidTopology
| | |   +--ro provider-ref?      ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:provider-id
| | |   +--ro client-ref?        ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:client-id
| | |   +--ro te-topology-ref?   ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:te-topology-id
| | |   +--ro network-ref?       ->
/nw:networks/network/network-id
| | |   +--ro path-id            yang-types:uuid
+--rw pathComputationService
+--ro _path-ref*                 -> /paths/path/path-id
+--rw _servicePort
| | | +--rw source?              inet:ip-address
| | | +--rw destination?        inet:ip-address
| | | +--rw src-tp-id?           binary
| | | +--rw dst-tp-id?           binary
| | | +--rw bidirectional
| | |   +--rw association
| | |     +--rw id?              uint16
| | |     +--rw source?          inet:ip-address
| | |     +--rw global-source?   inet:ip-address
| | |     +--rw type?            identityref
| | |     +--rw provisioing?     identityref
+--rw _routingConstraint
| | | +--ro requestedCapacity?    tet:te-bandwidth
| | | +--ro pathConstraints
| | |   +--ro path-constraints
| | |     +--ro topology-id?     te-types:te-topology-id
| | |     +--ro cost-limit?      uint32

```

si, Belotti, et al. Expires September 3, 2017 [Page 28]

```

    |      +---ro client-ref?      ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:client-id
    |      +---ro te-topology-ref?  ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:te-topology-id
    |      +---ro network-ref?      ->
/nw:networks/network/network-id
    +---rw _objectiveFunction
    |   +---ro objectiveFunction?   ObjectiveFunction
    +---rw _optimizationConstraint
    |   +---ro trafficInterruption?  DirectiveValue

rpcs:
+---x statelessComputeP2PPath
|   +---w input
|   |   +---w servicePort*
|   |   |   +---w source?          inet:ip-address
|   |   |   +---w destination?     inet:ip-address
|   |   |   +---w src-tp-id?       binary
|   |   |   +---w dst-tp-id?       binary
|   |   |   +---w bidirectional
|   |   |   |   +---w association
|   |   |   |   |   +---w id?          uint16
|   |   |   |   |   +---w source?     inet:ip-address
|   |   |   |   |   +---w global-source? inet:ip-address
|   |   |   |   |   +---w type?       identityref
|   |   |   |   |   +---w provisioing? identityref
|   |   +---w routingConstraint
|   |   |   +---w requestedCapacity?  tet:te-bandwidth
|   |   |   +---w pathConstraints
|   |   |   |   +---w path-constraints
|   |   |   |   |   +---w topology-id?  te-types:te-topology-id
|   |   |   |   |   +---w cost-limit?   uint32
|   |   |   |   |   +---w hop-limit?    uint8
|   |   |   |   |   +---w metric-type?   identityref
|   |   |   |   |   +---w tiebreaker-type? identityref
|   |   |   |   |   +---w ignore-overload? boolean
|   |   |   |   +---w path-affinities {named-path-affinities}?

```

```

| | | | | +---w (style)?
| | | | | +---:(values)
| | | | | | +---w value?          uint32
| | | | | | +---w mask?          uint32
| | | | | +---:(named)
| | | | | | +---w constraints* [usage]
| | | | | | | +---w usage          identityref
| | | | | | | +---w constraint
| | | | | | | | +---w affinity-names* [name]
| | | | | | | | | +---w name      string
| | | | | +---w path-srlgs
| | | | | | +---w (style)?
| | | | | | +---:(values)
| | | | | | | +---w usage?          identityref
| | | | | | | +---w values*         te-types:srlg
| | | | | | +---:(named)
| | | | | | | +---w constraints* [usage]
| | | | | | | | +---w usage          identityref
| | | | | | | | +---w constraint
| | | | | | | | | +---w srlg-names* [name]
| | | | | | | | | | +---w name      string
| | | | | +---w bidirectional
| | | | | | +---w association
| | | | | | | +---w id?              uint16
| | | | | | | +---w source?          inet:ip-address
| | | | | | | +---w global-source?   inet:ip-address
| | | | | | | +---w type?            identityref
| | | | | | | +---w provisioing?     identityref
| | | | | +---w _avoidTopology
| | | | | | +---w provider-ref?      ->
| | | | | /nw:networks/network[nw:network-id = current()/../network-
| | | | | ref]/tet:provider-id
| | | | | | +---w client-ref?        ->
| | | | | /nw:networks/network[nw:network-id = current()/../network-
| | | | | ref]/tet:client-id
| | | | | | +---w te-topology-ref?   ->
| | | | | /nw:networks/network[nw:network-id = current()/../network-
| | | | | ref]/tet:te-topology-id

```

```

| | | +---w network-ref? ->
/nw:networks/network/network-id
| | +---w objectiveFunction
| | +---w objectiveFunction? ObjectiveFunction
+--ro output
+--ro pathCompService
+--ro _path-ref* -> /paths/path/path-id
+--ro _servicePort
| +--ro source? inet:ip-address
| +--ro destination? inet:ip-address
| +--ro src-tp-id? binary
| +--ro dst-tp-id? binary
| +--ro bidirectional
| +--ro association
| | +--ro id? uint16
| | +--ro source? inet:ip-address
| | +--ro global-source? inet:ip-address
| | +--ro type? identityref
| | +--ro provisioing? identityref
+--ro _routingConstraint
| +--ro requestedCapacity? tet:te-bandwidth
+--ro pathConstraints
| +--ro path-constraints
| | +--ro topology-id? te-types:te-topology-
id | | |
| | | +--ro cost-limit? uint32
| | | +--ro hop-limit? uint8
| | | +--ro metric-type? identityref
| | | +--ro tiebreaker-type? identityref
| | | +--ro ignore-overload? boolean
| | | +--ro path-affinities {named-path-affinities}?
| | | | +--ro (style)?
| | | | | +--:(values)
| | | | | | +--ro value? uint32
| | | | | | +--ro mask? uint32
| | | | | +--:(named)
| | | | | +--ro constraints* [usage]
| | | | | | +--ro usage identityref
| | | | | +--ro constraint

```

```

/nw:networks/network[nw:network-id = current()../network-
ref]/tet:provider-id
|
|      +--ro client-ref?          ->
/nw:networks/network[nw:network-id = current()../network-
ref]/tet:client-id
|
|      +--ro te-topology-ref?    ->
/nw:networks/network[nw:network-id = current()../network-
ref]/tet:te-topology-id
|
|      +--ro network-ref?        ->
/nw:networks/network/network-id
|
|      +--ro _objectiveFunction
|      |      +--ro objectiveFunction?  ObjectiveFunctioni
|      |      +--ro _optimizationConstraint
|      |      |      +--ro trafficInterruption?  DirectiveValu
+---x optimizeP2PPath
+---w input
|      +---w pathIdOrName?          string

```



```

+---w routingConstraint
+---w requestedCapacity?    tet:te-bandwidth
+---w pathConstraints
+---w path-constraints
+---w topology-id?          te-types:te-topology-id
+---w cost-limit?           uint32
+---w hop-limit?            uint8
+---w metric-type?          identityref
+---w tiebreaker-type?      identityref
+---w ignore-overload?      boolean
+---w path-affinities {named-path-affinities}?
|   +---w (style)?
|   |   +---:(values)
|   |   |   +---w value?          uint32
|   |   |   +---w mask?          uint32
|   |   +---:(named)
|   |       +---w constraints* [usage]
|   |       |   +---w usage          identityref
|   |       |   +---w constraint
|   |       |       +---w affinity-names* [name]
|   |       |       +---w name      string
|   +---w path-srlgs
|   |   +---w (style)?
|   |   |   +---:(values)
|   |   |   |   +---w usage?          identityref
|   |   |   |   +---w values*        te-types:srlg
|   |   +---:(named)
|   |       +---w constraints* [usage]
|   |       |   +---w usage          identityref
|   |       |   +---w constraint
|   |       |       +---w srlg-names* [name]
|   |       |       +---w name      string
+---w bidirectional
+---w association
+---w id?                    uint16
+---w source?                inet:ip-address
+---w global-source?         inet:ip-address
+---w type?                  identityref
+---w provisioing?           identityref

```

```

      | | +---w _avoidTopology
      | | +---w provider-ref? ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:provider-id
      | | +---w client-ref? ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:client-id
      | | +---w te-topology-ref? ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:te-topology-id
      | | +---w network-ref? ->
/nw:networks/network/network-id
      | +---w optimizationConstraint
      | | +---w trafficInterruption? DirectiveValue
      | +---w objectiveFunction
      | | +---w objectiveFunction? ObjectiveFunction
+--ro output
+--ro pathCompService
+--ro _path-ref* -> /paths/path/path-id
+--ro _servicePort
| +--ro source? inet:ip-address
| +--ro destination? inet:ip-address
| +--ro src-tp-id? binary
| +--ro dst-tp-id? binary
| +--ro bidirectional
| | +--ro association
| | | +--ro id? uint16
| | | +--ro source? inet:ip-address
| | | +--ro global-source? inet:ip-address
| | | +--ro type? identityref
| | | +--ro provisioing? identityref
+--ro _routingConstraint
| +--ro requestedCapacity? tet:te-bandwidth
+--ro pathConstraints
| | +--ro path-constraints
| | | +--ro topology-id? te-types:te-topology-
id
| | +--ro cost-limit? uint32
| | +--ro hop-limit? uint8

```

```
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:provider-id
|      +--ro client-ref?      ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:client-id
```

```

      |      +--ro te-topology-ref?    ->
/nw:networks/network[nw:network-id = current()/../network-
ref]/tet:te-topology-id
      |      +--ro network-ref?        ->
/nw:networks/network/network-id
      +--ro _objectiveFunction
      |  +--ro objectiveFunction?      ObjectiveFunction
      +--ro _optimizationConstraint
          +--ro trafficInterruption?    DirectiveValue

```

Figure 9

- TE path computatio

n tree

6.2.2. YANG Module

```

<CODE BEGINS>file " ietf-te-path-computation.yang "
module ietf-te-path-computation {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";
  // replace with IANA namespace when assigned

  prefix "tepc";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang-types";
  }

  import ietf-te-types {
    prefix "te-types";
  }

  import ietf-te-topology {
    prefix "tet";
  }

  import ietf-network-topology {
    prefix "nt";
  }

```

```
}

organization
  "Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";

contact
  "WG Web:    <http://tools.ietf.org/wg/teas/>
  WG List:    <mailto:teas@ietf.org>

  WG Chair:   Lou Berger
              <mailto:lberger@labn.net>

  WG Chair:   Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>

  ";

description "YANG model for stateless TE path computation";

revision "2016-10-10" {
  description "Initial revision";
  reference "YANG model for stateless TE path computation";
}

/*
 * Features
 */

feature stateless-path-computation {
  description
    "This feature indicates that the system supports
    stateless path computation.";
}

/*
 * Typedefs
 */
```

```
typedef DirectiveValue {
  type enumeration {
    enum MINIMIZE {
      description "Minimize directive.";
    }
    enum MAXIMIZE {
      description "Maximize directive.";
    }
    enum ALLOW {
      description "Allow directive.";
    }
    enum DISALLOW {
      description "Disallow directive.";
    }
    enum DONT_CARE {
      description "Don't care directive.";
    }
  }
  description "Value to determine optimization type.";
}

typedef ObjectiveFunction {
  type enumeration {
    enum MCP {
      description "MCP.";
    }
    enum MLP {
      description "MLP.";
    }
    enum MBP {
      description "MBP.";
    }
    enum MBC {
      description "MBC.";
    }
    enum MLL {
      description "MLL.";
    }
    enum MCC {
```

```
        description "MCC.";
    }
}
description "RFC 5541 - Encoding of Objective Functions in the
    Path Computation Element Communication Protocol (PCEP)";
}

/*
 * Groupings
 */

grouping Path {
    list _telink {
        key 'link-ref';
        config false;
        uses nt:link-ref;
        description "List of telink refs.";
    }
    container _routingConstraint {
        config false;
        uses RoutingConstraint;
        description "Extended routing constraints.";
    }
    leaf path-id {
        type yang-types:uuid;
        config false;
        description "path-id ref.";
    }
    description "Path is described by an ordered list of TE Links.";
}

grouping PathCompServicePort {
    leaf source {
        type inet:ip-address;
        description "TE tunnel source address.";
    }
    leaf destination {
        type inet:ip-address;
        description "P2P tunnel destination address";
    }
}
```

```
    }
    leaf src-tp-id {
      type binary;
      description "TE tunnel source termination point identifier.";
    }
    leaf dst-tp-id {
      type binary;
      description "TE tunnel destination termination point
identifier.";
    }
    uses te-types:bidir-assoc-properties;
    description "Path Computation Service Port grouping.";
  }

  grouping PathComputationService {
    leaf-list _path-ref {
      type leafref {
        path '/paths/path/path-id';
      }
      config false;
      description "List of previously computed path references.";
    }
    container _servicePort {
      uses PathCompServicePort;
      description "Path Computation Service Port.";
    }
    container _routingConstraint {
      uses RoutingConstraint;
      description "Routing constraints.";
    }
    container _objectiveFunction {
      uses PathObjectiveFunction;
      description "Path Objective Function.";
    }
    container _optimizationConstraint {
      uses PathOptimizationConstraint;
      description "Path Optimization Constraint.";
    }
    description "Path computation service.";
```



```
}

grouping PathObjectiveFunction {
  leaf objectiveFunction {
    type ObjectiveFunction;
    config false;
    description "Objective Function.";
  }
  description "Path Objective Function.";
}

grouping PathOptimizationConstraint {
  leaf trafficInterruption {
    type DirectiveValue;
    config false;
    description "Traffic Interruption.";
  }
  description "Path Optimization Constraint.";
}

grouping RoutingConstraint {
  leaf requestedCapacity {
    type tet:te-bandwidth;
    config false;
    description "Capacity required for connectivity service.";
  }
  container pathConstraints {
    config false;
    uses te-types:path-constraints;
    description "Service connectivity path selection properties";
  }
  uses te-types:bidir-assoc-properties;
  // path-constraints contains include topology
  /*leaf _includeTopology {
    uses te-types:te-topology-ref;
    config false;
  }*/
  container _avoidTopology {
    uses tet:te-topology-ref;
  }
}
```

```
        config false;
        description "Topology to be avoided.";
    }
    // path-constrains already include/exclude path
    /*list _includePath {
        key 'link-ref';
        config false;
        uses nt:link-ref;
    }*/
    /*list _excludePath {
        key 'link-ref';
        config false;
        uses nt:link-ref;
    }*/
    description "Extended routing constraints. Created to align with
path-constraints.";
}

/*
 * Root container
 */
container paths {
    list path {
        key "path-id";
        uses Path;
        config false;
        description "List of previous computed paths.";
    }
    description "Root container for path-computation";
}

container pathComputationService {
    uses PathComputationService;
    description "Service for computing paths.";
}

/*****
 * package Interfaces
 *****/
```

```
rpc statelessComputeP2PPath {
  description "statelessComputeP2PPath";
  input {
    list servicePort {
      min-elements 1;
      uses PathCompServicePort;
      description "List of service ports.";
    }
    container routingConstraint {
      uses RoutingConstraint;
      description "routing constraint.";
    }
    container objectiveFunction {
      uses PathObjectiveFunction;
      description "objective function.";
    }
  }
  output {
    container pathCompService {
      uses PathComputationService;
      description "Path computation service.";
    }
  }
}

/**rpc computeP2PPath {
  input {
    list servicePort {
      min-elements 2;
      max-elements 2;
      uses PathCompServicePort;
    }
    container routingConstraint {
      uses RoutingConstraint;
    }
    container objectiveFunction {
      uses PathObjectiveFunction;
    }
  }
}
```

```
    output {
      container pathCompService {
        uses PathComputationService;
      }
    }
  }*/

rpc optimizeP2PPath {
  description "optimizeP2PPath.";
  input {
    leaf pathIdOrName {
      type string;
      description "path id or path name.";
    }
    container routingConstraint {
      uses RoutingConstraint;
      description "routing constraint.";
    }
    container optimizationConstraint {
      uses PathOptimizationConstraint;
      description "optimizationConstraint.";
    }
    container objectiveFunction {
      uses PathObjectiveFunction;
      description "objective function.";
    }
  }
  output {
    container pathCompService {
      uses PathComputationService;
      description "path computation service.";
    }
  }
}

/**rpc deleteP2PPath {
  input {
    leaf pathIdOrName {
      type string;
```

```
    }  
  }  
  output {  
    container pathCompService {  
      uses PathComputationService;  
    }  
  }  
}*/  
  
}  
<CODE ENDS>
```

Figure 10 - TE path computation YANG module

7. Security Considerations

This document describes use cases of requesting Path Computation using YANG models, which could be used at the ABNO Control Interface [RFC7491] and/or between controllers in ACTN [ACTN-frame]. As such, it does not introduce any new security considerations compared to the ones related to YANG specification, ABNO specification and ACTN Framework defined in [RFC6020], [RFC7950], [RFC7491] and [ACTN-frame].

This document also defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC6020] apply for this document as well.

8. IANA Considerations

This section is for further study: to be completed when the YANG model is more stable.

9. References

9.1. Normative References

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

- [RFC7139] Zhang, F. et al., "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.
- [RFC7491] Farrel, A., King, D., "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, March 2015.
- [RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.
- [RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [ACTN-Frame] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of Traffic Engineered Networks" draft-ietf-actn-framework, work in progress.
- [ITU-T G.709-2016] ITU-T Recommendation G.709 (06/16), "Interface for the optical transport network", June 2016

9.2. Informative References

- [RFC5440] Vasseur, YP., Le Roux, JL., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC5541] Le Roux, JL. et al., "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.
- [RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.

[L1-TOPO] Zhang, X. et al., "A YANG Data Model for Layer 1 (ODU) Network Topology", draft-zhang-ccamp-l1-topo-yang, work in progress.

[ACTN-Info] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., "Information Model for Abstraction and Control of Transport Networks", draft-leebelotti-actn-info, work in progress.

[PCEP-Service-Aware] Dhody, D. et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", draft-ietf-pce-pcep-service-aware, work in progress.

10. Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in discussions and providing valuable insights.

The authors would like to thank the authors of the TE Tunnel YANG model [TE-TUNNEL], in particular Igor Bryskin, Tarek Saad and Xufeng Liu, for their inputs to the discussions and support in having consistency between the Path Computation and TE Tunnel YANG models.

This document was prepared using 2-Word-v2.0.template.dot.

Contributors

Dieter Beller
Nokia
Email: dieter.beller@nokia.com

Gianmarco Bruno
Ericsson
Email: gianmarco.bruno@ericsson.com

Francesco Lazzeri
Ericsson
Email: francesco.lazzeri@ericsson.com

Young Lee
Huawei
Email: leeyoung@huawei.com

Carlo Perocchio
Ericsson
Email: carlo.perocchio@ericsson.com

Authors' Addresses

Italo Busi (Editor)
Huawei
Email: italo.busi@huawei.com

Sergio Belotti (Editor)
Nokia
Email: sergio.belotti@nokia.com

Victor Lopez
Telefonica
Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma
Infinera
Email: AnSharma@infinera.com

Yan Shi
China Unicom
Email: shiyan49@chinaunicom.cn

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

Karthik Sethuraman
NEC
Email: karthik.sethuraman@necam.com

Michael Scharf
Nokia
Email: michael.scharf@nokia.com

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com

TEAS Working Group
Internet Draft
Intended status: Informational
Expires: December 2017

Italo Busi (Ed.)
Huawei
Sergio Belotti (Ed.)
Nokia
Victor Lopez
Oscar Gonzalez de Dios
Telefonica
Anurag Sharma
Infinera
Yan Shi
China Unicom
Ricard Vilalta
CTTC
Karthik Sethuraman
NEC

June 30, 2017

Yang model for requesting Path Computation
draft-busibel-teas-yang-path-computation-03.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on December 30, 2016.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

This document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

This document also proposes a yang model for a stateless RPC which complements the stateful solution defined in [TE-TUNNEL].

Table of Contents

1. Introduction.....	3
2. Use Cases.....	4

2.1. IP-Optical integration.....	5
2.1.1. Inter-layer path computation.....	6
2.1.2. Route Diverse IP Services.....	8
2.2. Multi-domain TE Networks.....	8
2.3. Data center interconnections.....	9
3. Interactions with TE Topology.....	11
3.1. TE Topology Aggregation using the "virtual link model"...	11
3.2. TE Topology Abstraction.....	19
3.3. Complementary use of TE topology and path computation...	20
4. Motivation for a YANG Model.....	22
4.1. Benefits of common data models.....	22
4.2. Benefits of a single interface.....	23
4.3. Extensibility.....	23
5. Path Computation for multiple LSPs.....	24
6. YANG Model for requesting Path Computation.....	25
6.1. Stateless and Stateful Path Computation.....	25
6.2. YANG model for stateless TE path computation.....	26
6.2.1. YANG Tree.....	26
6.2.2. YANG Module.....	34
7. Security Considerations.....	40
8. IANA Considerations.....	41
9. References.....	41
9.1. Normative References.....	41
9.2. Informative References.....	42
10. Acknowledgments.....	42

1. Introduction

There are scenarios, typically in a hierarchical SDN context, in which an orchestrator may not have detailed information to be able to perform an end-to-end path computation and would need to request lower layer/domain controllers to calculate some (partial) feasible paths.

When we are thinking to this type of scenarios we have in mind specific level of interfaces on which this request can be applied.

We can reference ABNO Control Interface [RFC7491] in which an Application Service Coordinator can request ABNO controller to take in charge path calculation (see Figure 1 in the RFC) and/or ACTN [ACTN-frame], where controller hierarchy is defined, the need for path computation arises on both interfaces CMI (interface between Customer Network Controller(CNC) and Multi Domain Service Coordinator (MDSC)) and/or MPI (interface between MSDC-PNC).[ACTN-

Info] describes an information model for the Path Computation request.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF or RESTCONF).

Path Computation Elements, Controllers and Orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG Data Model for TE Topologies [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [L1-TOPO] for Layer-1 ODU technologies).

The availability of such topology models allows providing the TED using YANG-based protocols (e.g., NETCONF or RESTCONF). Furthermore, it enables a PCE/Controller performing the necessary abstractions or modifications and offering this customized topology to another PCE/Controller or high level orchestrator.

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via YANG-based protocols (e.g., NETCONF or RESTCONF) using the TE-Tunnel Yang model [TE-TUNNEL].

This document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

This document also proposes a yang model for a stateless RPC which complements the stateful solution defined in [TE-TUNNEL].

2. Use Cases

This section presents different use cases, where an orchestrator needs to request underlying SDN controllers for path computation.

The presented uses cases have been grouped, depending on the different underlying topologies: a) IP-Optical integration; b) Multi-domain Traffic Engineered (TE) Networks; and c) Data center interconnections.

2.1. IP-Optical integration

In these use cases, an Optical domain is used to provide connectivity between IP routers which are connected with the Optical domains using access links (see Figure 1).

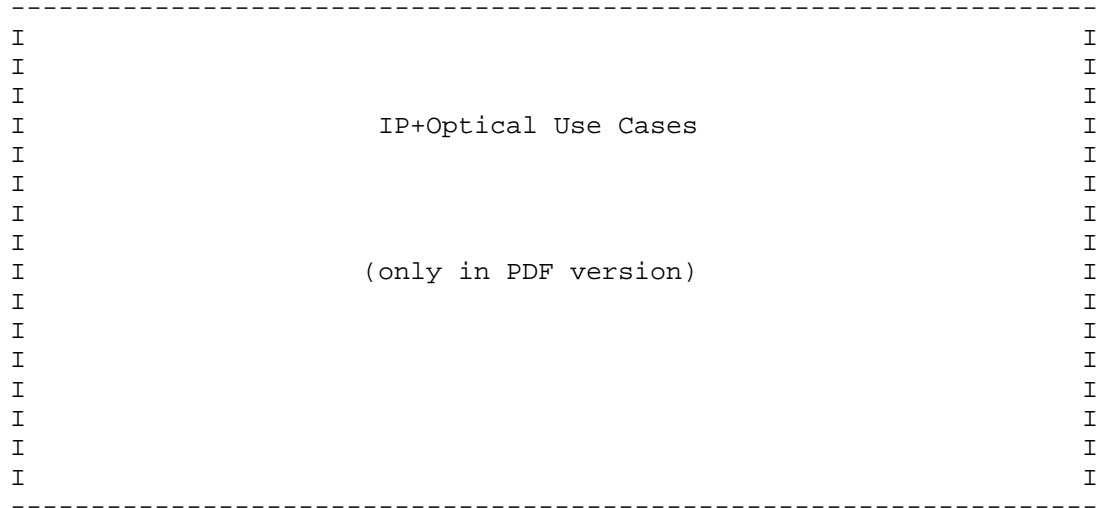


Figure 1 - IP+Optical Use Cases

It is assumed that the Optical domain controller provides to the orchestrator an abstracted view of the Optical network. A possible abstraction shall be representing the optical domain as one "virtual node" with "virtual ports" connected to the access links.

The path computation request helps the orchestrator to know which are the real connections that can be provided at the optical domain.

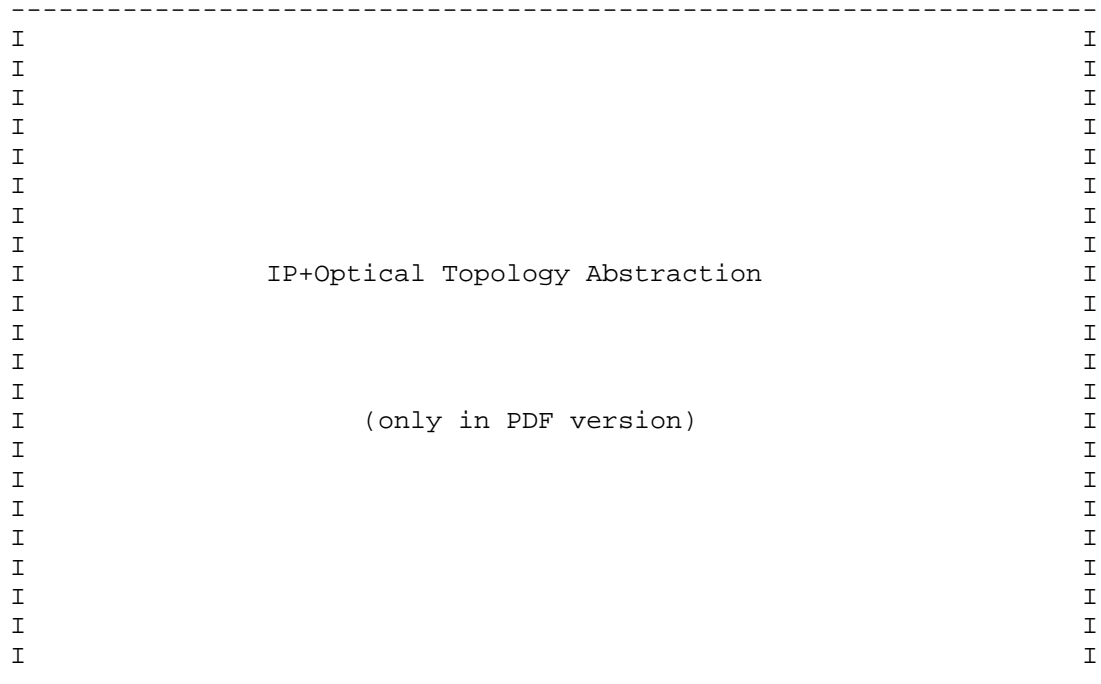


Figure 2 - IP+Optical Topology Abstraction

2.1.1.1. Inter-layer path computation

In this use case, the orchestrator needs to setup an optimal path between two IP routers R1 and R2.

As depicted in Figure 2, the Orchestrator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The orchestrator can request the underlying Optical domain controller to compute a set of potential optimal paths, taking into account optical constraints. Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose

which one of these potential paths to use to setup the optimal e2e path crossing optical network.

```

-----
I                                     I
I           IP+Optical Path Computation Example           I
I                                                         I
I                                                         I
I           (only in PDF version)                         I
I                                                         I
I                                                         I
-----

```

Figure 3 - IP+Optical Path Computation Example

For example, in Figure 3, the Orchestrator can request the Optical domain controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path using the VP2-VP5 Optical path even this is not the optimal path from the Optical domain perspective.

Considering the dynamicity of the connectivity constraints of an Optical domain, it is possible that a path computed by the Optical domain controller when requested by the Orchestrator is no longer valid when the Orchestrator requests it to be setup up.

It is worth noting that with the approach proposed in this document, the likelihood for this issue to happen can be quite small since the time window between the path computation request and the path setup request should be quite short (especially if compared with the time that would be needed to update the information of a very detailed abstract connectivity matrix).

If this risk is still not acceptable, the Orchestrator may also optionally request the Optical domain controller not only to compute the path but also to keep track of its resources (e.g., these resources can be reserved to avoid being used by any other connection). In this case, some mechanism (e.g., a timeout) needs to be defined to avoid having stranded resources within the Optical domain.

These issues and solutions can be fine-tuned during the design of the YANG model for requesting Path Computation.

2.1.2. Route Diverse IP Services

This is for further study.

2.2. Multi-domain TE Networks

In this use case there are two TE domains which are interconnected together by multiple inter-domains links.

A possible example could be a multi-domain optical network.

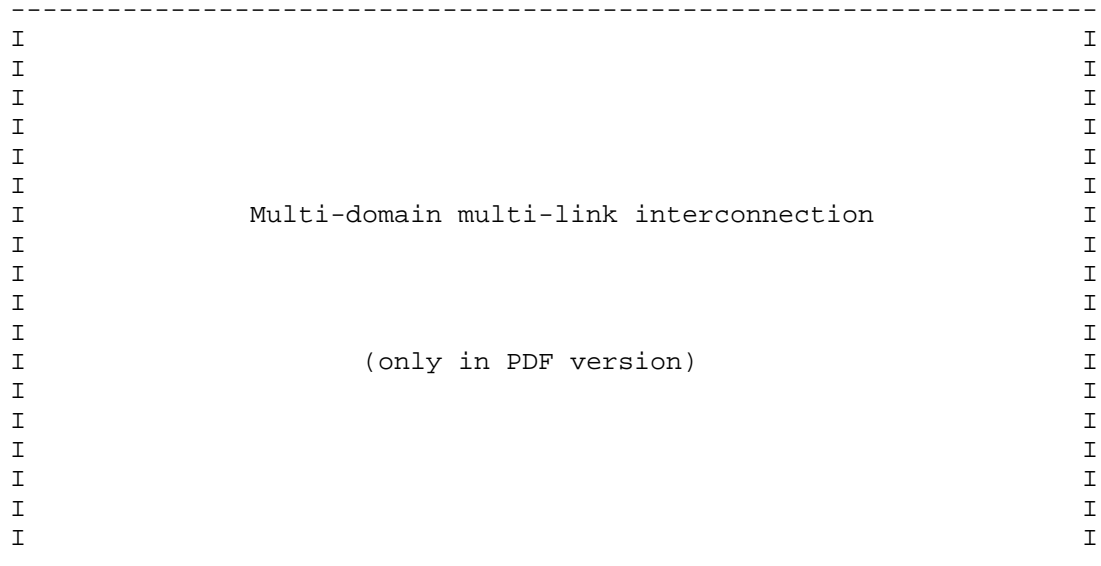


Figure 4 - Multi-domain multi-link interconnection

In order to setup an end-to-end multi-domain TEpath (e.g., between nodes A and H), the orchestrator needs to know the feasibility or the cost of the possible TE paths within the two TE domains, which depend from the current status of the physical resources within each TE network. This is more challenging in case of optical networks because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

In order to setup a multi-domain TE path (e.g., between nodes A and H), Orchestrator can request the TE domain controllers to compute a set of intra-domain optimal paths and take decisions based on the information received. For example:

- o The Orchestrator asks TE domain controllers to provide set of paths between A-C, A-D, E-H and F-H
- o TE domain controllers return a set of feasible paths with the associated costs: the path A-C is not part of this set (in optical networks, it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller)
- o The Orchestrator will select the path A- D-F- H since it is the only feasible multi-domain path and then request the TE domain controllers to setup the A-D and F-H intra-domain paths
- o If there are multiple feasible paths, the Orchestrator can select the optimal path knowing the cost of the intra-domain paths (provided by the TE domain controllers) and the cost of the inter-domain links (known by the Orchestrator)

This approach may have some scalability issues when the number of TE domains is quite big (e.g. 20).

In this case, it would be worthwhile using the abstract TE topology information provided by the domain controllers to limit the number of potential optimal end-to-end paths and then request path computation to fewer domain controllers in order to decide what the optimal path within this limited set is.

For more details, see section 3.3.

2.3. Data center interconnections

In these use case, there is an TE domain which is used to provide connectivity between data centers which are connected with the TE domain using access links.

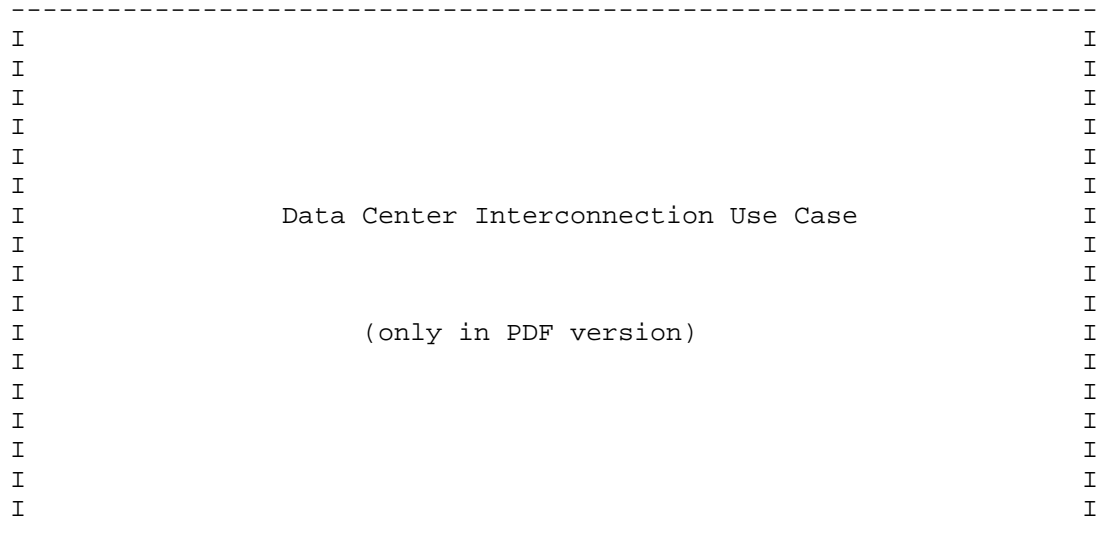


Figure 5 - Data Center Interconnection Use Case

In this use case, a virtual machine within Data Center 1 (DC1) needs to transfer data to another virtual machine that can reside either in DC2 or in DC3.

The optimal decision depends both on the cost of the TE path (DC1-DC2 or DC1-DC3) and of the computing power (data center resources) within DC2 or DC3.

The Cloud Orchestrator may not be able to make this decision because it has only an abstract view of the TE network (as in use case in 2.1).

The cloud orchestrator can request to the TE domain controller to compute the cost of the possible TE paths (e.g., DC1-DC2 and DC1-DC3) and to the DC controller to compute the cost of the computing power (DC resources) within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

3. Interactions with TE Topology

The use cases described in section 2 have been described assuming that the topology view exported by each underlying SDN controller to the orchestrator is aggregated using the "virtual node model", defined in [RFC7926].

TE Topology information, e.g., as provided by [TE-TOPO], could in theory be used by an underlying SDN controllers to provide TE information to the orchestrator thus allowing the Path Computation Element (PCE) within the Orchestrator to perform multi-domain path computation by its own, without requesting path computations to the underlying SDN controllers.

This section analyzes the need for an orchestrator to request underlying SDN controllers for path computation even in these scenarios as well as how the TE Topology information and the path computation can be complementary.

In nutshell, there is a scalability trade-off between providing all the TE information needed by the Orchestrator's PCE to take optimal path computation decisions by its own versus requesting the Orchestrator to ask to too many underlying SDN Domain Controllers a set of feasible optimal intra-domain TE paths.

3.1. TE Topology Aggregation using the "virtual link model"

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export the whole TE domain as a single abstract TE node with a "detailed connectivity matrix", which extends the "connectivity matrix", defined in [RFC7446], with specific TE attributes (e.g., delay, SRLGs and summary TE metrics).

The information provided by the "detailed abstract connectivity matrix" would be equivalent to the information that should be provided by "virtual link model" as defined in [RFC7926].

For example, in the IP-Optical integration use case, described in section 2.1, the Optical domain controller can make the information shown in Figure 3 available to the Orchestrator as part of the TE Topology information and the Orchestrator could use this information to calculate by its own the optimal path between routers R1 and R2, without requesting any additional information to the Optical Domain Controller.

However, there is a tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator's PCE) and scalability to be considered when designing the amount of information to provide within the "detailed abstract connectivity matrix".

Figure 6 below shows another example, similar to Figure 3, where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).

```

-----
I                                     I
I           IP+Optical Path Computation Example           I
I               with multiple choices                       I
I                                                         I
I                                                         I
I                                                         I
I               (only in PDF version)                      I
I                                                         I
-----

```

Figure 6 - IP+Optical Path Computation Example with multiple choices

Reporting all the information, as in Figure 6, using the "detailed abstract connectivity matrix", is quite challenging from a scalability perspective. The amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user constraints / policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min OSNR margin, max preFEC BER etc. All these constraints could be different based on connectivity requirements.

In the following table, a list of the possible constraints, associated with their potential cardinality, is reported.

The maximum number of potential connections to be computed and reported is, in first approximation, the multiplication of all of them.

Constraint Cardinality

End points $N(N-1)/2$ if connections are bidirectional (OTN and WDM),
 $N(N-1)$ for unidirectional connections.

Bandwidth In WDM networks, bandwidth values are expressed in GHz.

On fixed-grid WDM networks, the central frequencies are on a 50GHz grid and the channel width of the transmitters are typically 50GHz such that each central frequency can be used, i.e., adjacent channels can be placed next to each other in terms of central frequencies.

On flex-grid WDM networks, the central frequencies are on a 6.25GHz grid and the channel width of the transmitters can be multiples of 12.5GHz.

For fixed-grid WDM networks typically there is only one possible bandwidth value (i.e., 50GHz) while for flex-grid WDM networks typically there are 4 possible bandwidth values (e.g., 37.5GHz, 50GHz, 62.5GHz, 75GHz).

In OTN (ODU) networks, bandwidth values are expressed as pairs of ODU type and, in case of ODUFlex, ODU rate in bytes/sec as described in section 5 of [RFC7139].

For "fixed" ODUk types, 6 possible bandwidth values are possible (i.e., ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4).

For ODUFlex(GFP), up to 80 different bandwidth values can be specified, as defined in Table 7-8 of [ITU-T G.709-2016].

For other ODUFlex types, like ODUFlex(CBR), the number of possible bandwidth values depends on the rates of the clients that could be mapped over these ODUFlex types, as shown in Table 7.2 of [ITU-T G.709-2016], which in theory could be a continuum of values. However, since different ODUFlex bandwidths that use the same number of TSS on each link along the path are equivalent for path computation purposes, up to 120 different bandwidth ranges can be specified.

Ideas to reduce the number of ODUFlex bandwidth values in the detailed connectivity matrix, to less than 100, are for further study.

Bandwidth specification for ODUCn is currently for further study but it is expected that other bandwidth values can be specified as integer multiples of 100Gb/s.

In IP we have bandwidth values in bytes/sec. In principle, this is a countinuum of values, but in practice we can identify a set of bandwidth ranges, where any bandwidth value inside the same range produces the same path.

The number of such ranges is the cardinality, which depends on the topology, available bandwidth and status of the network. Simulations (Note: reference paper submitted for publication) show that values for medium size topologies (around 50-150 nodes) are in the range 4-7 (5 on average) for each end points couple.

Metrics IGP, TE and hop number are the basic objective metrics defined so far. There are also the 2 objective functions defined in [RFC5541]: Minimum Load Path (MLP) and Maximum Residual Bandwidth Path (MBP). Assuming that one only metric or objective function can be optimized at once, the total cardinality here is 5.

With [PCEP-Service-Aware], a number of additional metrics are defined, including Path Delay metric, Path Delay Variation metric and Path Loss metric, both for point-to-point and point-to-multipoint paths. This increases the cardinality to 8.

Bounds Each metric can be associated with a bound in order to find a path having a total value of that metric lower than the given bound. This has a potentially very high cardinality (as any value for the bound is allowed). In practice there is a maximum value of the bound (the one with the maximum value of the associated metric) which results always in the same path, and a range approach like for bandwidth in IP should produce also in this case the cardinality. Assuming to have a cardinality similar to the one of the bandwidth (let say 5 on average) we should have 6 (IGP, TE, hop, path delay, path delay variation and path loss; we don't consider here the two

objective functions of [RFC5541] as they are conceived only for optimization)*5 = 30 cardinality.

Priority We have 8 values for setup priority, which is used in path computation to route a path using free resources and, where no free resources are available, resources used by LSPs having a lower holding priority.

Local prot It's possible to ask for a local protected service, where all the links used by the path are protected with fast reroute (this is only for IP networks, but line protection schemas are available on the other technologies as well). This adds an alternative path computation, so the cardinality of this constraint is 2.

Administrative

Colors Administrative colors (aka affinities) are typically assigned to links but when topology abstraction is used affinity information can also appear in the detailed connectivity matrix.

There are 32 bits available for the affinities. Links can be tagged with any combination of these bits, and path computation can be constrained to include or exclude any or all of them. The relevant cardinality is 3 (include-any, exclude-any, include-all) times 2^{32} possible values. However, the number of possible values used in real networks is quite small.

Included Resources

A path computation request can be associated to an ordered set of network resources (links, nodes) to be included along the computed path. This constraint would have a huge cardinality as in principle any combination of network resources is possible. However, as far as the Orchestrator doesn't know details about the internal topology of the domain, it shouldn't include this type of constraint at all (see more details below).

Excluded Resources

A path computation request can be associated to a set of network resources (links, nodes, SRLGs) to be excluded from the computed path. Like for included resources,

this constraint has a potentially very high cardinality, but, once again, it can't be actually used by the Orchestrator, if it's not aware of the domain topology (see more details below).

As discussed above, the Orchestrator can specify include or exclude resources depending on the abstract topology information that the domain controller exposes:

- o In case the domain controller exposes the entire domain as a single abstract TE node with his own external terminations and connectivity matrix (whose size we are estimating), no other topological details are available, therefore the size of the connectivity matrix only depends on the combination of the constraints that the Orchestrator can use in a path computation request to the domain controller. These constraints cannot refer to any details of the internal topology of the domain, as those details are not known to the Orchestrator and so they do not impact size of connectivity matrix exported.
- o Instead in case the domain controller exposes a topology including more than one abstract TE nodes and TE links, and their attributes (e.g. SRLGs, affinities for the links), the Orchestrator knows these details and therefore could compute a path across the domain referring to them in the constraints. The connectivity matrixes to be estimated here are the ones relevant to the abstract TE nodes exported to the Orchestrator. These connectivity matrixes and therefore theirs sizes, while cannot depend on the other abstract TE nodes and TE links, which are external to the given abstract node, could depend to SRLGs (and other attributes, like affinities) which could be present also in the portion of the topology represented by the abstract nodes, and therefore contribute to the size of the related connectivity matrix.

We also don't consider here the possibility to ask for more than one path in diversity or for point-to-multi-point paths, which are for further study.

Considering for example an IP domain without considering SRLG and affinities, we have an estimated number of paths depending on these estimated cardinalities:

Endpoints = $N*(N-1)$, Bandwidth = 5, Metrics = 6, Bounds = 20,
Priority = 8, Local prot = 2

The number of paths to be pre-computed by each IP domain is therefore $24960 * N(N-1)$ where N is the number of domain access points.

This means that with just 4 access points we have nearly 300000 paths to compute, advertise and maintain (if a change happens in the domain, due to a fault, or just the deployment of new traffic, a substantial number of paths need to be recomputed and the relevant changes advertised to the upper controller).

This seems quite challenging. In fact, if we assume a mean length of 1K for the json describing a path (a quite conservative estimate), reporting 300000 paths means transferring and then parsing more than 300 Mbytes for each domain. If we assume that 20% (to be checked) of this paths change when a new deployment of traffic occurs, we have 60 Mbytes of transfer for each domain traversed by a new end-to-end path. If a network has, let say, 20 domains (we want to estimate the load for a non-trivial domain setup) in the beginning a total initial transfer of 6Gigs is needed, and eventually, assuming 4-5 domains are involved in mean during a path deployment we could have 240-300 Mbytes of changes advertised to the higher order controller.

Further bare-bone solutions can be investigated, removing some more options, if this is considered not acceptable; in conclusion, it seems that an approach based only on connectivity matrix is hardly feasible, and could be applicable only to small networks with a limited meshing degree between domains and renouncing to a number of path computation features.

It is also worth noting that the "connectivity matrix" has been originally defined in WSON, [RFC7446] to report the connectivity constraints of a physical node within the WDM network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

Using the "connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the connectivity constraints of an Optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established. The information in the "detailed abstract connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or

available bandwidth) in the "detailed abstract connectivity matrix" while not changing the feasibility of the path.

"Connectivity matrix" is sometimes confused with optical reach table that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools, and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from src to dest, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though "optical reachability table" is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

There is therefore the need to keep the information in the "connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the Orchestrator's PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the Orchestrator's PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed abstract connectivity matrix" that provides accurate, scalable and updated information to allow the Orchestrator's PCE to take optimal decisions by its own.

If the information in the "detailed abstract connectivity matrix" is not complete/accurate, we can have the following drawbacks considering for example the case in Figure 6:

- o If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the Orchestrator's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;

- o If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the Orchestrator's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the Optical domain while the optimal path would actually be the one going through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

Instead, using the approach proposed in this document, the Orchestrator, when it needs to setup an end-to-end path, it can request the Optical domain controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- o When setting up a 5 Gb/s path between routers R1 and R2, the Optical domain controller may report only the VP1-VP4 path as the only feasible path: the Orchestrator can successfully setup the end-to-end path passing through this Optical path;
- o When setting up a 1 Gb/s path between routers R1 and R2, the Optical domain controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Orchestrator can then compute the optimal path which is passing through the VP1-VP4 sub-path (with cost 50) within the Optical domain.

3.2. TE Topology Abstraction

Using the TE Topology model, as defined in [TE-TOP0], the underlying SDN controller can export an abstract TE Topology, composed by a set of TE nodes and TE links, which are abstracting the topology controlled by each domain controller.

Considering the example in Figure 4, the TE domain controller 1 can export a TE Topology encompassing the TE nodes A, B, C and D and the TE Link interconnecting them. In a similar way, TE domain controller 2 can export a TE Topology encompassing the TE nodes E, F, G and H and the TE Link interconnecting them.

In this example, for simplicity reasons, each abstract TE node maps with each physical node, but this is not necessary.

In order to setup a multi-domain TE path (e.g., between nodes A and H), the Orchestrator can compute by its own an optimal end-to-end path based on the abstract TE topology information provided by the domain controllers. For example:

- o Orchestrator's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H, and then request the TE domain controllers to setup the A-B-C and E-G-H intra-domain paths
- o But, during path setup, the domain controller may find out that A-B-C intra-domain path is not feasible (as discussed in section 2.2, in optical networks it is typical to have some paths not being feasible due to optical constraints that are known only by the optical domain controller), while only the path A-B-D is feasible
- o So what the hierarchical controller computed is not good and need to re-start the path computation from scratch

As discussed in section 3.1, providing more extensive abstract information from the TE domain controllers to the multi-domain Orchestrator may lead to scalability problems.

In a sense this is similar to the problem of routing and wavelength assignment within an Optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for Optical path setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (step 1) and then compute an intra-domain path within each Optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path that performing per-domain path computation and then stitch them.

3.3. Complementary use of TE topology and path computation

As discussed in section 2.2, there are some scalability issues with path computation requests in a multi-domain TE network with many TE domains, in terms of the number of requests to send to the TE domain controllers. It would therefore be worthwhile using the TE topology information provided by the domain controllers to limit the number of requests.

An example can be described considering the multi-domain abstract topology shown in Figure 7. In this example, an end-to-end TE path between domains A and F needs to be setup. The transit domain should be selected between domains B, C, D and E.

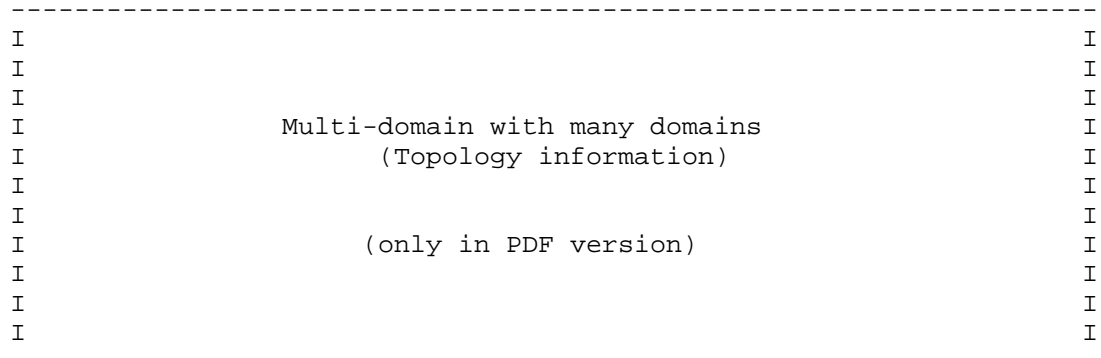


Figure 7 - Multi-domain with many domains (Topology information)

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Orchestrator only knows, from the TE topology provided by the underlying domain controllers, the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Orchestrator can understand by its own that:

- o Domain B cannot be selected as the path connecting domains A and E is not feasible;
- o Domain E cannot be selected as a transit domain since it is known from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-E-F (which is 80, in the worst case)

Therefore, the Orchestrator can understand by its own that the optimal multi-domain path could be either A-D-F or A-E-F but it cannot know which one of the two possible options actually provides the optimal end-to-end path.

The Orchestrator can therefore request path computation only to the TE domain controllers A, D, E and F (and not to all the possible TE domain controllers).

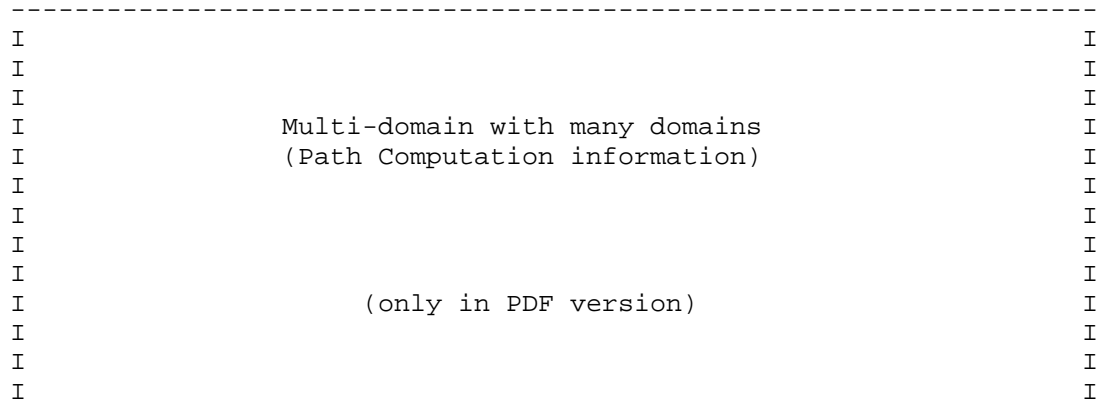


Figure 8 - Multi-domain with many domains (Path Computation information)

Based on these requests, the Orchestrator can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 8, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

4. Motivation for a YANG Model

4.1. Benefits of common data models

Path computation requests should be closely aligned with the YANG data models that provide (abstract) TE topology information, i.e., [TE-TOPO] as well as that are used to configure and manage TE Tunnels, i.e., [TE-TUNNEL]. Otherwise, an error-prone mapping or correlation of information would be required. For instance, there is benefit in using the same endpoint identifiers in path computation requests and in the topology modeling. Also, the attributes used in path computation constraints could use the same or similar data models. As a result, there are many benefits in aligning path computation requests with YANG models for TE topology information and TE Tunnels configuration and management.

4.2. Benefits of a single interface

A typical use case for path computation requests is the interface between an orchestrator and a domain controller. The system integration effort is typically lower if a single, consistent interface is used between such systems, i.e., one data modeling language (i.e., YANG) and a common protocol (e.g., NETCONF or RESTCONF).

Practical benefits of using a single, consistent interface include:

1. Simple authentication and authorization: The interface between different components has to be secured. If different protocols have different security mechanisms, ensuring a common access control model may result in overhead. For instance, there may be a need to deal with different security mechanisms, e.g., different credentials or keys. This can result in increased integration effort.
2. Consistency: Keeping data consistent over multiple different interfaces or protocols is not trivial. For instance, the sequence of actions can matter in certain use cases, or transaction semantics could be desired. While ensuring consistency within one protocol can already be challenging, it is typically cumbersome to achieve that across different protocols.
3. Testing: System integration requires comprehensive testing, including corner cases. The more different technologies are involved, the more difficult it is to run comprehensive test cases and ensure proper integration.
4. Middle-box friendliness: Provider and consumer of path computation requests may be located in different networks, and middle-boxes such as firewalls, NATs, or load balancers may be deployed. In such environments it is simpler to deploy a single protocol. Also, it may be easier to debug connectivity problems.
5. Tooling reuse: Implementers may want to implement path computation requests with tools and libraries that already exist in controllers and/or orchestrators, e.g., leveraging the rapidly growing eco-system for YANG tooling.

4.3. Extensibility

Path computation is only a subset of the typical functionality of a controller. In many use cases, issuing path computation requests comes along with the need to access other functionality on the same

system. In addition to obtaining TE topology, for instance also configuration of services (setup/modification/deletion) may be required, as well as:

1. Receiving notifications for topology changes as well as integration with fault management
2. Performance management such as retrieving monitoring and telemetry data
3. Service assurance, e.g., by triggering OAM functionality
4. Other fulfilment and provisioning actions beyond tunnels and services, such as changing QoS configurations

YANG is a very extensible and flexible data modeling language that can be used for all these use cases.

Adding support for path computation requests to YANG models would seamlessly complement with [TE-TOPO] and [TE-TUNNEL] in the use cases where YANG-based protocols (e.g., NETCONF or RESTCONF) are used.

5. Path Computation for multiple LSPs

There are use cases, where path computation is required for multiple Traffic Engineering Label Switched Paths (TE LSPs) through a network or through a network domain. It may be advantageous to request the new paths for a set of LSPs in one single path computation request [RFC5440] that also includes information regarding the desired objective function, see [RFC5541].

In the context of abstraction and control of TE networks (ACTN), as defined in [ACTN-Frame], when a MDSC receives a virtual network (VN) request from a CNC, the MDSC needs to perform path computation for multiple LSPs as a typical VN is constructed by a set of multiple paths also called end-to-end tunnels. The MDSC may send a single path computation request to the PNC for multiple LSPs, i.e. between the VN end points (access points in ACTN terminology).

In a more general context, when a MDSC needs to send multiple path provisioning requests to the PNC, the MDSC may also group these path provisioning requests together and send them in a single message to the PNC instead of sending separate requests for each path.

6. YANG Model for requesting Path Computation

The TE Tunnel YANG model has been extended to support the need to request path computation.

It is possible to request path computation by configuring a "compute-only" TE tunnel and retrieving the computed path(s) in the LSP(s) Record-Route Object (RRO) list as described in section 3.3.1 of [TE-TUNNEL].

This is a stateful solution since the state of each created "compute-only" TE tunnel needs to be maintained and updated, when underlying network conditions change.

The need also for a stateless solution, based on an RPC, has been recognized, as outlined in section 6.1.

A proposal for a stateless RPC to request path computation is provided in section 6.2.

6.1. Stateless and Stateful Path Computation

It is very useful to provide options for both stateless and stateful path computation mechanisms. It is suggested to use stateless mechanisms as much as possible and to rely on stateful path computation when really needed.

Stateless RPC allows requesting path computation using a simple atomic operation and it is the natural option/choice, especially with stateless PCE.

Since the operation is stateless, there is no guarantee that the returned path would still be available when path setup is requested: this is not a major issue in case the time between path computation and path setup is short.

The RPC response must be provided synchronously and, if collaborative computations are time consuming, it may not be possible to immediate reply to client.

In this case, the client can define a maximum time it can wait for the reply, such that if the computation does not complete in time, the server will abort the path computation and reply to the client with an error. It may be possible that the server has tighter timing

constraints than the client: in this case the path computation is aborted earlier than the time specified by the client.

Note - The RPC response issue (slow RPC server) is not specific to the path computation RPC case so, it may be worthwhile, evaluating whether a more generic solution applicable to any YANG RPC can be used instead.

In case the stateless solution is not sufficient, a stateful solution, based on "compute-only" TE tunnel, could be used to support asynchronous operations and/or to get notifications in case the computed path has been changed.

It is worth noting that also the stateful solution, although increasing the likelihood that the computed path is available at path setup, it does not guaranteed that because notifications may not be reliable or delivered on time.

The stateful path computation has also the following drawbacks:

- o Several messages required for any path computation
- o Requires persistent storage in the provider controller
- o Need for garbage collection for stranded paths
- o Process burden to detect changes on the computed paths in order to provide notifications update

6.2. YANG model for stateless TE path computation

6.2.1. YANG Tree

Figure 9 below shows the tree diagram of the YANG model defined in module ietf-te-path-computation.yang.

```

module: ietf-te-path-computation
  +--rw paths
  |   +--ro path* [path-id]
  |   |   +--ro _telink* [link-ref]
  |   |   |   +--ro link-ref ->
  |   |   |   /nd:networks/network[nd:network-id=current()/../network-
  |   |   |   ref]/lnk:link/link-id

```

```

|      |  +--ro network-ref?  -> /nd:networks/network/network-id
+--ro path-constraints
|  +--ro path-metric-bound* [metric-type]
|  |  +--ro metric-type      identityref
|  |  +--ro upper-bound?    uint64
+--ro topology-id?          te-types:te-topology-id
+--ro ignore-overload?      boolean
+--ro bandwidth-generic
|  +--ro te-bandwidth
|  |  +--ro (technology)?
|  |  |  +--:(psc)
|  |  |  |  +--ro psc?      rt-types:bandwidth-ieee-
float32
|  |  |  |  +--:(otn)
|  |  |  |  |  +--ro otn* [rate-type]
|  |  |  |  |  |  +--ro rate-type      identityref
|  |  |  |  |  |  +--ro counter?      uint16
|  |  |  |  +--:(lsc)
|  |  |  |  |  +--ro wdm* [spectrum slot]
|  |  |  |  |  |  +--ro spectrum      identityref
|  |  |  |  |  |  +--ro slot          int16
|  |  |  |  |  |  +--ro width?       uint16
|  |  |  |  +--:(generic)
|  |  |  |  |  +--ro generic?      te-bandwidth
disjointness
+--ro disjointness?          te-types:te-path-
+--ro setup-priority?        uint8
+--ro hold-priority?         uint8
+--ro signaling-type?        identityref
+--ro path-affinities
|  +--ro constraint* [usage]
|  |  +--ro usage      identityref
|  |  +--ro value?     admin-groups
+--ro path-srlgs
|  +--ro usage?      identityref
|  +--ro values*     srlg
+--ro path-id        yang-types:uuid
+--ro pathComputationService
+--ro _path-ref*      -> /paths/path/path-id

```

```

+--ro _servicePort
|   +--ro source?          inet:ip-address
|   +--ro destination?     inet:ip-address
|   +--ro src-tp-id?       binary
|   +--ro dst-tp-id?       binary
|   +--ro bidirectional
|       +--ro association
|           +--ro id?          uint16
|           +--ro source?      inet:ip-address
|           +--ro global-source? inet:ip-address
|           +--ro type?        identityref
|           +--ro provisioing? identityref
+--ro path-constraints
|   +--ro path-metric-bound* [metric-type]
|       |   +--ro metric-type    identityref
|       |   +--ro upper-bound?   uint64
|   +--ro topology-id?       te-types:te-topology-id
|   +--ro ignore-overload?    boolean
|   +--ro bandwidth-generic
|       |   +--ro te-bandwidth
|       |       +--ro (technology)?
|       |           +--:(psc)
|       |               |   +--ro psc?          rt-types:bandwidth-ieee-
float32 |
|       |   +--:(otn)
|       |       |   +--ro otn* [rate-type]
|       |       |       +--ro rate-type    identityref
|       |       |       +--ro counter?     uint16
|       |   +--:(lsc)
|       |       |   +--ro wdm* [spectrum slot]
|       |       |       +--ro spectrum    identityref
|       |       |       +--ro slot        int16
|       |       |       +--ro width?      uint16
|       |   +--:(generic)
|       |       +--ro generic?    te-bandwidth
+--ro disjointness?          te-types:te-path-disjointness
+--ro setup-priority?        uint8
+--ro hold-priority?         uint8
+--ro signaling-type?        identityref

```

```

|   +--ro path-affinities
|   |   +--ro constraint* [usage]
|   |   |   +--ro usage      identityref
|   |   |   +--ro value?    admin-groups
|   +--ro path-srlgs
|   |   +--ro usage?      identityref
|   |   +--ro values*    srlg
+--ro optimizations
|   +--ro (algorithm)?
|   |   +--:(metric) {path-optimization-metric}?
|   |   |   +--ro optimization-metric* [metric-type]
|   |   |   |   +--ro metric-type      identityref
|   |   |   |   +--ro weight?          uint8
|   |   |   +--ro tiebreakers
|   |   |   |   +--ro tiebreaker* [tiebreaker-type]
|   |   |   |   +--ro tiebreaker-type identityref
|   |   +--:(objective-function) {path-optimization-objective-
function}?
|   |   |   +--ro objective-function
|   |   |   |   +--ro objective-function-type? identityref
augment /te:tunnels-rpc/te:input/te:tunnel-info:
+---- request-list* [request-id-number]
|   +---- request-id-number      uint32
|   +---- servicePort*
|   |   +---- source?            inet:ip-address
|   |   +---- destination?       inet:ip-address
|   |   +---- src-tp-id?         binary
|   |   +---- dst-tp-id?         binary
|   |   +---- bidirectional
|   |   |   +---- association
|   |   |   |   +---- id?          uint16
|   |   |   |   +---- source?      inet:ip-address
|   |   |   |   +---- global-source? inet:ip-address
|   |   |   |   +---- type?        identityref
|   |   |   |   +---- provisioing? identityref
+---- path-constraints
|   +---- path-metric-bound* [metric-type]
|   |   +---- metric-type      identityref
|   |   +---- upper-bound?    uint64

```

```

| | +---- topology-id?          te-types:te-topology-id
| | +---- ignore-overload?      boolean
| | +---- bandwidth-generic
| | |   +---- te-bandwidth
| | |   |   +---- (technology)?
| | |   |   |   +---:(psc)
| | |   |   |   |   +---- psc?          rt-types:bandwidth-ieee-
float32
| | |   |   |   |   +---:(otn)
| | |   |   |   |   |   +---- otn* [rate-type]
| | |   |   |   |   |   |   +---- rate-type      identityref
| | |   |   |   |   |   |   +---- counter?      uint16
| | |   |   |   |   |   +---:(lsc)
| | |   |   |   |   |   |   +---- wdm* [spectrum slot]
| | |   |   |   |   |   |   |   +---- spectrum    identityref
| | |   |   |   |   |   |   |   +---- slot        int16
| | |   |   |   |   |   |   |   +---- width?      uint16
| | |   |   |   |   |   +---:(generic)
| | |   |   |   |   |   |   +---- generic?      te-bandwidth
| | |   |   |   +---- disjointness?      te-types:te-path-disjointness
| | |   |   +---- setup-priority?      uint8
| | |   |   +---- hold-priority?      uint8
| | |   |   +---- signaling-type?      identityref
| | |   +---- path-affinities
| | |   |   +---- constraint* [usage]
| | |   |   |   +---- usage      identityref
| | |   |   |   +---- value?    admin-groups
| | |   +---- path-srlgs
| | |   |   +---- usage?      identityref
| | |   |   +---- values*    srlg
+---- optimizations
|   +---- (algorithm)?
|   |   +---:(metric) {path-optimization-metric}?
|   |   |   +---- optimization-metric* [metric-type]
|   |   |   |   +---- metric-type      identityref
|   |   |   |   +---- weight?          uint8
|   |   +---- tiebreakers
|   |   |   +---- tiebreaker* [tiebreaker-type]
|   |   |   |   +---- tiebreaker-type  identityref

```



```

|          +---:(objective-function) {path-optimization-objective-
function}?
|          +----- objective-function
|          |          +----- objective-function-type?  identityref
+----- synchronization* [synchronization-index]
|          +----- synchronization-index      uint32
+----- svec
|          +----- relaxable?                  boolean
|          +----- link-diverse?              boolean
|          +----- node-diverse?              boolean
|          +----- srlg-diverse?              boolean
|          +----- request-id-number*         uint32
+----- path-constraints
|          +----- path-metric-bound* [metric-type]
|          |          +----- metric-type      identityref
|          |          +----- upper-bound?    uint64
+----- topology-id?      te-types:te-topology-id
+----- ignore-overload?  boolean
+----- bandwidth-generic
|          +----- te-bandwidth
|          |          +----- (technology)?
|          |          |          +---:(psc)
|          |          |          |          +----- psc?      rt-types:bandwidth-ieee-
float32
|          |          |          |          |
|          |          |          |          +---:(otn)
|          |          |          |          |          +----- otn* [rate-type]
|          |          |          |          |          |          +----- rate-type  identityref
|          |          |          |          |          |          +----- counter?   uint16
|          |          |          |          |          +---:(lsc)
|          |          |          |          |          |          +----- wdm* [spectrum slot]
|          |          |          |          |          |          |          +----- spectrum  identityref
|          |          |          |          |          |          |          +----- slot      int16
|          |          |          |          |          |          |          +----- width?   uint16
|          |          |          |          |          +---:(generic)
|          |          |          |          |          |          +----- generic?   te-bandwidth
+----- disjointness?      te-types:te-path-disjointness
+----- setup-priority?    uint8
+----- hold-priority?     uint8
+----- signaling-type?    identityref

```

```

+---- path-affinities
|   +---- constraint* [usage]
|       +---- usage      identityref
|       +---- value?     admin-groups
+---- path-srlgs
|   +---- usage?         identityref
|   +---- values*        srlg
augment /te:tunnels-rpc/te:output/te:result:
+--ro response* [response-index]
+--ro response-index      uint32
+--ro (response-type)?
+--:(no-path-case)
|   +--ro no-path
+--:(path-case)
+--ro pathCompService
+--ro _path-ref*           -> /paths/path/path-id
+--ro _servicePort
|   +--ro source?          inet:ip-address
|   +--ro destination?     inet:ip-address
|   +--ro src-tp-id?       binary
|   +--ro dst-tp-id?       binary
|   +--ro bidirectional
|       +--ro association
|           +--ro id?              uint16
|           +--ro source?          inet:ip-address
|           +--ro global-source?   inet:ip-address
|           +--ro type?            identityref
|           +--ro provisioing?     identityref
+--ro path-constraints
|   +--ro path-metric-bound* [metric-type]
|       |   +--ro metric-type      identityref
|       |   +--ro upper-bound?     uint64
|       +--ro topology-id?         te-types:te-topology-
id
|   +--ro ignore-overload?         boolean
+--ro bandwidth-generic
|   +--ro te-bandwidth
|       +--ro (technology)?
|       +--:(psc)

```

```

| | | | +--ro psc?          rt-types:bandwidth-
ieee-float32
| | | | +--:(otn)
| | | | | +--ro otn* [rate-type]
| | | | | | +--ro rate-type      identityref
| | | | | | +--ro counter?     uint16
| | | | +--:(lsc)
| | | | | +--ro wdm* [spectrum slot]
| | | | | | +--ro spectrum      identityref
| | | | | | +--ro slot         int16
| | | | | | +--ro width?       uint16
| | | | +--:(generic)
| | | | | +--ro generic?       te-bandwidth
disjointness
| | | | +--ro disjointness?    te-types:te-path-
| | | | +--ro setup-priority?   uint8
| | | | +--ro hold-priority?    uint8
| | | | +--ro signaling-type?   identityref
| | | | +--ro path-affinities
| | | | | +--ro constraint* [usage]
| | | | | | +--ro usage        identityref
| | | | | | +--ro value?      admin-groups
| | | | +--ro path-srlgs
| | | | | +--ro usage?         identityref
| | | | | +--ro values*       srlg
+--ro optimizations
+--ro (algorithm)?
+--:(metric) {path-optimization-metric}?
| | +--ro optimization-metric* [metric-type]
| | | +--ro metric-type      identityref
| | | +--ro weight?         uint8
| | +--ro tiebreakers
| | | +--ro tiebreaker* [tiebreaker-type]
| | | | +--ro tiebreaker-type identityref
+--:(objective-function) {path-optimization-
objective-function}?
+--ro objective-function
+--ro objective-function-type?
identityref
```

Figure 9 - TE path computation tree

6.2.2. YANG Module

```
<CODE BEGINS>file " ietf-te-path-computation.yang "  
module ietf-te-path-computation {  
  yang-version 1.1;  
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";  
  // replace with IANA namespace when assigned  
  
  prefix "tepc";  
  
  import ietf-inet-types {  
    prefix "inet";  
  }  
  
  import ietf-yang-types {  
    prefix "yang-types";  
  }  
  
  import ietf-network-topology {  
    prefix "nt";  
  }  
  
  import ietf-te {  
    prefix "te";  
  }  
  
  import ietf-te-types {  
    prefix "te-types";  
  }  
  
  organization  
    "Traffic Engineering Architecture and Signaling (TEAS)  
    Working Group";  
  
  contact  
    "WG Web:    <http://tools.ietf.org/wg/teas/>  
    WG List:    <mailto:teas@ietf.org>
```

```
WG Chair: Lou Berger
          <mailto:lberger@labn.net>

WG Chair: Vishnu Pavan Beeram
          <mailto:vbeeram@juniper.net>

";

description "YANG model for stateless TE path computation";

revision "2016-10-10" {
  description "Initial revision";
  reference "YANG model for stateless TE path computation";
}

/*
 * Features
 */

feature stateless-path-computation {
  description
    "This feature indicates that the system supports
    stateless path computation.";
}

/*
 * Groupings
 */

grouping Path {
  list _telink {
    key 'link-ref';
    config false;
    uses nt:link-ref;
    description "List of telink refs.";
  }
  uses te-types:generic-path-constraints;
```

```
    leaf path-id {
      type yang-types:uuid;
      config false;
      description "path-id ref.";
    }
    description "Path is described by an ordered list of TE Links.";
  }

  grouping PathCompServicePort {
    leaf source {
      type inet:ip-address;
      description "TE tunnel source address.";
    }
    leaf destination {
      type inet:ip-address;
      description "P2P tunnel destination address";
    }
    leaf src-tp-id {
      type binary;
      description "TE tunnel source termination point identifier.";
    }
    leaf dst-tp-id {
      type binary;
      description "TE tunnel destination termination point
identifier.";
    }
    uses te:bidir-assoc-properties;
    description "Path Computation Service Port grouping.";
  }

  grouping PathComputationService {
    leaf-list _path-ref {
      type leafref {
        path '/paths/path/path-id';
      }
      config false;
      description "List of previously computed path references.";
    }
  }
```

```
    container _servicePort {
        uses PathCompServicePort;
        description "Path Computation Service Port.";
    }
    uses te-types:generic-path-constraints;
    uses te-types:generic-path-optimization;

    description "Path computation service.";
}

grouping synchronization-info {
    description "Information for sync";
    list synchronization {
        key "synchronization-index";
        description "sync list";
        leaf synchronization-index {
            type uint32;
            description "index";
        }
        container svec {
            description
                "Synchronization VECtor";
            leaf relaxable {
                type boolean;
                default true;
                description
                    "If this leaf is true, path computation process is free
to ignore svec content.
                    otherwise it must take into account this svec.";
            }
            leaf link-diverse {
                type boolean;
                default false;
                description "link-diverse";
            }
            leaf node-diverse {
```

```
        type boolean;
        default false;
        description "node-diverse";
    }
    leaf srlg-diverse {
        type boolean;
        default false;
        description "srlg-diverse";
    }
    leaf-list request-id-number {
        type uint32;
        description
            "This list reports the set of M path computation requests
that must be synchronized.";
    }
    }
    uses te-types:generic-path-constraints;
}

grouping no-path-info {
    description "no-path-info";
    container no-path {
        description "no-path container";
    }
}

/*
 * Root container
 */
container paths {
    list path {
        key "path-id";
        config false;
        uses Path;

        description "List of previous computed paths.";
    }
    description "Root container for path-computation";
}
```



```
}

container pathComputationService {
  config false;
  uses PathComputationService;
  description "Service for computing paths.";
}

/**
 * AUGMENTS TO TE RPC
 */

augment "/te:tunnels-rpc/te:input/te:tunnel-info" {
  description "statelessComputeP2PPath input";
  list request-list {
    key "request-id-number";
    description "request-list";
    leaf request-id-number {
      type uint32;
      mandatory true;
      description "Each path computation request is uniquely
identified by the request-id-number.
      It must be present also in rpcs.";
    }
    list servicePort {
      min-elements 1;
      uses PathCompServicePort;
      description "List of service ports.";
    }
    uses te-types:generic-path-constraints;
    uses te-types:generic-path-optimization;
  }
  uses synchronization-info;
}

augment "/te:tunnels-rpc/te:output/te:result" {
```

<CODE ENDS>

Figure 10 - TE path computation YANG module

7. Security Considerations

This document describes use cases of requesting Path Computation using YANG models, which could be used at the ABNO Control Interface [RFC7491] and/or between controllers in ACTN [ACTN-frame]. As such, it does not introduce any new security considerations compared to the ones related to YANG specification, ABNO specification and ACTN Framework defined in [RFC6020], [RFC7950], [RFC7491] and [ACTN-frame].

This document also defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC6020] apply for this document as well.

8. IANA Considerations

This section is for further study: to be completed when the YANG model is more stable.

9. References

9.1. Normative References

- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC7139] Zhang, F. et al., "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.
- [RFC7491] Farrel, A., King, D., "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, March 2015.
- [RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.
- [RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [TE-TUNNEL] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [ACTN-Frame] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of Traffic Engineered Networks" draft-ietf-actn-framework, work in progress.

[ITU-T G.709-2016] ITU-T Recommendation G.709 (06/16), "Interface for the optical transport network", June 2016

9.2. Informative References

[RFC5541] Le Roux, JL. et al., " Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

[RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.

[OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.

[ACTN-Info] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., "Information Model for Abstraction and Control of Transport Networks", draft-leebelotti-actn-info, work in progress.

[PCEP-Service-Aware] Dhody, D. et al., " Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", draft-ietf-pce-pcep-service-aware, work in progress.

10. Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in discussions and providing valuable insights.

The authors would like to thank the authors of the TE Tunnel YANG model [TE-TUNNEL], in particular Igor Bryskin, Vishnu Pavan Beeram, Tarek Saad and Xufeng Liu, for their inputs to the discussions and support in having consistency between the Path Computation and TE Tunnel YANG models.

This document was prepared using 2-Word-v2.0.template.dot.

Contributors

Dieter Beller
Nokia
Email: dieter.beller@nokia.com

Gianmarco Bruno
Ericsson
Email: gianmarco.bruno@ericsson.com

Francesco Lazzeri
Ericsson
Email: francesco.lazzeri@ericsson.com

Young Lee
Huawei
Email: leeyoung@huawei.com

Carlo Perocchio
Ericsson
Email: carlo.perocchio@ericsson.com

Authors' Addresses

Italo Busi (Editor)
Huawei
Email: italo.busi@huawei.com

Sergio Belotti (Editor)
Nokia
Email: sergio.belotti@nokia.com

Victor Lopez
Telefonica
Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma
Infinera
Email: AnSharma@infinera.com

Yan Shi
China Unicom
Email: shiyan49@chinaunicom.cn

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

Karthik Sethuraman
NEC
Email: karthik.sethuraman@necam.com

Michael Scharf
Nokia
Email: michael.scharf@nokia.com

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com

MPLS WG
Internet-Draft
Intended status: Standards Track
Expires: March 12, 2018

A. Deshmukh
K. Kompella
Juniper Networks, Inc.
September 8, 2017

RSVP Extensions for RMR
draft-deshmukh-mpls-rsvp-rmr-extension-01

Abstract

Rings are the most common topology in access and aggregation networks. However, the use of MPLS as the transport protocol for rings is very limited today. draft-ietf-mpls-rmr-02 describes a mechanism to handle rings efficiently using MPLS. This document describes the extensions to the RSVP protocol for signaling MPLS label switched paths in rings.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 12, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. RSVP Extensions	4
3.1. Session Object	4
3.2. SENDER_TEMPLATE, FILTER_SPEC Objects	5
4. Ring Signaling Procedures	5
4.1. Differences from regular RSVP-TE LSPs	5
4.2. LSP signaling	5
4.2.1. Path Propagation for RMR	7
4.2.2. Resv Processing for RMR	8
4.3. Protection	9
4.4. Ring changes	10
4.5. Bandwidth management	11
5. Security Considerations	13
6. Contributors	13
7. IANA Considerations	13
8. References	13
8.1. Normative References	13
8.2. Informative References	14
Authors' Addresses	14

1. Introduction

This document extends RSVP-TE [RFC3209] to establish label-switched path (LSP) tunnels in the ring topology. Rings are auto-discovered using the mechanisms mentioned in the [draft-ietf-mppls-rmr-02]. Either IS-IS [RFC5305] or OSPF[RFC3630] can be used as the IGP for auto-discovering the rings.

After the rings are auto-discovered, each ring node knows its clockwise (CW) and anti-clockwise (AC) ring neighbors and its ring links. All of the express links in the ring also get identified as part of the auto-discovery process. At this point, every node in the ring informs the RSVP protocol to begin the signaling of the ring LSPs.

Section 2 covers the terminology used in this document. Section 3 presents the RSVP protocol extensions needed to support MPLS rings. Section 4 describes the procedures of RSVP LSP signaling in detail.

2. Terminology

A ring consists of a subset of n nodes $\{R_i, 0 \leq i < n\}$. We define the direction from node R_i to R_{i+1} as "clockwise" (CW) and the reverse direction as "anti-clockwise" (AC). As there may be several rings in a graph, we number each ring with a distinct ring ID RID.

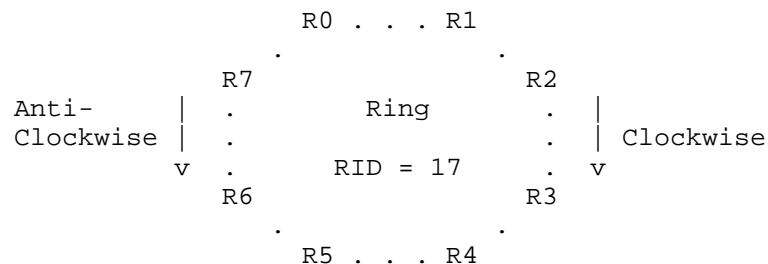


Figure 1: Ring with 8 nodes

The following terminology is used for ring LSPs:

Ring ID (RID): A non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network. A node may belong to multiple rings.

Ring node: A member of a ring. Note that a device may belong to several rings.

Node index: A logical numbering of nodes in a ring, from zero upto one less than the ring size. Used purely for exposition in this document.

Ring neighbors: Nodes whose indices differ by one (modulo ring size).

Ring links: Links that connect ring neighbors.

Express links: Links that connect non-neighboring ring nodes.

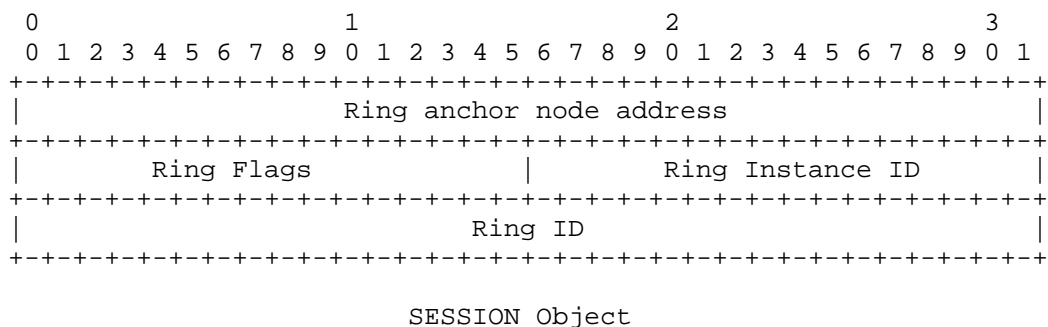
MP2P LSP: Each LSP in the ring is a multipoint to point LSP such that LSP can have multiple ingress nodes and one egress node.

3. RSVP Extensions

Due to the new ring LSP semantics, the signaling-message identification of ring LSPs will be different than the regular RSVP LSPs. So, a new C-Type is defined here for the SESSION object. This new C-Type will help to clearly differentiate ring LSPs from regular LSPs. In addition, new flags are introduced in the SESSION object to represent the ring direction of the corresponding Path message.

3.1. Session Object

Class = SESSION, LSP_TUNNEL_IPv4 C-Type = TBD



Ring anchor node address: IPv4 address of the anchor node. Each anchor node creates a LSP addressed to itself.

Ring Instance ID: A 16-bit identifier used in the SESSION. This Ring Instance ID is useful for graceful ring changes. If a new node is being added to the ring or some existing node goes down and we have to signal a smaller ring, in those cases, anchor node creates a new tunnel with a different Ring Instance ID.

Ring ID: A 32-bit number that identifies a ring; this is unique in some scope of a Service Provider's network. This number remains constant throughout the existence of ring.

Ring Flags: For each ring, the anchor node starts signaling of a ring LSP. Ring LSP RL_i, anchored on node R_i, consists of two counter-rotating unicast LSPs that start and end at R_i. One LSP will be in the clockwise direction and other LSP will be in the anti-clockwise direction. A ring LSP is "multipoint": any node R_j can use RL_i to send traffic to R_i; this can be in either the CW or AC directions, or both (i.e., load balanced). Two new flags are defined in the SESSION object which define the ring direction of the corresponding Path message.

ClockWise(CW) Direction 0x01: This flag indicates that the corresponding Path message is traveling in the ClockWise(CW) direction along the ring.

Anti-ClockWise(AC) Direction 0x02: This flag indicates that the corresponding Path message is traveling in the Anti-ClockWise(AC) direction along the ring.

3.2. SENDER_TEMPLATE, FILTER_SPEC Objects

There will be no changes to the SENDER_TEMPLATE and FILTER_SPEC objects. The format of the above 2 objects will be similar to the definitions in RFC 3209. [RFC3209] Only the semantics of these objects will slightly change. This will be explained in section Section 4.5 below.

4. Ring Signaling Procedures

A ring node indicates in its IGP updates the ring LSP signaling protocols that it supports. This can be LDP and/or RSVP-TE. Ideally, each node should support both. If the ring is configured with RSVP as the signaling protocol, then once a ring node R_i knows the RID, its ring links and directions, it kicks off ring RSVP LSP signaling automatically.

4.1. Differences from regular RSVP-TE LSPs

Ring LSPs differ from regular RSVP-TE LSPs in several ways:

1. Ring LSPs (by construction) form a loop.
2. Ring LSPs are multipoint-to-point. Any ring node can inject traffic into a ring LSP.
3. The bandwidth of a ring LSP can change hop-to-hop.
4. Ring LSPs are protected without the use of bypass or detour LSPs. Ring LSP protection is akin to SONET/SDH ring protection.

4.2. LSP signaling

After the ring auto-discovery process, each anchor node creates a LSP addressed to itself. This ring LSP contains a pair of counter-rotating unicast LSPs. So, for a ring containing N nodes, there will be 2N total LSPs signaled.

There is no need for ERO object in the Path message. The Path message for ring LSPs has the following format:

```

<Path Message> ::= <Common Header> [ <INTEGRITY> ]
                        <SESSION> <RSVP_HOP>
                        <TIME_VALUES>
                        <LABEL_REQUEST>
                        [ <SESSION_ATTRIBUTE> ]
                        <sender descriptor list>

<sender descriptor list> ::= <sender descriptor>|
                                <sender descriptor list> <sender descri
ptor>

<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>

```

The anchor node creates 2 Path messages traveling in opposite directions. The SESSION format MUST be as per the description in Section 3.1. The anchor node which creates the LSP will insert it's own address in the "Ring node anchor address" field of the SESSION object. So effectively, the Path messages are addressed to the originating node itself.

The SESSION flags of these 2 Path messages are different. The Path message sent to the CW neighbor MUST have the CW flag set in the SESSION object to signal the LSP going in the clockwise direction. The Path message sent to the AC neighbor MUST have the AC flag set to signal the LSP in the anti-clockwise direction. The details for signaling over express links will be given in a future version.

When an incoming Path message is received at the ring node R_i, it consults the results of auto-discovery to find the appropriate ring neighbor. If the incoming Path message has CW direction flag set, then R_i sends a Path message to its CW ring neighbor (and vice versa) after including its own SENDER_DESCRIPTOR in the path message. Thus, there is no need of ERO in the Path message. The Path message is routed locally at each ring based on the ring auto-discovery calculations.

The RESV message for ring LSPs also uses the new RING_IPv4 SESSION object. When the Path message originated from the anchor node R_i reaches back to R_i, R_i generates a Resv message. Note that this means that anchor node is both Ingress and Egress for the Path message. The Resv message copies the same ring flags as received in the corresponding Path message. So, a Resv message for a CW LSP goes in the AC direction (unlike the Path message, which goes CW). This is done to correctly match Path and corresponding Resv messages at transit ring nodes. Upon receiving Resv message with CW flag set, the ring node will forward the Resv message to its AC neighbor.

Each ring node R_i allocates CW and AC labels for each ring LSP RL_k . As the signaling propagates around the ring, CW and AC labels are exchanged. When R_i receives CW and AC labels for RL_k from its ring neighbors, primary and fast reroute (FRR) paths for RL_k are installed at R_i .

Consider the following three nodes of the ring, and their signaling interactions for LSP RL_5 originating from anchor node R_5 :

```

                P5_CW ->      P5_CW ->
                Q5_CW <-      Q5_CW <-
... ----- R7 ----- R8 ----- R9 ----- ...
                P5_AC <-      P5_AC <-
                Q5_AC ->      Q5_AC ->

```

P corresponds to the Path message and Q corresponds to the Resv message.

As explained above, an RMR LSP consists of two counter-rotating ring LSPs that start and end at the same node, say R_1 . As such, this appears to cause a loop, something that is normally avoided by RSVP-TE. There are some benefits to this:

Having a ring LSP form a loop allows the anchor node R_1 to ping itself and thus verify the end-to-end operation of the LSP. This, in conjunction with link-level OAM, offers a good indication of the operational state of the LSP. Also, having R_1 to be the ingress means that R_1 can initiate the Path messages for the two ring LSPs. This avoids R_1 having to coordinate with its neighbors to signal the LSPs, and simplifies the case where a ring update changes R_1 's ring neighbors. The cost of this is a little more signaling and a couple more label entries in the LFIB. However, we will let implementation guide us to the wisdom of this approach.

4.2.1. Path Propagation for RMR

Ring LSPs are MP2P in nature. It means that every non-egress node is also an ingress and a merge-point for the LSP. Focussing on ring-LSP-0 (i.e ring-LSPs starting at R_0):

```

R0---->R1---->R2---->R3---->R4---->R5---->R6---->R7---->R0(CW LSP)
R0---->R7---->R6---->R5---->R4---->R3---->R2---->R1---->R0(ACW LSP)

```

Each ring node inserts a new SENDER_TEMPLATE object into an incoming Path message. The procedure for that is as follows:

When a ring node R3 receives a Path message initiated by anchor node R0 (for anchor lsp "lsp0"), R3 SHOULD make a copy of the received Path message for "lsp0". R3 then inserts a new sender-template object into the Path message for "lsp0". In the sender-template object, R3 uses the sender address as the loopback address of node R3 and lsp-id = X. R3 then forwards this modified Path message to its ring neighbor.

So at this point, when Path messages head out at R3, there will be 4 different SENDER_TEMPLATE objects in the outgoing Path message for lsp0:

```

-----
| SENDER_TEMPLATE_0 : SENDER_ADDRESS = R0, LSP_ID = 1 |
-----
| SENDER_TEMPLATE_1 : SENDER_ADDRESS = R1, LSP_ID = 1 |
-----
| SENDER_TEMPLATE_2 : SENDER_ADDRESS = R2, LSP_ID = 1 |
-----
| SENDER_TEMPLATE_3 : SENDER_ADDRESS = R3, LSP_ID = 1 |
-----

```

4.2.2. Resv Processing for RMR

When Egress node R0 receives the modified Path message, it replies with a Resv message containing multiple FLOW_DESCRIPTOR objects. There should be 1 FLOW_DESCRIPTOR object corresponding to each of the SENDER_TEMPLATE object in the incoming Path message. The SESSION object of the Resv message will exactly match with the received Path message.

[RFC 3209] already supports receiving a Resv message with multiple flow-descriptors in it, as described in section 3.2 in that document. In each flow-descriptor there is a separate:

- a. FLOW_SPEC object corresponding to the SENDER_TSPEC that was sent in the Path message which could be admitted after admission-control downstream, and
- b. FILTER_SPEC object corresponding to SENDER_TEMPLATE that was sent in the Path message that could be admitted after admission-control downstream.

Each transit node removes the FLOW-DESCRIPTOR corresponding to itself from the Resv message before sending the Resv message upstream.

4.3. Protection

In the rings, there are no protection LSPs -- no node or link bypass LSPs, no standby LSPs and no detours. Protection is via the "other" direction around the ring, which is why ring LSPs are in counter-rotating pairs. Protection works in the same way for link, node and ring LSP failures.

Since each ring LSP is a MP2P LSP, any ring node can inject traffic onto a LSP whose anchor might be a different ring node. To achieve the above, an ingress route will be installed as follows at every ring node J, for a given ring-LSP with anchor Rk (say 1.2.3.4).

```
1.2.3.4  -> (Push CL_J+1,K, NH: R_J+1)      # CW
          -> (Push AL_J-1,K, NH: R_J-1)      # AC

CL = Clockwise label
AL = Anti-Clockwise label
```

Traffic will either be load balanced in the CW and AC directions or the traffic will be sent on just CW or AC lsp based on parameters such as hop-count, policy etc.

Also, 2 transit routes will be installed for the anchor LSP transiting from node Rj as follows:

```
CL_J,K -> SWAP(CL_J+1,K, NH: R_J+1)      #CW
          -> SWAP(AL_J-1,K , NH: R_J-1)    #AC

CL = Clockwise label
AL = Anti-Clockwise label
CW NH has weight 1, AC NH has higher-weight.

AL_J,K -> SWAP(AL_J-1,K , NH: R_J-1)    #AC
          -> SWAP(CL_J+1,K, NH: R_J+1)    #CW

CL = Clockwise label
AL = Anti-Clockwise label
AC NH has weight 1, CW NH has higher weight.
```

Suppose a packet headed in anti-clockwise direction towards R5 and it arrives at node R7. Lets say that now R7 learns there is a link

failure in the AC direction. R7 reroutes this packet back onto the clockwise direction. This reroute action is pre-programmed in the LFIB, to minimize the time between detection of a fault and the corresponding recovery action.

At this time, R7 also sends a notification to R0 that the AC direction is not working. R0 modifies its ingress route(for R5 LSP) by removing the AC direction LSP's route. Thus, R0 switches traffic to the CW direction.

These notification propagate CW until each traffic source on the ring CW of the failure uses the CW direction. For RSVP-TE, this notification is sent in the form of PathErr message.

To provide this notification, the ring node detecting failure SHOULD send a Path Error message with error code of "Notify" and an error value field of ("Tunnel locally repaired"). This Path Error code and value is same as defined in RFC 4090[RFC4090] for the notification of local repair.

Note that the failure of a node or a link will not necessarily affect all ring LSPs. Thus, it is important to identify the affected LSPs and only switch the affected LSPs.

4.4. Ring changes

A ring node can go down resulting in a smaller ring or a new node can be added to the ring which will increase the ring size. In both of the above cases, the ring auto-discovery process SHOULD kick in and it SHOULD calculate a new ring with the changed ring nodes.

When the ring auto-discovery process is complete, IGP will signal RSVP to begin the MBB process for the existing ring LSPs. For this MBB process, the anchor node will create a new Path message with a different Ring Instance ID in the SESSION object. All other fields in the SESSION Object will remain same as the existing Path message(before the ring change).

This new Path message will then propagate along the ring neighbors in the same way as the original Path message. Each ring neighbor SHOULD forward the Path message to its appropriate neighbor based on the new auto-discovery calculations.

For the ring links which are common between the old and new LSPs, the LSPs will share resources(SE style reservation) on those ring links. Note that here we are using Ring Instance ID in the SESSION object to share resources instead of the LSP_ID in the SENDER_TEMPLATE Object(which is used in RSVP-TE for sharing resources as described in

RFC 3209 [RFC4090]). The LSP_ID use is reserved for a different functionality as described in section Section 4.5.

4.5. Bandwidth management

For RSVP-TE LSPs, bandwidths may be signaled in both directions. However, these are not provisioned either; rather, one does "reverse call admission control". When a service needs to use an LSP, the ring node where the traffic enters the ring attempts to increase the bandwidth on the LSP to the egress. If successful, the service is admitted to the ring.

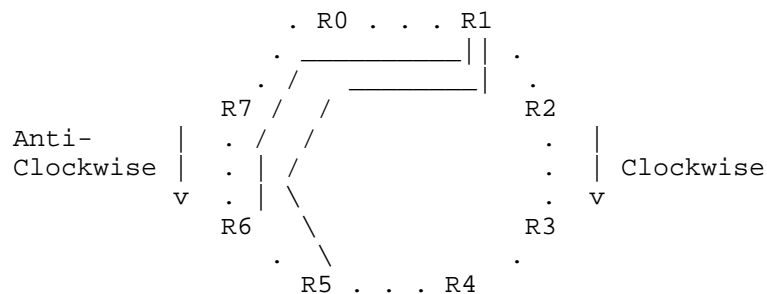


Figure 2: BW Management in Ring with 8 nodes

Let's say that Ring node R5 wants to increase the BW for the LSP whose egress is at node R1. To achieve this BW increase, Ring node R5 has to increase BW along the LSP anchored at node R1 (say lsp1).

R5 makes a copy of the existing ring Path message for lsp1. R5 then modifies the sender-template object from the copied Path message for "lsp1". In the sender-template object, R5 uses the sender address as the loopback address of node R5 and lsp-id = X+1. R5 also modifies the TSPEC object which represents the BW increase/decrease in this new Path message. R5 then forwards this new Path message to its ring neighbor. The original anchor Path message has sender address as loopback address of R1.

Now, let's say, node 5 wants to increase BW again for lsp1, then R5 adds a new SENDER_TEMPLATE object in the existing Path message for "lsp1" with sender address as loopback of node 5 and lsp-id = X+2. So at this point, there will be 2 different SENDER_TEMPLATE objects corresponding to node 5 in the outgoing path message.

```

-----
| SENDER_TEMPLATE_0 : SENDER_ADDRESS = R0, LSP_ID = 1 |
-----
| SENDER_TEMPLATE_1 : SENDER_ADDRESS = R1, LSP_ID = 1 |
-----
| ..... |
-----
| SENDER_TEMPLATE_5 : SENDER_ADDRESS = R5, LSP_ID = 1 |
-----
| SENDER_TEMPLATE_5 : SENDER_ADDRESS = R5, LSP_ID = 2 |
-----

```

Similarly, if node R6 wants to increase the BW for "lsp1", it SHOULD create a new Path message containing SENDER_TEMPLATE object with sender address = loopback of node 6 and lsp-id = Y+1. Thus, it should be noted that each ring-node independently tracks its own lsp-ID that is currently in-use on a given RMR sub-LSP. This lsp-ID value will (could) be different for each ring-node for a given ring sub-LSP.

If sufficient BW is available all the way towards ring node R1, then this new Path message reaches node R1. R1 generates a Resv message with the correct FILTER_SPEC object corresponding to the received SENDER_TEMPLATE object. This Resv message will also have the correct FLOWSPEC object as per the requested bandwidth.

If sufficient BW is not available at some downstream (say node R9), then ring node R9 SHOULD generate a PathErr message with the corresponding Sender Template Object. When node R5 receives this PathErr message, R5 understands that the BW increase was not successful. Note that the existing established bandwidths for lsp1 are not affected by this new PathErr message.

When ring node R5 no longer needs the BW reservation, then ring node R5 SHOULD originate a new Path message with the appropriate Sender Template Object containing 0 BW as described above. Every downstream node SHOULD then remove bandwidth allocated on the corresponding link on receipt of this Path message.

Also, note that as part of this BW increase or decrease process, any ring node does not actually change any label associated with the LSP. So, the label remains same as it was signaled initially when the anchor LSP came up.

5. Security Considerations

It is not anticipated that either the notion of MPLS rings or the extensions to various protocols to support them will cause new security loopholes. As this document is updated, this section will also be updated.

6. Contributors

Ravi Singh
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: ravis@juniper.net

Santosh Esale
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: sesale@juniper.net

Raveendra Torvi
Juniper Networks, Inc.
10 Technology Park Dr
Westford, MA 01886
USA

Email: rtorvi@juniper.net

7. IANA Considerations

Requests to IANA will be made in a future version of this document.

8. References

8.1. Normative References

[I-D.ietf-mpls-rmr]
Kompella, K. and L. Contreras, "Resilient MPLS Rings",
draft-ietf-mpls-rmr-05 (work in progress), July 2017.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

- [I-D.dai-mppls-rsvp-te-mbb-label-reuse] Dai, M. and M. Chaudhry, "MPLS RSVP-TE MBB Label Reuse", draft-dai-mppls-rsvp-te-mbb-label-reuse-01 (work in progress), September 2015.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.

Authors' Addresses

Abhishek Deshmukh
Juniper Networks, Inc.
10 Technology Park Dr
Westford, MA 01886
USA

Email: adeshmukh@juniper.net

Kireeti Kompella
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: kireeti@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: September 14, 2017

G.Galimberti, Ed.
Cisco
R.Kunze
Deutsche Telekom
D. Hiremagalur, Ed.
G. Grammel, Ed.
Juniper
March 13, 2017

A YANG model to manage the optical interface parameters for an external
transponder in a WDM network
draft-dharini-ccamp-dwdm-if-param-yang-01

Abstract

This memo defines a Yang model related to the Optical Transceiver parameters characterising coherent 100G and above interfaces. 100G and above Transceivers support coherent modulation, multiple modulation formats, multiple FEC codes including some not yet specified by ITU-T G.698.2 [ITU.G698.2] or any other ITU-T recommendation. More context about the state of the Coherent transceivers is described in draft-many-coherent-DWDM-if-control. Use cases are described in draft-ietf-ccamp-flexi-grid-fwk

The Yang model defined in this memo can be used for Optical Parameters monitoring and/or configuration of the endpoints of a multi-vendor IaDI optical link.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. The Internet-Standard Management Framework	3
3. Conventions	3
4. Overview	4
4.1. Optical Parameters Description	4
4.1.1. Parameters at Ss	5
4.1.2. Interface at point Rs	5
4.2. Use Cases	5
4.3. Optical Interface for external transponder in a WDM network	5
5. Structure of the Yang Module	8
6. Yang Module	8
7. Security Considerations	21
8. IANA Considerations	21
9. Acknowledgements	22
10. Contributors	22
11. References	23
11.1. Normative References	23
11.2. Informative References	25
Appendix A. Change Log	25
Appendix B. Open Issues	25
Authors' Addresses	26

1. Introduction

This memo defines a Yang model that translates and obsolete the SNMP mib module defined in draft-galikunze-ccamp-dwdm-if-snmp-mib for managing single channel optical interface parameters of DWDM applications, using the approach specified in G.698.2. This model

supports parameters to characterize coherent transceivers found in current implementations to specify the mode of operation. As application identifiers like those specified in ITU-T G.874.1 [ITU.G874.1] are not available we use mode templates instead. A mode template describes transceiver characteristics in detail and can be identified by a mode-id.

This draft refers and supports the draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk and draft-many-coherent-DWDM-if-control.

The YANG model describing and extending the optical parameters allows different vendors and operators to retrieve, provision and exchange information across the multi-vendor IaDI interfaces in an abstract manner.

The concept introduced by this YANG model is the notion of a mode. A mode is a combination of parameters or parameter ranges that is supported by a transceiver. As an example, operating a device in QPSK mode may use a different FEC and requires less OSNR to reach the FEC limit than the same transceiver operating in QAM16 mode. Given the number of parameters and their possible combinations it is important for vendors to be able to qualify a set of combinations which is the basis to define a mode. The YANG model furthermore provides means to selecting one mode as current-mode from that pre-defined list of modes supported by the transceiver module. Once selected, current-opt-if-och-mode-params provide the means to configure specific parameters at run time and retrieve actual parameters from the module. For example, the frequency is a parameter that can be set within min/max boundaries set by the current mode. Laser Temperature however is a read-only parameter available at run-time that can be checked against the mode boundaries and may trigger an event.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

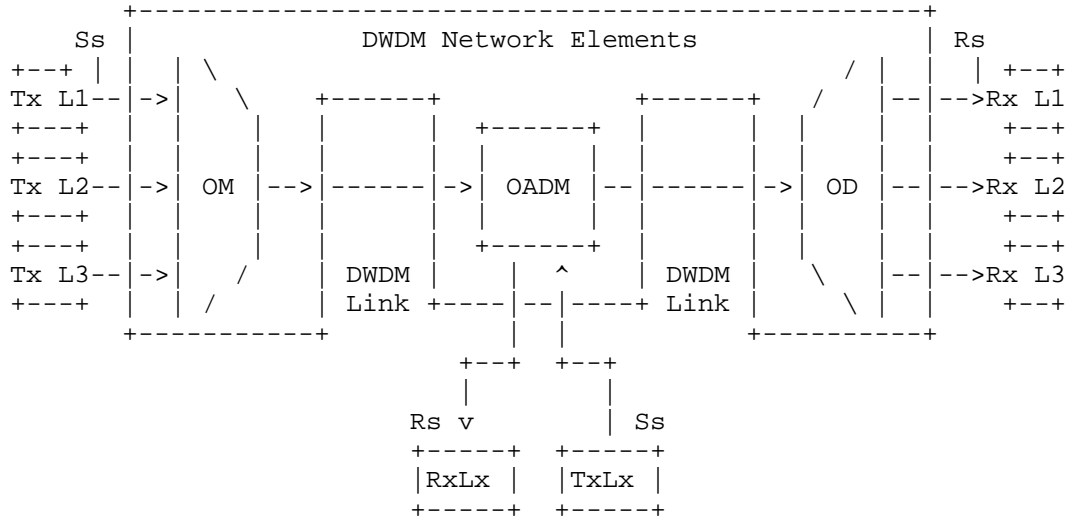
This memo specifies a Yang model for optical interfaces.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. In the description of OIDs the convention: Set (S) Get (G) and Trap (T) conventions will describe the action allowed by the parameter.

4. Overview

Figure 1 shows a set of reference points, for single-channel connection between transmitters (Tx) and receivers (Rx). Here the DWDM network elements include an OM and an OD (which are used as a pair with the opposing element), one or more optical amplifiers and may also include one or more OADMs.



Ss = reference point at the DWDM network element tributary output
Rs = reference point at the DWDM network element tributary input
Lx = Lambda x
OM = Optical Mux
OD = Optical Demux
OADM = Optical Add Drop Mux

from Fig. 5.1/G.698.2

Figure 1: External transponder in WDM networks

4.1. Optical Parameters Description

The link between the external transponders through a WDM network media channels are managed at the edges, i.e. at the transmitters (Tx) and receivers (Rx) attached to the S and R reference points respectively.

Definitions of the optical parameters are provided below to increase the readability of the document.

4.1.1. Parameters at Ss

output-power:

The mean launched power at Ss is the average power (in dBm) of a pseudo-random data sequence coupled into the DWDM link.

central frequency:

This parameter indicates the Central frequency value that Ss and Rs will be set to work (in THz)

4.1.2. Interface at point Rs

input-power:

The average received power (in dBm) at point Rs.

Curr-OSNR:

Current Optical Signal to Noise Ratio (OSNR) estimated at Rx Transceiver port.

Curr-q-factor:

"Q" factor estimated at Rx Transceiver port.

4.2. Use Cases

The use cases are described in draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

4.3. Optical Interface for external transponder in a WDM network

The ietf-ext-xponder-wdm-if is an augment to the ietf-interface. It allows the user to set the operating mode of transceivers as well as other operational parameters. The module provides also threshold settings and notifications to supervise measured parameters and notify the client.

```

module: ietf-ext-xponder-wdm-if
augment /if:interfaces/if:interface:
  +--rw optIfOChRsSs
    +--rw if-current-mode
      | +--ro mode-id?                string
      | +--ro min-central-frequency? uint32
      | +--ro max-central-frequency? uint32
      | +--ro min-input-power?       dbm-t
      | +--ro max-input-power?       dbm-t
      | +--ro min-output-power?      dbm-t
      | +--ro max-output-power?      dbm-t
      | +--ro osnr-margin?           int32
      | +--ro q-margin?              int32
      | +--ro fec-info?              string

```

```

|   +--ro fec-bitrate?                string
|   +--ro fec-gain?                  string
|   +--rw fec-ber-mantissa-threshold? uint32
|   +--rw fec-ber-exponent-threshold? int32
|   +--ro number-of-lanes?          uint32
|   +--ro min-laser-temperature?    int32
|   +--ro max-laser-temperature?    int32
|   +--ro min-rx-optical-power?     dbm-t
|   +--ro max-rx-optical-power?     dbm-t
|   +--ro min-chromatic-dispersion? int32
|   +--ro max-chromatic-dispersion? int32
|   +--ro min-diff-group-delay?     int32
|   +--ro max-diff-group-delay?     int32
|   +--ro modulation-format?        string
|   +--rw bits-per-symbol?          uint32
|   +--rw num-symbols-in-alphabet?   uint32
|   +--rw symbols-index?            uint32
|   +--ro i-center?                 int32
|   +--ro q-center?                 int32
|   +--ro i-noise-variance?          int32
|   +--ro q-noise-variance?          int32
|   +--ro a-noise-variance?          int32
|   +--ro p-noise-variance?          int32
+--ro if-supported-mode
|   +--ro number-of-modes-supported? uint32
|   +--ro mode-list* [mode-id]
|       +--ro mode-id                string
|       +--ro min-central-frequency?  uint32
|       +--ro max-central-frequency?  uint32
|       +--ro min-input-power?        dbm-t
|       +--ro max-input-power?        dbm-t
|       +--ro min-output-power?       dbm-t
|       +--ro max-output-power?       dbm-t
|       +--ro osnr-margin?            int32
|       +--ro q-margin?               int32
|       +--ro fec-info?               string
|       +--ro fec-bitrate?            string
|       +--ro fec-gain?               string
|       +--ro fec-ber-mantissa-threshold? uint32
|       +--ro fec-ber-exponent-threshold? int32
|       +--ro number-of-lanes?        uint32
|       +--ro min-laser-temperature?  int32
|       +--ro max-laser-temperature?  int32
|       +--ro min-rx-optical-power?   dbm-t
|       +--ro max-rx-optical-power?   dbm-t
|       +--ro min-chromatic-dispersion? int32
|       +--ro max-chromatic-dispersion? int32
|       +--ro min-diff-group-delay?   int32

```

```

|      +--ro max-diff-group-delay?          int32
|      +--ro modulation-format?            string
|      +--ro bits-per-symbol?              uint32
|      +--ro num-symbols-in-alphabet?      uint32
|      +--ro symbols-index?                uint32
|      +--ro i-center?                      int32
|      +--ro q-center?                      int32
|      +--ro i-noise-variance?              int32
|      +--ro q-noise-variance?              int32
|      +--ro a-noise-variance?              int32
|      +--ro p-noise-variance?              int32
+--rw current-opt-if-och-mode-params
|   +--rw mode-id?                          string
|   +--ro osnr-margin?                      int32
|   +--ro q-margin?                         int32
|   +--rw central-frequency?                uint32
|   +--rw output-power?                     int32
|   +--ro input-power?                      int32
|   +--rw min-fec-ber-mantissa-threshold?  uint32
|   +--rw min-fec-ber-exponent-threshold?  int32
|   +--rw max-fec-ber-mantissa-threshold?  uint32
|   +--rw max-fec-ber-exponent-threshold?  int32
|   +--rw number-of-tcas-supported?         uint32
|   +--rw mode-list* [tca-type]
|   |   +--rw tca-type                      opt-if-och-tca-types
|   |   +--rw min-threshold?                int32
|   |   +--rw max-threshold?                int32
|   +--ro cur-osnr?                         int32
|   +--ro cur-q-factor?                     int32
|   +--ro uncorrected-words?                uint64
|   +--ro fec-ber-mantissa?                  uint32
|   +--ro fec-ber-exponent?                  int32
notifications:
+---n opt-if-och-central-frequency-change
|   +--ro if-name?    -> /if:interfaces/interface/name
|   +--ro new-opt-if-och-central-frequency
|       +--ro central-frequency?          uint32
+---n opt-if-och-mode-change
|   +--ro if-name?    -> /if:interfaces/interface/name
|   +--ro mode-id?      string
+---n opt-if-och-min-tca
|   +--ro if-name?    -> /if:interfaces/interface/name
|   +--ro tca-type?    opt-if-och-tca-types

```

5. Structure of the Yang Module

ietf-ext-xponder-wdm-if is a top level model for the support of this feature.

6. Yang Module

The ietf-ext-xponder-wdm-if is defined as an extension to ietf interfaces.

```
<CODE BEGINS> file "ietf-ext-xponder-wdm-if.yang"

module ietf-ext-xponder-wdm-if {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ext-xponder-wdm-if";
  prefix ietf-ext-xponder-wdm-if;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP
     Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/ccamp/>
     WG List: <mailto:ccamp@ietf.org>

     Editor:  Dharini Hiremagalur
              <mailto:dharinih@juniper.net>";

  description
    "This module contains a collection of YANG definitions for
    configuring Optical interfaces.

    Copyright (c) 2016 IETF Trust and the persons identified
    as authors of the code.  All rights reserved.

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and
    subject to the license terms contained in, the Simplified
    BSD License set forth in Section 4.c of the IETF Trust's
    Legal Provisions Relating to IETF Documents
    (http://trustee.ietf.org/license-info).";
  revision "2017-03-06" {
    description
      "Revision 1.0";
    reference
      "";
  }
  revision "2016-03-17" {
    description
      "Initial revision.";
    reference
      "";
  }
}
```

```
typedef dbm-t {
    type decimal64 {
        fraction-digits 2;
        range "-50..-30 | -10..5 | 10000000";
    }
    description "
        Amplifier Power in dBm ";
}
typedef opt-if-och-tca-types {

    type enumeration {
        enum min-tx-power-tca {
            description " The min tx power tca";
        }
        enum max-tx-power-tca {
            description " The min tx power tca";
        }
        enum min-rx-power-tca{
            description " The min tx power tca";
        }
        enum max-rx-power-tca{
            description " The min tx power tca";
        }
        enum min-frequency-offset-tca{
            description " Min Frequency offset tca";
        }
        enum max-frequency-offset-tca{
            description " Max Frequency offset tca";
        }
        enum min-osnr-tca{
            description " Min OSNR tca";
        }
        enum max-osnr-tca{
            description " Max OSNR tca";
        }
        enum min-laser-temperature-tca{
            description " The min tx power tca";
        }
        enum max-laser-temperature-tca{
            description " Temperature tca";
        }
        enum min-fec-ber-tca{
            description " Min Pre Fec BER tca";
        }
        enum max-fec-ber-tca{
            description " Max Pre Fec BER tca";
        }
        enum min-q-tca{
```



```
        description "Min Q tca";
    }
    enum max-q-tca {
        description "Max Q tca";
    }
}
description " The different types of TCA's";
}

grouping opt-if-och-power {
    description "Interface optical Power";
    leaf output-power {
        type int32;
        units ".01dbm";
        description "The output power for this interface
                    in .01 dBm.
                    The setting of the output power is
                    optional";
    }

    leaf input-power {
        type int32;
        units ".01dbm";
        config false;
        description "The current input power of this
                    interface";
    }
}

grouping opt-if-och-tca-thresholds {
    description "Thresholds for TCA's";
    leaf tca-type {
        type opt-if-och-tca-types;
        description "type of the TCA eg TX Power";
    }
    leaf min-threshold {
        type int32;
        description " A TCA is generated if the variable is
                    less than this value";
    }

    leaf max-threshold {
        type int32;
        description " A TCA is generated if the variable is
                    more than this value";
    }
}
```

```
grouping opt-if-och-fec {
    description "Fec info";
    leaf fec-info {
        type string {
            length "1..255";
        }
        config false;
        description
            "Fec Type - eg GFEC";
    }
    leaf fec-bitrate {
        type string {
            length "1..255";
        }
        config false;
        description
            "Fec Overhead rate ";
    }
    leaf fec-gain {
        type string {
            length "1..255";
        }
        config false;
        description
            "Fec Overhead rate ";
    }
    leaf fec-ber-mantissa-threshold {
        type uint32;
        description " Mantissa of the FEC BER threshold";
    }
    leaf fec-ber-exponent-threshold {
        type int32;
        description " Exponent of the FEC BER threshold";
    }
}

grouping opt-if-och-central-frequency {
    description "Interface Central Frequency";
    leaf central-frequency {
        type uint32;
        description " This parameter indicates the frequency
            of this interface ";
    }
}

grouping opt-if-och-constellation {
    description "Optical constellation parameters";
```

```
    leaf i-center {
        type int32;
        units ".0001";
        config false;
        description "The In-phase coordinate of the selected
                    constellation symbol for this mode";
    }
    leaf q-center {
        type int32;
        units ".0001";
        config false;
        description "The Quadrature coordinate of the selected
                    constellation symbol for this mode";
    }
    leaf i-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the in-phase noise
                    component for this mode";
    }
    leaf q-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the quadrature noise
                    component for this mode";
    }
    leaf a-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the radial noise
                    component for this mode";
    }
    leaf p-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the phase noise
                    component for this mode";
    }
}

grouping opt-if-och-modulation-params {
    description "Optical modulation parameters for the lane";
    leaf modulation-format {
        type string {
```

```
        length "1..255";
    }
    config false;
    description
        "Modulation format for this mode";
}
leaf bits-per-symbol {
    type uint32;
    description " This parameter the bits per symbol for
        this mode.";
}
leaf num-symbols-in-alphabet {
    type uint32;
    description " This parameter the bits per symbol for
        this mode.";
}
leaf symbols-index {
    type uint32;
    description " This parameter is the symbol index this
        mode.";
}
}
uses opt-if-och-constellation;
}

grouping opt-if-och-lane-param {
    description "Optical parameters for the lane";
    leaf number-of-lanes {
        type uint32;
        config false;
        description
            "Number of optical lanes of this interface";
    }
    leaf min-laser-temperature {
        type int32;
        units ".01C";
        config false;
        description
            "Minimum Laser Temperature of this mode for
            this interface";
    }
    leaf max-laser-temperature {
        type int32;
        units ".01C";
        config false;
    }
}
```

```
        description
            "Maximum Laser Temperature of this mode for
             this interface";
    }
    leaf min-rx-optical-power {
        type dbm-t;
        config false;
        description
            "Minimum rx optical power of this mode for
             this interface";
    }
    leaf max-rx-optical-power {
        type dbm-t;
        config false;
        description
            "Maximum rx optical power of this mode for
             this interface";
    }
    leaf min-chromatic-dispersion {
        type int32;
        config false;
        description
            "Minimum chromatic dispersion of this mode
             for this interface";
    }
    leaf max-chromatic-dispersion {
        type int32;
        config false;
        description
            "Maximum chromatic dispersion of this
             mode for this interface";
    }
    leaf min-diff-group-delay {
        type int32;
        config false;
        description
            "Minimum Differential group delay of this
             mode for this interface";
    }
    leaf max-diff-group-delay {
        type int32;
        config false;
        description
            "Maximum Differential group delay of this
             mode for this interface";
    }
    uses opt-if-och-modulation-params;
}
```

```
grouping opt-if-och-tca-list {
  description "List of TCA's.";
  leaf number-of-tcas-supported {
    type uint32;
    description "Number of tcas
                  supported by this interface";
  }
  list mode-list {
    key "tca-type";
    description "List of the tcas";
    uses opt-if-och-tca-thresholds;
  }
}

grouping opt-if-och-fec-tca-thresholds {
  description "Pre FEC BER Thresholds for TCA's";
  leaf min-fec-ber-mantissa-threshold {
    type uint32;
    description " Min Mantissa of the FEC BER threshold";
  }
  leaf min-fec-ber-exponent-threshold {
    type int32;
    description " Min Exponent of the FEC BER threshold";
  }
  leaf max-fec-ber-mantissa-threshold {
    type uint32;
    description " Max Mantissa of the FEC BER threshold";
  }
  leaf max-fec-ber-exponent-threshold {
    type int32;
    description " Max Exponent of the FEC BER threshold";
  }
}

grouping opt-if-och-mode-params {
  description "OCh mode parameters.";
  leaf mode-id {
    type string {
      length "1..255";
    }
    description
      "Id for the OCh mode template";
  }
}
```

```
    leaf osnr-margin {
        type int32;
        units "dB";
        config false;
        description " OSNR margin to FEC threshold";
    }
    leaf q-margin {
        type int32;
        units "dB";
        config false;
        description " Q-Factor margin to FEC threshold";
    }
    uses opt-if-och-central-frequency;
    uses opt-if-och-power;
    uses opt-if-och-fec-tca-thresholds;
    uses opt-if-och-tca-list;
}

grouping opt-if-och-statistics {
    description "OCh statistics.";
    leaf cur-osnr {
        type int32;
        units "dB";
        config false;
        description " OSNR margin to FEC threshold";
    }
    leaf cur-q-factor {
        type int32;
        units "dB";
        config false;
        description " Q-Factor of the interface";
    }
    leaf uncorrected-words {
        type uint64;
        config false;
        description " Post FEC errored words";
    }
    leaf fec-ber-mantissa {
        type uint32;
        config false;
        description " Pre fec FEC errored words mantissa";
    }
    leaf fec-ber-exponent {
        type int32;
        config false;
        description " Pre fec FEC errored words exponent";
    }
}
```

```
}

grouping opt-if-och-mode {
  description "OCh mode template.";
  leaf mode-id {
    type string {
      length "1..255";
    }
    config false;
    description
      "Id for the OCh mode template";
  }
  leaf min-central-frequency {
    type uint32;
    config false;
    description "This parameter indicates the minimum
      frequency for this template ";
  }
  leaf max-central-frequency {
    type uint32;
    config false;
    description "This parameter indicates the minimum
      frequency for this template ";
  }
  leaf min-input-power {
    type dbm-t;
    config false;
    description "The minimum input power of this
      interface";
  }
  leaf max-input-power {
    type dbm-t;
    config false;
    description "The maximum input power of this
      interface";
  }
  leaf min-output-power {
    type dbm-t;
    config false;
    description "The minimum output power of this
      interface";
  }
  leaf max-output-power {
    type dbm-t;
    config false;
    description "The maximum output power of this
      interface";
  }
}
```



```
        leaf osnr-margin {
            type int32;
            units "dB";
            config false;
            description "OSNR margin to FEC threshold";
        }
        leaf q-margin {
            type int32;
            units "dB";
            config false;
            description "Q-Factor margin to FEC threshold";
        }
        uses opt-if-och-fec;
        uses opt-if-och-lane-param;
    }

    grouping opt-if-och-mode-list {
        description "List of Mode list group.";
        leaf number-of-modes-supported {
            type uint32;
            description "Number of modes
                supported by this interface";
        }
        list mode-list {
            key "mode-id";
            description "List of the modes ";
            uses opt-if-och-mode;
        }
    }

    notification opt-if-och-central-frequency-change {
        description "A change of Central Frequency has been
            detected.";
        leaf "if-name" {
            type leafref {
                path "/if:interfaces/if:interface/if:name";
            }
            description "Interface name";
        }
        container new-opt-if-och-central-frequency {
            description "The new Central Frequency of the
                interface";
            uses opt-if-och-central-frequency;
        }
    }
}
```

```
    }
  }

  notification opt-if-och-mode-change {
    description "A change of Mode Template has been
      detected.";
    leaf "if-name" {
      type leafref {
        path "/if:interfaces/if:interface/if:name";
      }
      description "Interface name";
    }
    leaf mode-id {
      type string {
        length "1..255";
      }
      description "Id for the OCh mode template";
    }
  }
}

notification opt-if-och-min-tca {
  description "A min output TCA notification.";
  leaf "if-name" {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
    description "Interface name";
  }
  leaf tca-type {
    type opt-if-och-tca-types;
    description "Type of TCA for eg min tx power TCA";
  }
}

augment "/if:interfaces/if:interface" {
  description "Parameters for an optical interface";
  container optIfOChRsSs {
    description "RsSs path configuration for an interface";
    container if-current-mode {
      description "Current mode template of the
        interface";
      uses opt-if-och-mode;
    }

    container if-supported-mode {
      config false;
      description "Supported mode list of
```

```
        this interface";
        uses opt-if-och-mode-list;
    }
    container current-opt-if-och-mode-params {
        description "Current parameters of
            this interface";
        uses opt-if-och-mode-params;
        uses opt-if-och-statistics;
    }
}
}
}

<CODE ENDS>
```

7. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operation and content.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces:ietf-ext-xponder-wdm-if

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

This document registers a YANG module in the YANG Module Names registry [RFC6020].

prefix: ietf-ext-xponder-wdm-if reference: RFC XXXX

9. Acknowledgements

10. Contributors

Dean Bogdanovic
Westford
U.S.A.
email ivandean@gmail.com

Bernd Zeuner
Deutsche Telekom
Darmstadt
Germany
email B.Zeuner@telekom.de

Massimiliano Salsi
Juniper Networks
1133 Innovation Way
Sunnyvale, CA, 94089
U.S.A.
+1 408936847
email msalsi@juniper.net

Arnold Mattheus
Deutsche Telekom
Darmstadt
Germany
email a.mattheus@telekom.de

Manuel Paul
Deutsche Telekom
Berlin
Germany
email Manuel.Paul@telekom.de

Walid Wakim
Cisco
9501 Technology Blvd
ROSEMONT, ILLINOIS 60018
UNITED STATES
email wwakim@cisco.com

Kam Lam
Nokia
USA
+1 732 331 3476
kam.lam@nokia.com

11. References

11.1. Normative References

- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<http://www.rfc-editor.org/info/rfc2863>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIV2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<http://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIV2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<http://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIV2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<http://www.rfc-editor.org/info/rfc2580>>.
- [RFC3591] Lam, H-K., Stewart, M., and A. Huynh, "Definitions of Managed Objects for the Optical Interface Type", RFC 3591, DOI 10.17487/RFC3591, September 2003, <<http://www.rfc-editor.org/info/rfc3591>>.
- [RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <<http://www.rfc-editor.org/info/rfc6205>>.
- [ITU.G698.2] International Telecommunications Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T Recommendation G.698.2, November 2009.

- [ITU.G709]
International Telecommunications Union, "Interface for the Optical Transport Network (OTN)", ITU-T Recommendation G.709, March 2003.
- [ITU.G872]
International Telecommunications Union, "Architecture of optical transport networks", ITU-T Recommendation G.872, November 2001.
- [ITU.G798]
International Telecommunications Union, "Characteristics of optical transport network hierarchy equipment functional blocks", ITU-T Recommendation G.798, October 2010.
- [ITU.G874]
International Telecommunications Union, "Management aspects of optical transport network elements", ITU-T Recommendation G.874, July 2010.
- [ITU.G874.1]
International Telecommunications Union, "Optical transport network (OTN): Protocol-neutral management information model for the network element view", ITU-T Recommendation G.874.1, January 2002.
- [ITU.G959.1]
International Telecommunications Union, "Optical transport network physical layer interfaces", ITU-T Recommendation G.959.1, November 2009.
- [ITU.G826]
International Telecommunications Union, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T Recommendation G.826, November 2009.
- [ITU.G8201]
International Telecommunications Union, "Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)", ITU-T Recommendation G.8201, April 2011.
- [ITU.G694.1]
International Telecommunications Union, "Spectral grids for WDM applications: DWDM frequency grid", ITU-T Recommendation G.694.1, June 2002.

[ITU.G7710]

International Telecommunications Union, "Common equipment management function requirements", ITU-T Recommendation G.7710, May 2008.

11.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<http://www.rfc-editor.org/info/rfc3410>>.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<http://www.rfc-editor.org/info/rfc2629>>.
- [RFC4181] Heard, C., Ed., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, DOI 10.17487/RFC4181, September 2005, <<http://www.rfc-editor.org/info/rfc4181>>.
- [I-D.ietf-ccamp-dwdm-if-mng-ctrl-fwk]
Kunze, R., Grammel, G., Beller, D., and G. Galimberti, "A framework for Management and Control of DWDM optical interface parameters", draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk-00 (work in progress), April 2016.
- [RFC4054] Strand, J., Ed. and A. Chiu, Ed., "Impairments and Other Constraints on Optical Layer Routing", RFC 4054, DOI 10.17487/RFC4054, May 2005, <<http://www.rfc-editor.org/info/rfc4054>>.

Appendix A. Change Log

This optional section should be removed before the internet draft is submitted to the IESG for publication as an RFC.

Note to RFC Editor: please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Gabriele Galimberti (editor)
Cisco
Via Santa Maria Molgora, 48 c
20871 - Vimercate
Italy

Phone: +390392091462
Email: ggalimbe@cisco.com

Ruediger Kunze
Deutsche Telekom
Winterfeldtstr. 21-27
10781 Berlin
Germany

Phone: +491702275321
Email: RKunze@telekom.de

Dharini Hiremagalur (editor)
Juniper
1133 Innovation Way
Sunnyvale - 94089 California
USA

Email: dharinih@juniper.net

Gert Grammel (editor)
Juniper
Oskar-Schlemmer Str. 15
80807 Muenchen
Germany

Phone: +49 1725186386
Email: ggrammel@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: April 25, 2019

G. Galimberti, Ed.
Cisco
R. Kunze
Deutsche Telekom
D. Hiremagalur, Ed.
G. Grammel, Ed.
Juniper
October 22, 2018

A YANG model to manage the optical interface parameters for an external
transponder in a WDM network
draft-dharini-ccamp-dwdm-if-param-yang-06

Abstract

This memo defines a Yang model related to the Optical Transceiver parameters characterising coherent 100G and above interfaces. 100G and above Transceivers support coherent modulation, multiple modulation formats, multiple FEC codes including some not yet specified (or by in phase of specification by) ITU-T G.698.2 [ITU.G698.2] or any other ITU-T recommendation. More context about the state of the Coherent transceivers is described in draft-many-coherent-DWDM-if-control. Use cases are described in RFC7698

The Yang model defined in this memo can be used for Optical Parameters monitoring and/or configuration of the endpoints of a multi-vendor IaDI optical link. The use of this model does not guarantee interworking of transceivers over a DWDM. Optical path feasibility and interoperability has to be determined by means outside the scope of this document. The purpose of this model is to program interface parameters to consistently configure the mode of operation of transceivers.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. The Internet-Standard Management Framework	3
3. Conventions	4
4. Overview	4
4.1. Optical Parameters Description	5
4.1.1. Parameters at Ss	5
4.1.2. Interface at point Rs	5
4.2. Use Cases	5
4.3. Optical Interface for external transponder in a WDM network	5
5. Structure of the Yang Module	8
6. Yang Module	8
7. Security Considerations	21
8. IANA Considerations	21
9. Acknowledgements	22
10. Contributors	22
11. References	23
11.1. Normative References	23
11.2. Informative References	25
Appendix A. Change Log	25
Appendix B. Open Issues	25
Authors' Addresses	26

1. Introduction

This memo defines a Yang model that translates and obsolete the SNMP mib module defined in draft-galikunze-ccamp-dwdm-if-snmp-mib for managing single channel optical interface parameters of DWDM applications, using the approach specified in G.698.2. This model supports parameters to characterize coherent transceivers found in current implementations to specify the mode of operation. As application identifiers like those specified in ITU-T G.874.1 [ITU.G874.1] are not available we use mode templates instead. A mode template describes transceiver characteristics in detail and can be identified by a mode-id.

This draft refers and supports the RFC7698 and draft-many-coherent-DWDM-if-control.

The YANG model describing and extending the optical parameters allows different vendors and operators to retrieve, provision and exchange information across the multi-vendor IaDI interfaces in an abstract manner.

The they concept introduced by this YANG model is the notion of a mode. A mode is a combination of parameters or parameter ranges that is supported by a transceiver. As an example, operating a device in QPSK mode may use a different FEC and requires less OSNR to reach the FEC limit than the same transceiver operating in QAM16 mode. Given the number of parameters and their possible combinations it is important for vendors to be able to qualify a set of combinations which is the basis to define a mode. The YANG model furthermore provides means to selecting one mode as current-mode from that pre-defined list of modes supported by the transceiver module. Once selected, current-opt-if-och-mode-params provide the means to configure specific parameters at run time and retrieve actual parameters from the module. For example, the frequency is a parameter that can be set within min/max boundaries set by the current mode. Laser Temperature however is a ro parameter available at run-time that can be checked against the mode boundaries and may trigger an event.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

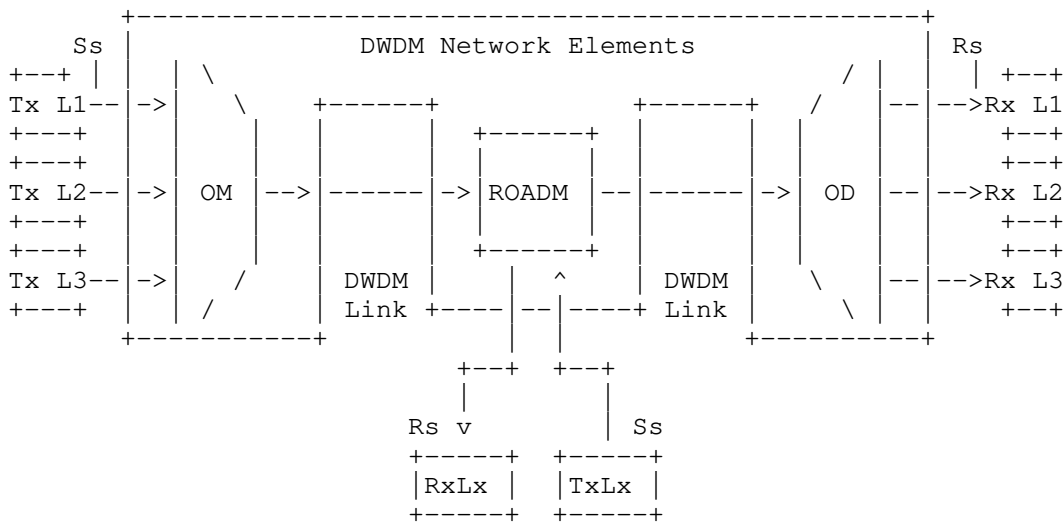
This memo specifies a Yang model for optical interfaces.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. In the description of OIDs the convention: Set (S) Get (G) and Trap (T) conventions will describe the action allowed by the parameter.

4. Overview

Figure 1 shows a set of reference points, for single-channel connection between transmitters (Tx) and receivers (Rx). Here the DWDM network elements include an OM and an OD (which are used as a pair with the opposing element), one or more optical amplifiers and may also include one or more OADMs.



Ss = reference point at the DWDM network element tributary output
Rs = reference point at the DWDM network element tributary input
Lx = Lambda x
OM = Optical Mux
OD = Optical Demux
ROADM = Reconfigurable Optical Add Drop Mux

from Fig. 5.1/G.698.2

Figure 1: External transponder in WDM networks

4.1. Optical Parameters Description

The link between the external transponders through a WDM network media channels are managed at the edges, i.e. at the transmitters (Tx) and receivers (Rx) attached to the S and R reference points respectively.

Definitions of the optical parameters are provided below to increase the readability of the document.

4.1.1. Parameters at Ss

output-power:

The mean launched power at Ss is the average power (in dBm) of a pseudo-random data sequence coupled into the DWDM link.

central frequency:

This parameter indicates the Central frequency value that Ss and Rs will be set to work (in THz)

4.1.2. Interface at point Rs

input-power:

The average received power (in dBm) at point Rs.

Curr-OSNR:

Current Optical Signal to Noise Ratio (OSNR) estimated at Rx Transceiver port.

Curr-q-factor:

"Q" factor estimated at Rx Transceiver port.

4.2. Use Cases

The use cases are described in draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

4.3. Optical Interface for external transponder in a WDM network

The ietf-ext-xponder-wdm-if is an augment to the ietf-interface. It allows the user to set the operating mode of transceivers as well as other operational parameters. The module provides also threshold settings and notifications to supervise measured parameters and notify the client.

```
module: ietf-ext-xponder-wdm-if
  augment /if:interfaces/if:interface:
    +--rw optIfOChRsSs
    +--rw if-current-mode
```

```

+--ro mode-id?                string
+--ro min-central-frequency?   uint32
+--ro max-central-frequency?   uint32
+--ro min-input-power?         dbm-t
+--ro max-input-power?         dbm-t
+--ro min-output-power?        dbm-t
+--ro max-output-power?        dbm-t
+--ro osnr-margin?             int32
+--ro q-margin?                int32
+--ro fec-info?                string
+--ro fec-bitrate?             string
+--ro fec-gain?                string
+--rw fec-ber-mantissa-threshold? uint32
+--rw fec-ber-exponent-threshold? int32
+--ro number-of-lanes?         uint32
+--ro min-laser-temperature?    int32
+--ro max-laser-temperature?    int32
+--ro min-rx-optical-power?     dbm-t
+--ro max-rx-optical-power?     dbm-t
+--ro min-chromatic-dispersion? int32
+--ro max-chromatic-dispersion? int32
+--ro min-diff-group-delay?     int32
+--ro max-diff-group-delay?     int32
+--ro modulation-format?       string
+--rw bits-per-symbol?         uint32
+--rw num-symbols-in-alphabet?  uint32
+--rw symbols-index?           uint32
+--ro i-center?                int32
+--ro q-center?                int32
+--ro i-noise-variance?         int32
+--ro q-noise-variance?         int32
+--ro a-noise-variance?         int32
+--ro p-noise-variance?         int32
+--ro if-supported-mode
+--ro number-of-modes-supported? uint32
+--ro mode-list* [mode-id]
    +--ro mode-id                string
    +--ro min-central-frequency?  uint32
    +--ro max-central-frequency?  uint32
    +--ro min-input-power?        dbm-t
    +--ro max-input-power?        dbm-t
    +--ro min-output-power?       dbm-t
    +--ro max-output-power?       dbm-t
    +--ro osnr-margin?            int32
    +--ro q-margin?              int32
    +--ro fec-info?              string
    +--ro fec-bitrate?           string
    +--ro fec-gain?              string

```

```

    +--ro fec-ber-mantissa-threshold?  uint32
    +--ro fec-ber-exponent-threshold?  int32
    +--ro number-of-lanes?             uint32
    +--ro min-laser-temperature?       int32
    +--ro max-laser-temperature?       int32
    +--ro min-rx-optical-power?        dbm-t
    +--ro max-rx-optical-power?        dbm-t
    +--ro min-chromatic-dispersion?    int32
    +--ro max-chromatic-dispersion?    int32
    +--ro min-diff-group-delay?        int32
    +--ro max-diff-group-delay?        int32
    +--ro modulation-format?           string
    +--ro bits-per-symbol?             uint32
    +--ro num-symbols-in-alphabet?     uint32
    +--ro symbols-index?               uint32
    +--ro i-center?                    int32
    +--ro q-center?                    int32
    +--ro i-noise-variance?            int32
    +--ro q-noise-variance?            int32
    +--ro a-noise-variance?            int32
    +--ro p-noise-variance?            int32
+--rw current-opt-if-och-mode-params
    +--rw mode-id?                     string
    +--ro osnr-margin?                 int32
    +--ro q-margin?                    int32
    +--rw central-frequency?           uint32
    +--rw output-power?                int32
    +--ro input-power?                 int32
    +--rw min-fec-ber-mantissa-threshold?  uint32
    +--rw min-fec-ber-exponent-threshold?  int32
    +--rw max-fec-ber-mantissa-threshold?  uint32
    +--rw max-fec-ber-exponent-threshold?  int32
    +--rw number-of-tcas-supported?       uint32
    +--rw mode-list* [tca-type]
        +--rw tca-type                 opt-if-och-tca-types
        +--rw min-threshold?            int32
        +--rw max-threshold?            int32
    +--ro cur-osnr?                     int32
    +--ro cur-q-factor?                 int32
    +--ro uncorrected-words?            uint64
    +--ro fec-ber-mantissa?             uint32
    +--ro fec-ber-exponent?             int32

```

notifications:

```

+---n opt-if-och-central-frequency-change
    +--ro if-name?  -> /if:interfaces/interface/name
    +--ro new-opt-if-och-central-frequency
        +--ro central-frequency?  uint32

```

```
+---n opt-if-och-mode-change
|   +---ro if-name?    -> /if:interfaces/interface/name
|   +---ro mode-id?           string
+---n opt-if-och-min-tca
|   +---ro if-name?    -> /if:interfaces/interface/name
|   +---ro tca-type?   opt-if-och-tca-types
```

5. Structure of the Yang Module

ietf-ext-xponder-wdm-if is a top level model for the support of this feature.

6. Yang Module

The ietf-ext-xponder-wdm-if is defined as an extension to ietf interfaces.

<CODE BEGINS> file "ietf-ext-xponder-wdm-if.yang"

```
module ietf-ext-xponder-wdm-if {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ext-xponder-wdm-if";
  prefix ietf-ext-xponder-wdm-if;

  import ietf-interfaces {
    prefix if;
  }

  organization
    "IETF CCAMP
    Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/ccamp/>
    WG List:  <mailto:ccamp@ietf.org>

    Editor:   Dharini Hiremagalur
              <mailto:dharith@juniper.net>";

  description
    "This module contains a collection of YANG definitions for
    configuring Optical interfaces.

    Copyright (c) 2016 IETF Trust and the persons identified
    as authors of the code. All rights reserved."
```


Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).";

```
revision "2018-10-22" {
    description
        "Revision 1.2";
    reference
        "";
}

revision "2018-03-06" {
    description
        "Revision 1.1";
    reference
        "";
}

revision "2017-03-06" {
    description
        "Revision 1.0";
    reference
        "";
}
revision "2016-03-17" {
    description
        "Initial revision.";
    reference
        "";
}

typedef dbm-t {
    type decimal64 {
        fraction-digits 2;
        range "-50..-30 | -10..5 | 10000000";
    }
    description "
        Amplifier Power in dBm ";
}
typedef opt-if-och-tca-types {
    type enumeration {
        enum max-laser-linewdt {
```

```
        description " The maximum laser linewidth";
    }
    enum min-tx-power-tca {
        description " The min tx power tca";
    }
    enum max-tx-power-tca {
        description " The min tx power tca";
    }
    enum min-rx-power-tca{
        description " The min tx power tca";
    }
    enum max-rx-power-tca{
        description " The min tx power tca";
    }
    enum max-pol-power-diff-tca{
        description " The power diff. between polariz. tca";
    }
    enum max-pol-skew-diff-tca{
        description " The Skew between the two polariz. tca";
    }
    enum min-frequency-offset-tca{
        description " Min Frequency offset tca";
    }
    enum max-frequency-offset-tca{
        description " Max Frequency offset tca";
    }
    enum min-osnr-tca{
        description " Min OSNR tca";
    }
    enum max-osnr-tca{
        description " Max OSNR tca";
    }
    enum min-laser-temperature-tca{
        description " The min tx power tca";
    }
    enum max-laser-temperature-tca{
        description " Temperature tca";
    }
    enum min-fec-ber-tca{
        description " Min Pre Fec BER tca";
    }
    enum max-fec-ber-tca{
        description " Max Pre Fec BER tca";
    }
    enum min-q-tca{
        description "Min Q tca";
    }
    enum max-q-tca {
```

```

        description "Max Q tca";
    }
}
description " The different types of TCA's";
}

grouping opt-if-och-power {
    description "Interface optical Power";
    leaf output-power {
        type int32;
        units ".01dbm";
        description "The output power for this interface
                    in .01 dBm.
                    The setting of the output power is
                    optional";
    }

    leaf input-power {
        type int32;
        units ".01dbm";
        config false;
        description "The current input power of this
                    interface";
    }
}

grouping opt-if-och-tca-thresholds {
    description "Thresholds for TCA's";
    leaf tca-type {
        type opt-if-och-tca-types;
        description "type of the TCA eg TX Power";
    }
    leaf min-threshold {
        type int32;
        description " A TCA is generated if the variable is
                    less than this value";

    }
    leaf max-threshold {
        type int32;
        description " A TCA is generated if the variable is
                    more than this value";
    }
}

grouping opt-if-och-fec {
    description "Fec info";
}

```

```

        leaf fec-info {
            type string {
                length "1..255";
            }
            config false;
            description
                "Fec Type - eg GFEC";
        }
        leaf fec-bitrate {
            type string {
                length "1..255";
            }
            config false;
            description
                "Fec Overhead rate ";
        }
        leaf fec-gain {
            type string {
                length "1..255";
            }
            config false;
            description
                "Fec Overhead rate ";
        }
        leaf fec-ber-mantissa-threshold {
            type uint32;
            description " Mantissa of the FEC BER threshold";
        }
        leaf fec-ber-exponent-threshold {
            type int32;
            description " Exponent of the FEC BER threshold";
        }
    }

    grouping opt-if-och-central-frequency {
        description "Interface Central Frequency";
        leaf central-frequency {
            type uint32;
            description " This parameter indicates the frequency
                of this interface ";
        }
    }

    grouping opt-if-och-constellation {
        description "Optical constellation parameters";
        leaf i-center {
            type int32;

```

```

        units ".0001";
        config false;
        description "The In-phase coordinate of the selected
                    constellation symbol for this mode";
    }
    leaf q-center {
        type int32;
        units ".0001";
        config false;
        description "The Quadrature coordinate of the selected
                    constellation symbol for this mode";
    }
    leaf i-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the in-phase noise
                    component for this mode";
    }
    leaf q-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the quadrature noise
                    component for this mode";
    }
    leaf a-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the radial noise
                    component for this mode";
    }
    leaf p-noise-variance {
        type int32;
        units ".001";
        config false;
        description "The Variance of the phase noise
                    component for this mode";
    }
}

grouping opt-if-och-modulation-params {
    description "Optical modulation parameters for the lane";
    leaf modulation-format {
        type string {
            length "1..255";
        }
    }
}

```

```

        config false;
        description
            "Modulation format for this mode";
    }
    leaf bits-per-symbol {
        type uint32;
        description " This parameter the bits per symbol for
            this mode.";
    }
    leaf num-symbols-in-alphabet {
        type uint32;
        description " This parameter the bits per symbol for
            this mode.";
    }
    leaf symbols-index {
        type uint32;
        description " This parameter is the symbol index this
            mode.";
    }
    uses opt-if-och-constellation;
}

grouping opt-if-och-lane-param {
    description "Optical parameters for the lane";
    leaf number-of-lanes {
        type uint32;
        config false;
        description
            "Number of optical lanes of this interface";
    }
    leaf min-laser-temperature {
        type int32;
        units ".01C";
        config false;
        description
            "Minimum Laser Temperature of this mode for
            this interface";
    }
    leaf max-laser-temperature {
        type int32;
        units ".01C";
        config false;
        description
            "Maximum Laser Temperature of this mode for

```

```

        this interface";
    }
    leaf min-rx-optical-power {
        type dbm-t;
        config false;
        description
            "Minimum rx optical power of this mode for
             this interface";
    }
    leaf max-rx-optical-power {
        type dbm-t;
        config false;
        description
            "Maximum rx optical power of this mode for
             this interface";
    }
    leaf min-chromatic-dispersion {
        type int32;
        config false;
        description
            "Minimum chromatic dispersion of this mode
             for this interface";
    }
    leaf max-chromatic-dispersion {
        type int32;
        config false;
        description
            "Maximum chromatic dispersion of this
             mode for this interface";
    }
    leaf min-diff-group-delay {
        type int32;
        config false;
        description
            "Minimum Differential group delay of this
             mode for this interface";
    }
    leaf max-diff-group-delay {
        type int32;
        config false;
        description
            "Maximum Differential group delay of this
             mode for this interface";
    }
    uses opt-if-och-modulation-params;
}

```

```

grouping opt-if-och-tca-list {
  description "List of TCA's.";
  leaf number-of-tcas-supported {
    type uint32;
    description "Number of tcas
                 supported by this interface";
  }
  list mode-list {
    key "tca-type";
    description "List of the tcas";
    uses opt-if-och-tca-thresholds;
  }
}

grouping opt-if-och-fec-tca-thresholds {
  description "Pre FEC BER Thresholds for TCA's";
  leaf min-fec-ber-mantissa-threshold {
    type uint32;
    description " Min Mantissa of the FEC BER threshold";
  }
  leaf min-fec-ber-exponent-threshold {
    type int32;
    description " Min Exponent of the FEC BER threshold";
  }
  leaf max-fec-ber-mantissa-threshold {
    type uint32;
    description " Max Mantissa of the FEC BER threshold";
  }
  leaf max-fec-ber-exponent-threshold {
    type int32;
    description " Max Exponent of the FEC BER threshold";
  }
}

grouping opt-if-och-mode-params {
  description "OCh mode parameters.";
  leaf mode-id {
    type string {
      length "1..255";
    }
    description
      "Id for the OCh mode template";
  }
}

```



```

    leaf osnr-margin {
        type int32;
        units "dB";
        config false;
        description " OSNR margin to FEC threshold";
    }
    leaf q-margin {
        type int32;
        units "dB";
        config false;
        description " Q-Factor margin to FEC threshold";
    }
    uses opt-if-och-central-frequency;
    uses opt-if-och-power;
    uses opt-if-och-fec-tca-thresholds;
    uses opt-if-och-tca-list;
}

grouping opt-if-och-statistics {
    description "OCh statistics.";
    leaf cur-osnr {
        type int32;
        units "dB";
        config false;
        description " OSNR margin to FEC threshold";
    }
    leaf cur-q-factor {
        type int32;
        units "dB";
        config false;
        description " Q-Factor of the interface";
    }
    leaf uncorrected-words {
        type uint64;
        config false;
        description " Post FEC errored words";
    }
    leaf fec-ber-mantissa {
        type uint32;
        config false;
        description " Pre fec FEC errored words mantissa";
    }
    leaf fec-ber-exponent {
        type int32;
        config false;
        description " Pre fec FEC errored words exponent";
    }
}

```

```

}

grouping opt-if-och-mode {
  description "OCh mode template.";
  leaf mode-id {
    type string {
      length "1..255";
    }
    config false;
    description
      "Id for the OCh mode template";
  }
  leaf min-central-frequency {
    type uint32;
    config false;
    description "This parameter indicates the minimum
      frequency for this template ";
  }
  leaf max-central-frequency {
    type uint32;
    config false;
    description "This parameter indicates the minimum
      frequency for this template ";
  }
  leaf min-input-power {
    type dbm-t;
    config false;
    description "The minimum input power of this
      interface";
  }
  leaf max-input-power {
    type dbm-t;
    config false;
    description "The maximum input power of this
      interface";
  }
  leaf min-output-power {
    type dbm-t;
    config false;
    description "The minimum output power of this
      interface";
  }
  leaf max-output-power {
    type dbm-t;
    config false;
    description "The maximum output power of this
      interface";
  }
}

```

```

        leaf osnr-margin {
            type int32;
            units "dB";
            config false;
            description "OSNR margin to FEC threshold";
        }
        leaf q-margin {
            type int32;
            units "dB";
            config false;
            description "Q-Factor margin to FEC threshold";
        }
        uses opt-if-och-fec;
        uses opt-if-och-lane-param;
    }

    grouping opt-if-och-mode-list {
        description "List of Mode list group.";
        leaf number-of-modes-supported {
            type uint32;
            description "Number of modes
                supported by this interface";
        }
        list mode-list {
            key "mode-id";
            description "List of the modes ";
            uses opt-if-och-mode;
        }
    }

    notification opt-if-och-central-frequency-change {
        description "A change of Central Frequency has been
            detected.";
        leaf "if-name" {
            type leafref {
                path "/if:interfaces/if:interface/if:name";
            }
            description "Interface name";
        }
        container new-opt-if-och-central-frequency {
            description "The new Central Frequency of the
                interface";
            uses opt-if-och-central-frequency;
        }
    }

```

```

    }
  }

  notification opt-if-och-mode-change {
    description "A change of Mode Template has been
      detected.";
    leaf "if-name" {
      type leafref {
        path "/if:interfaces/if:interface/if:name";
      }
      description "Interface name";
    }
    leaf mode-id {
      type string {
        length "1..255";
      }
      description "Id for the OCh mode template";
    }
  }
}

notification opt-if-och-min-tca {
  description "A min output TCA notification.";
  leaf "if-name" {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
    description "Interface name";
  }
  leaf tca-type {
    type opt-if-och-tca-types;
    description "Type of TCA for eg min tx power TCA";
  }
}

augment "/if:interfaces/if:interface" {
  description "Parameters for an optical interface";
  container optIfOChRsSs {
    description "RsSs path configuration for an interface";
    container if-current-mode {
      description "Current mode template of the
        interface";
      uses opt-if-och-mode;
    }

    container if-supported-mode {
      config false;
      description "Supported mode list of

```

```
        this interface";
        uses opt-if-och-mode-list;
    }
    container current-opt-if-och-mode-params {
        description "Current parameters of
                    this interface";
        uses opt-if-och-mode-params;
        uses opt-if-och-statistics;
    }
}
}
```

<CODE ENDS>

7. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operation and content.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces:ietf-ext-xponder-wdm-if

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

This document registers a YANG module in the YANG Module Names registry [RFC6020].

prefix: ietf-ext-xponder-wdm-if reference: RFC XXXX

9. Acknowledgements

10. Contributors

Dean Bogdanovic
Westford
U.S.A.
email ivandean@gmail.com

Bernd Zeuner
Deutsche Telekom
Darmstadt
Germany
email B.Zeuner@telekom.de

Massimiliano Salsi
Juniper Networks
1133 Innovation Way
Sunnyvale, CA, 94089
U.S.A.
+1 408936847
email msalsi@juniper.net

Arnold Mattheus
Deutsche Telekom
Darmstadt
Germany
email a.mattheus@telekom.de

Manuel Paul
Deutsche Telekom
Berlin
Germany
email Manuel.Paul@telekom.de

Walid Wakim
Cisco
9501 Technology Blvd
ROSEMONT, ILLINOIS 60018
UNITED STATES
email wwakim@cisco.com

Kam Lam
Nokia
USA
+1 732 331 3476
kam.lam@nokia.com

11. References

11.1. Normative References

- [ITU.G694.1]
International Telecommunications Union, "Spectral grids for WDM applications: DWDM frequency grid", ITU-T Recommendation G.694.1, February 2012.
- [ITU.G698.2]
International Telecommunications Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T Recommendation G.698.2, November 2009.
- [ITU.G709]
International Telecommunications Union, "Interface for the Optical Transport Network (OTN)", ITU-T Recommendation G.709, June 2016.
- [ITU.G7710]
International Telecommunications Union, "Common equipment management function requirements", ITU-T Recommendation G.7710, August 2017.
- [ITU.G798]
International Telecommunications Union, "Characteristics of optical transport network hierarchy equipment functional blocks", ITU-T Recommendation G.798, December 2017.
- [ITU.G8201]
International Telecommunications Union, "Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)", ITU-T Recommendation G.8201, April 2011.
- [ITU.G826]
International Telecommunications Union, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T Recommendation G.826, December 2002.
- [ITU.G872]
International Telecommunications Union, "Architecture of optical transport networks", ITU-T Recommendation G.872, January 2017.

- [ITU.G874] International Telecommunications Union, "Management aspects of optical transport network elements", ITU-T Recommendation G.874, July 2010.
- [ITU.G874.1] International Telecommunications Union, "Optical transport network (OTN): Protocol-neutral management information model for the network element view", ITU-T Recommendation G.874.1, November 2016.
- [ITU.G959.1] International Telecommunications Union, "Optical transport network physical layer interfaces", ITU-T Recommendation G.959.1, November 2009.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<https://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<https://www.rfc-editor.org/info/rfc2580>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.
- [RFC3591] Lam, H-K., Stewart, M., and A. Huynh, "Definitions of Managed Objects for the Optical Interface Type", RFC 3591, DOI 10.17487/RFC3591, September 2003, <<https://www.rfc-editor.org/info/rfc3591>>.

- [RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <<https://www.rfc-editor.org/info/rfc6205>>.

11.2. Informative References

- [I-D.ietf-ccamp-dwdm-if-mng-ctrl-fwk]
Kunze, R., Grammel, G., Beller, D., Galimberti, G., and J. Meuric, "A framework for Management and Control of DWDM optical interface parameters", draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk-11 (work in progress), June 2018.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<https://www.rfc-editor.org/info/rfc2629>>.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<https://www.rfc-editor.org/info/rfc3410>>.
- [RFC4054] Strand, J., Ed. and A. Chiu, Ed., "Impairments and Other Constraints on Optical Layer Routing", RFC 4054, DOI 10.17487/RFC4054, May 2005, <<https://www.rfc-editor.org/info/rfc4054>>.
- [RFC4181] Heard, C., Ed., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, DOI 10.17487/RFC4181, September 2005, <<https://www.rfc-editor.org/info/rfc4181>>.

Appendix A. Change Log

This optional section should be removed before the internet draft is submitted to the IESG for publication as an RFC.

Note to RFC Editor: please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Gabriele Galimberti (editor)
Cisco
Via Santa Maria Molgora, 48 c
20871 - Vimercate
Italy

Phone: +390392091462
Email: ggalimbe@cisco.com

Ruediger Kunze
Deutsche Telekom
Winterfeldtstr. 21-27
10781 Berlin
Germany

Phone: +491702275321
Email: RKunze@telekom.de

Dharini Hiremagalur (editor)
Juniper
1133 Innovation Way
Sunnyvale - 94089 California
USA

Email: dharinih@juniper.net

Gert Grammel (editor)
Juniper
Oskar-Schlemmer Str. 15
80807 Muenchen
Germany

Phone: +49 1725186386
Email: ggrammel@juniper.net

INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: May 4, 2017

Santosh Esale
Kireeti Kompella
Juniper Networks
October 31, 2016

LDP Extensions for RMR
draft-esale-ldp-rmr-extensions-00

Abstract

This document describes LDP extensions to signal Resilient MPLS Ring (RMR) Label Switched Paths (LSPs). An RMR LSP is a multipoint to point LSP signaled using LDP (Label Distribution Protocol). RMR Architecture document - draft-ietf-mpls-rmr-02 - describes why and how MPLS should be used in ring topologies.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
2.	Terminology	3
3.	Protocol extensions	4
3.1.	Ring LSP Capability	4
3.2.	Ring FEC Element	4
4.	Ring Procedures	6
4.1	Upstream LSR	6
4.2	Ring Label Mapping Procedures	7
4.2.1	Definitions	7
4.2.2	Preliminary	7
4.2.3	Egress LSR	7
4.2.4	Ingress and Transit LSR	8
4.3	Equal Cost Multipath (ECMP)	8
4.4	Protection	9
5.	LSP Hierarchy	9
6.	Ring LSPs	10
7.	Security Considerations	11
8.	IANA Considerations	11
9.	Acknowledgments	11
10.	Contributors	11
11.	References	12
11.1	Normative References	12
11.2	Informative References	12
	Authors' Addresses	13

1 Introduction

This document describes LDP extensions to signal resilient MPLS ring (RMR) label switched paths (LSPs). An RMR LSP is a multipoint to point LSP signaled using LDP (Label Distribution Protocol). Architecture document of RMR - draft-ietf-mpls-rmr-02 - describes why and how MPLS should be used in ring topologies.

The ring is either auto-discovered or configured using IGP protocol such as OSPF or ISIS. IGP extensions for RMR will be described in a companion documents. After the ring discovery, each ring node acting as egress constructs and signals a clockwise (CW) and anti-clockwise (AC) ring FEC towards AC and CW direction respectively. Each transit node that receives the RMR FEC signals this LSP further in same direction using RMR link state database. In addition, it also adds a transit and ingress route for this LSP. Once the signaling is complete, every node in a ring should have two counter rotating LSPs in CW and AC direction to reach every other node on the ring.

2. Terminology

A ring consists of a subset of n nodes $\{R_i, 0 \leq i < n\}$. The direction from node R_i to R_{i+1} is defined as "clockwise" (CW) and the reverse direction is defined as "anti-clockwise" (AC). As there may be several rings in a graph, each ring is numbered with a distinct ring ID (RID).

The following terminology is used for ring LSPs:

Ring ID (RID): A non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network. A node may belong to multiple rings.

Ring node: A member of a ring. Note that a device may belong to several rings.

Node index: A logical numbering of nodes in a ring, from zero upto one less than the ring size. Used purely for exposition in this document.

Ring neighbors: Nodes whose indices differ by one (modulo ring size).

Ring links: Links that connect ring neighbors.

Express links: Links that connect non-neighboring ring nodes.

MP2P LSP: Each LSP in the ring is a multipoint to point LSP such that LSP can have multiple ingress nodes and one egress node.

3. Protocol extensions

This section describes LDP extensions to signal RMR LSP in a ring.

3.1. Ring LSP Capability

RMR LSPs support for a LSR is advertised using LDP capabilities as defined in [RFC5561]. An implementation that supports the RMR procedures specified in this document MUST add the procedures pertaining to Capability Parameters for Initialization messages.

A new optional capability parameter TLV, RMR Capability, is defined. Following is the format of the RMR Capability Parameter:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F| RMR Capability (TBD) | Length (= 1) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|S| Reserved |
+---+---+---+---+---+

```

As described in [RFC5561]

U: set to 1. Ignore, if not known.

F: Set to 0. Do not forward.

S: MUST be set to 1 to advertise the RMR Capability TLV.

The RMR Capability TLV MUST be advertised in the LDP Initialization message. If the peer has not advertised the RMR capability, then label messages pertaining to RMR FEC Element MUST not be sent to the peer.

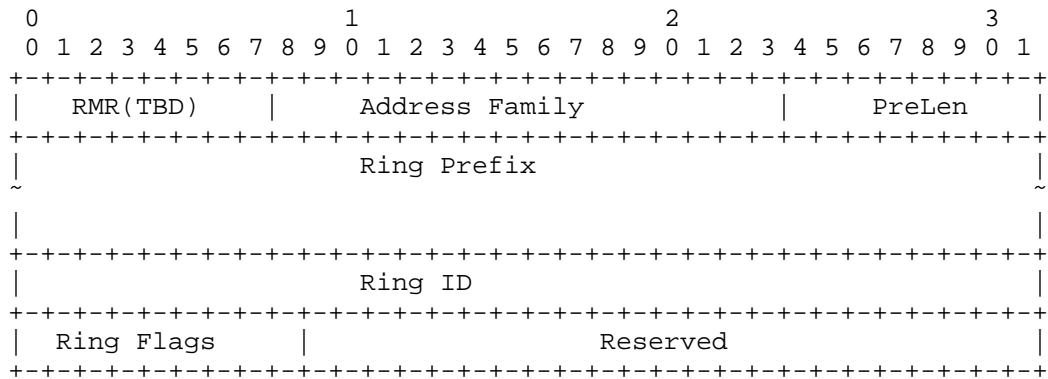
3.2. Ring FEC Element

In order to setup RMR LSP in clockwise and anti-clockwise direction for every ring node, this document defines new protocol entity, the RMR FEC Element, to be used as a FEC Element in the FEC TLV.

The RMR FEC Element consists of the ring address, ring identifier and ring flags which depicts ring direction. The combination of ring address, ring identifier and ring flags uniquely identifies a ring

LSP within the MPLS network.

The RMR FEC Element value encoding is as follows:



FEC Type

One octet quantity containing a value from FEC Type Name Space that encodes the fec type for a RMR LDP LSP.

Address Family

Two octet quantity containing a value from ADDRESS FAMILY NUMBERS in [ASSIGNED_AF] that encodes the address family for the address prefix in the Prefix field.

PreLen

One octet unsigned integer containing the length in bits of the address prefix that follows. A length of zero indicates a prefix that matches all addresses (the default destination); in this case, the Prefix itself is zero octets).

Prefix

An address prefix encoded according to the Address Family field, whose length, in bits, was specified in the PreLen field, padded to a byte boundary.

Ring ID (RID)

A four-octet non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network.

Ring Flags

0	1	2	3	4	5	6	7

```

|RF | Reserved |
+-----+
The value of ring flags (RF) is defined as follows:
1: Clockwise (CW) FEC
2: Anti-clockwise (AC) FEC

```

4. Ring Procedures

Resilient MPLS ring architecture needs interaction between MPLS protocols such as LDP and RSVP and IGP to signal a RMR LSP.

4.1 Upstream LSR

This section describes how to select a upstream LSR for RMR LSP. Consider MPLS rings as follows:

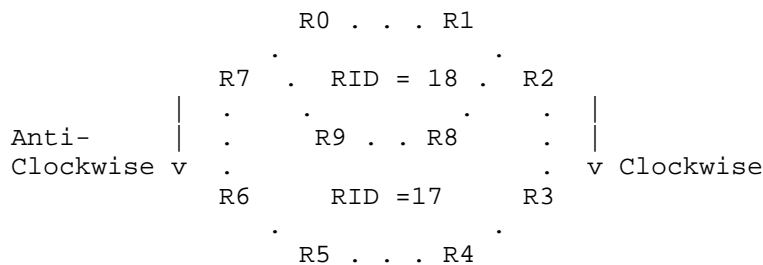


Figure 1: Two Rings with 10 nodes

During the discovery of a MPLS ring, IGP populates its link state database with ring information. After the discovery, there are just two paths - one in clockwise direction and other in anti-clockwise direction - for every ring neighbor on a specific ring. For instance, the following table shows router R0's path for ring 17 and 18 depicted in figure 1.

RID	CW neighbor	AC neighbor
17	R1	R7
18	R1	R9

Figure 2: R0's RMR upstream signaling table

IGP informs LDP that a new MPLS ring, RID 17, is discovered. A LDP transit LSR uses this information to establish RMR LSPs. For

instance, suppose R5 receives a FEC with prefix R0, RID 17 and ring flags AC. R5 knows that its clockwise path is R6 and anti-clockwise path is R4 to reach R0 and that the label map arrived from router R4 for a anti-clockwise LSP. Therefore, R5 selects the upstream session for this LSP as R6.

4.2 Ring Label Mapping Procedures

The procedures in the subsequent sections are organized by the role that a node plays to establish a ring LSP. Each node is egress for its own prefixes and transit for every prefix received with a Label Mapping message. Every transit node is also a ingress for that LSP.

4.2.1 Definitions

This section defines the notations for initiation, decoding, processing and propagation of RMR FEC Element.

1. RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$: a FEC Element with egress prefix P, RID R and clockwise direction C or anti-clockwise direction A.
2. RMR Label Mapping $\langle P, R, C, L \rangle$ or $\langle P, R, A, L \rangle$: a Label Mapping message with a FEC TLV with a single RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$ and Label TLV with label L. Label L MUST be allocated from the per-platform label space of the LSR sending the Label Mapping message. The use of the interface label space is outside the scope of this document.
3. RMR Label Withdraw $\langle P, R, C, L \rangle$ or $\langle P, R, A, L \rangle$: a Label Withdraw message with a FEC TLV with a single RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$ and Label TLV with label L.
4. RMR LSP $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$: A RMR LSP with egress prefix P, Ring ID R and clockwise direction C or anti-clockwise direction A.

4.2.2 Preliminary

A node X wishing to participate in LDP RMR signaling SHOULD negotiate the RMR capability with all its neighbors. When the IGP informs X of its RMR neighbors A and C for RID R, it MUST check that A and C have also negotiated the RMR capability with X. If these conditions are not satisfied, X cannot participate in signaling for ring R. This applies for all roles that X may play: ingress, transit and egress.

4.2.3 Egress LSR

Every ring node initiates two counter-rotating LSPs that egress on that node. After the IGP discovers the ring, LDP constructs the clockwise RMR FEC $\langle P, R, C \rangle$ and sends it in a Label Mapping message

to anti-clockwise neighbor. Similarly, LDP constructs a anti-clockwise RMR FEC <P, R, A> and sends it in a Label Mapping message to clockwise neighbor. This SHOULD establish a clockwise and anti-clockwise LSP - in terms of data traffic - in the clockwise and anti-clockwise direction respectively.

Furthermore, if a label other than implicit or explicit null is advertised for a LSP, LDP SHOULD add a pop route for this label in the Incoming Label Map (ILM) MPLS table.

When the node is no longer part of the ring, it SHOULD tear down its egress LSPs - CW and AC - by sending a label withdraw message.

4.2.4 Ingress and Transit LSR

When a transit LSR R5 depicted in figure 1 receives a label map message with RMR FEC Element <R0, 17, A, L1> from a downstream LDP session to R4, it SHOULD verify that R4 is indeed its anticlockwise neighbor for ring 17. If not, it SHOULD stop decoding the FEC TLV, abort processing the message containing the TLV, send an "Unknown FEC" Notification message to its LDP peer R4 signaling an error and close the session.

If the LSR encounters no other error while parsing the RMR FEC element, it allocates a Label L2 and determines a upstream LDP session as R6 using the algorithm described in section 'Upstream LSR'. It also programs a MPLS ILM table with label route L2 swapped to L1 and Ingress tunnel table with prefix R0 with label push L1 on all the LDP interfaces to R4, and sends the RMR FEC Element <R0, 17, A, L2> to R6.

If a session to the anti-clockwise neighbor for RID 17 depicted in Figure 2, namely R6, does not exist, the RMR FEC Element <R0, 17, A, L2> SHOULD not be propagated further. Similarly, when the upstream session changes because of ring topology change, transit LSR should send a label withdraw for RMR FEC Element <R0, 17, A, L2> to older upstream session R6 before sending Label Mapping message with RMR FEC Element <R0, 17, A, L2> to a new upstream session.

4.3 Equal Cost Multipath (ECMP)

A transit and ingress LSR of RMR LSP uses all the links between itself and downstream LSR to program transit and ingress route. Thus, ECMP works automatically for a LDP RMR LSP. A vendor could provide exception when necessary to this behavior by disabling certain ring links for RMR LSPs.

RMR uses the two counter-rotating LSPs to protect the other. Say that R5 wants to protect the LSP to R0 for RID 17. R5 receives RMR FEC Element <R0, 17, A, L1> from R4 and sends RMR FEC Element <R0, 17, A, L2> to R6. Then the primary path for the AC LSP is to swap L1 with L2 with next hop R4. Also, R5 receives RMR FEC Element <R0, 17, C, L3> from R6 and sends RMR FEC Element <R0, 17, C, L4> to R4. The primary path for the CW LSP is to swap L3 with L4. The protection path for the AC LSP is to swap L1 with L4 with next hop R6, thus sending the traffic back where it came from, but with a different label. The protection path for the CW LSP is to swap L3 with L2 with next hop R4.

```

      R9   R10  R11
      .     .     .
      .   .   .   .
      .           .
R8    .   .   .   R9
      .           .
      .           .
      .           .
R0    .   .   .   R1
      .               .
Anti-  R7                R2
Clockwise |              |
          |              |
          |              |
          v              v
          R6            R3
          .             .
          R5   .   .   R4

```

Suppose R5 needs to reach R10. Only RMR LSPs are setup inside the ring 17. Additionally, whenever services on R5 need to reach R10, R5 dynamically establishes a tLDP session to ring 17 master node R0 and R1. Further, suppose it only learns IPv4 and IPv6 FECs only over this session using [draft-ietf-mpls-app-aware-tldp-05]. Thus, in order to reach R10, R5 uses top label as RMR LSP label to R0 or R1 and bottom label as R10's FEC label received over tLDP session of R0 or R1 respectively.

6. Ring LSPs

An RMR LSP consists of two counter-rotating ring LSPs that start and end at the same node, say R1. As such, this appears to cause a loop, something that is normally to be avoided by LDP [RSVP-TE]. There are some benefits to this. Having a ring LSP allows the anchor node R1 to ping itself and thus verify the end-to-end operation of the LSP. This, in conjunction with link-level OAM, offers a good indication of the operational state of the LSP. [Also, having R1 be the ingress means that R1 can initiate the Path messages for the two ring LSPs. This avoids R1 having to coordinate with its neighbors to signal the LSPs, and simplifies the case where a ring update changes R1's ring neighbors.] The cost of this is a little more signaling and a couple more label entries in the LFIB. However, we will let implementation guide us to the wisdom of this approach.

7. Security Considerations

The Capability and RMR FEC procedures described in this document will not introduce any change to LDP Security Considerations section described in [RFC5036].

8. IANA Considerations

This document requires the assignment of a new code point for a Capability Parameter TLVs from the IANA managed LDP registry "TLV Type Name Space", corresponding to the advertisement of the RMR capability. IANA is requested to assign the lowest available value.

Value	Description	Reference
-----	-----	-----
TBD1	RMR capability	[this document]

This document requires the assignment of a new code point for a FEC type from the IANA managed LDP registry "Forwarding Equivalence Class (FEC) Type Name Space". IANA is requested to assign the lowest available value.

Value	Description	Reference
-----	-----	-----
TBD1	RMR FEC type	[this document]

9. Acknowledgments

TODO.

10. Contributors

Raveendra Torvi
Juniper Networks
10 Technology Park Dr
Westford, MA 01886
USA
Email: rtorvi@juniper.net

Ravi Singh
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: ravis@juniper.net

Abhishek Deshmukh
Juniper Networks
10 Technology Park Dr
Westford, MA 01886
USA
Email: adeshmukh@juniper.net

11. References

11.1 Normative References

- [I-D.ietf-mpls-rmr] Kompella, K. and L. Contreras, "Resilient MPLS Rings", draft-ietf-mpls-rmr-02 (work in progress), July 2016.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, October 2007, <<http://www.rfc-editor.org/info/rfc5036>>.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009, <<http://www.rfc-editor.org/info/rfc5561>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

11.2 Informative References

- [RFC6388] IJ. Wijnands, I. Minei, K. Kompella, B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011, <<http://www.rfc-editor.org/info/rfc6388>>
- [I-D.draft-deshmukh-rsvp-rmr-extension] A. Deshmukh, K. Kompella, "RSVP Extensions for RMR", draft-deshmukh-rsvp-rmr-extension-00 (work in progress), July 2016.
- [I-D.draft-ietf-mpls-app-aware-tldp] Santosh Esale, Raveendra Torvi, Luay Jalil, U. Chunduri, Kamran Raza, "Application-aware

Targeted LDP", draft-ietf-mpls-app-aware-tldp-05 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

Authors' Addresses

Santosh Esale
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA
Email: sesale@juniper.net

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA
Email: kireeti@juniper.net

MPLS Working Group
INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: September 14, 2017

Santosh Esale
Raveendra Torvi
Juniper Networks

Luyuan Fang
Microsoft

Luay Jalil
Verizon

March 13, 2017

Fast Reroute for Node Protection in LDP-based LSPs
draft-esale-mpls-ldp-node-frr-05

Abstract

This document describes procedures to support node protection for unicast Label Switched Paths (LSPs) established by Label Distribution Protocol (LDP). In order to protect a node N, the Point of Local Repair (PLR) of N must discover the Merge Points (MPs) of node N such that traffic can be redirected to them in case of node N failure. Redirecting the traffic around the failed node N depends on existing point-to-point LSPs originated from the PLR to the MPs while bypassing the protected node N. The procedures described in this document are topology independent in a sense that they provide node protection in any topology so long as there is an alternate path in the network that avoids the protected node.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at

<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1 Abbreviations	4
3. Merge Point (MP) Discovery	4
4. Constructing Bypass LSPs	5
5. Obtaining Label Mapping from MP	6
6. Forwarding Considerations	6
7. Synergy with node protection in mLDP	7
8. Security Considerations	7
9. IANA Considerations	7
10. Acknowledgements	7
11. Normative References	7
12. Informative References	7
Authors' Addresses	8

1. Introduction

This document describes procedures to support node protection for unicast Label Switched Paths (LSPs) established by Label Distribution Protocol (LDP) [RFC5036]. In order to protect a node N, the Point of Local Repair (PLR) of N must discover the Merge Points (MPs) of node N such that traffic can be redirected to them in case of node N

failure. Redirecting the traffic around the failed node N depends on existing explicit path Point-to-Point (P2P) LSPs originated from the PLR LSR to the MPs while bypassing node N. The procedures to setup these P2P LSPs are outside the scope of this document, but one option is to use RSVP-TE based techniques [RFC3209] to accomplish it. Finally, sending traffic from the PLR to the MPs requires the PLR to obtain FEC-label bindings from the MPs. The procedures described in this document relies on Targeted LDP (tLDP) session [RFC5036] for the PLR to obtain such FEC-Label bindings.

The procedure described in this document assumes the use of platform-wide label space. The procedures for node protection described in this document fall into the category of local protection. The procedures described in this document apply to LDP LSPs bound to either an IPv4 or IPv6 Prefix FEC element. The procedures described in this document are topology independent in a sense that they provide node protection in any topology so long as there is a alternate path in the network that avoids the protected node. Thus these procedures provide topology independent fast reroute.

1.1 Abbreviations

PLR: Point of Local Repair - the LSR that redirects the traffic to one or more Merge Point LSRs.

MP: Merge Point. Any LSR on the LDP-signaled (multi-point to point) LSP, provided that the path from that LSR to the egress of that LSP is not affected by the failure of the protected node.

tLDP: A targeted LDP session is an LDP session between non-directly connected LSRs, established using the LDP extended discovery mechanism.

FEC: Forwarding equivalence class.

IGP: Interior Gateway Protocol.

BR: Border Router.

3. Merge Point (MP) Discovery

For a given LSP that traverses the PLR, the protected node N, and a particular neighbor of the protected node, we'll refer to this neighbor as the "next next-hop". Note that from the PLR's perspective the protected node N is the next hop for the FEC associated with that LSP. Likewise, from the protected node's perspective the next next-hop is the next hop for that FEC. If for a given <LSP, PLR, N> triplet the next next-hop is in the same routing subdomain (area) as the PLR, then that next next-hop acts as the MP for that triplet. For a given LSP traversing a PLR and the node protected by the PLR, the PLR discovers its next next-hops (MPs) that are in the same routing subdomain (IGP area) as the PLR from IGP shortest path first (SPF) calculations. The discovery of next next-hop, depending on an implementation, may not involve any additional SPF, above and beyond what will be needed by either ISIS or OSPF anyway, as the next next-hop, just like the next-hop, is a by-product of SPF computation.

Also, the PLR may discover all possible MPs from either its traffic engineering database or link state database. Some implementations MAY need appropriate configuration to populate the traffic engineering database. The traffic engineering database is populated by routing protocols such as ISIS and OSPF or configured statically.

If for a given <LSP, PLR, N> triplet the node protected by the PLR is an Border Router (BR), then the PLR and the next next-hop may end up in different routing subdomain. This could happen when an LSP

traversing the PLR and the protected node does not terminate in the same routing subdomain as the PLR. In this situation the PLR may not be able to determine the next next-hop from shortest path first (SPF) calculations, and thus may not be able to use the next next-hop as the MP. In this scenario the PLR uses an "alternative" BR as the MP, where an alternative BR is defined as follows. For a given LSP that traverses the PLR and the (protected) BR, an alternative BR is defined as any BR that advertises into PLR's own routing subdomain reachability to the FEC associated with the LSP.

Note that even if a PLR protects an BR, for some of the LSPs traversing the PLR and the BR, the next next-hops may be in the same routing subdomain as the PLR, in which case these next next-hops act as MPs for these LSPs. Note that even if the protected node is not an BR, if an LSP traversing the PLR and the protected node does not terminate in the same routing subdomain as the PLR, then for this LSP the PLR MAY use an alternative BR (as defined earlier), rather than the next next-hop as the MP. When there are several candidate BRs for alternative BR, the LSR MUST select one BR. The algorithm used for the alternative BR selection is a local matter but one option is to select the BR per FEC based on shortest path from PLR to the BR.

4. Constructing Bypass LSPs

As mentioned before, redirecting traffic around the failed node N depends on existing explicit path Point-to-Point (P2P) LSPs originated from the PLR to the MPs while bypassing node N. Let's refer to these LSPs as "bypass LSPs". While the procedures to signal these bypass LSPs are outside the scope of this document, this document assumes use of RSVP-TE LSPs [RFC3209] to accomplish it. Once a PLR that protects a given node N discovers the set of MPs associated with itself and the protected node, at the minimum the PLR MUST (automatically) establish bypass LSPs to all these MPs. The bypass LSPs MUST be established prior to the failure of the protected node.

One could observe that if the protected node is not an BR and the PLR does not use alternative BR(s) as MP(s), then the set of all the IGP neighbors of the protected node forms a superset of the MPs. Thus it would be sufficient for the PLR to establish bypass LSPs with all the IGP neighbors of the protected node, even though some of these neighbors may not be MPs for any of the LSPs traversing the PLR and the protected node.

The bypass LSPs MUST avoid traversing the protected node, which means that the bypass LSPs are explicitly routed LSPs. Of course, using

RSVP-TE to establish bypass LSPs allows these LSPs to be explicitly routed. As a given router may act as an MP for more than one LSP traversing the PLR, the protected node, and the MP, the same bypass LSP will be used to protect all those LSPs.

5. Obtaining Label Mapping from MP

As mentioned before, sending traffic from the PLR to the MPs requires the PLR to obtain FEC-label bindings from the MPs. The solution described in this document relies on Targeted LDP (tLDP) session [RFC5036] for the PLR to obtain such mappings. Specifically, for a given PLR and the node protected by this PLR, at the minimum the PLR MUST (automatically) establish tLDP with all the MPs associated with this PLR and the protected node. These tLDP sessions MUST be established prior to the failure of the protected node. One could observe that if the protected node is not an BR and the PLR does not use alternative BR(s) as MP(s), then the set of all the IGP neighbors of the protected node forms a superset of the MPs. Thus it will be sufficient for the PLR to (automatically) establish tLDP session with all the IGP neighbors of the protected node - except the PLR - that are in the same area as the PLR, even though some of these neighbors may not be MPs for any of the LSPs traversing the PLR and the protected node.

At the minimum for a given tLDP peer the PLR MUST obtain FEC-label mapping for the FEC(s) for which the peer acts as an MP. The PLR MUST obtain this mapping before the failure of the protected node. To obtain this mapping for only these FECs and no other FECs that the peer may maintain, the PLR SHOULD rely on the LDP Downstream on Demand (DoD) procedures [RFC5036]. Otherwise, without relying on the DoD procedures, the PLR may end up receiving from a given tLDP peer FEC-label mappings for all the FECs maintained by the peer, even if the peer does not act as an MP for some of these FECs. If the LDP DoD procedures are not used, then for the purpose of the procedures specified in this draft the only label mappings that SHOULD be exchanged are for the Prefix FEC elements whose PreLen value is either 32 (IPv4), or 128 (IPv6); label mappings for the Prefix FEC elements with any other PreLen value SHOULD NOT be exchanged.

When a PLR has one or more BRs acting as MPs, the PLR MAY use the procedures specified in [draft-ietf-mpls-app-aware-tldp] to limit the set of FEC-label mappings received from non-BR MPs to only the mappings for the FECs associated with the LSPs that terminate in the PLR's own routing subdomain (area).

6. Forwarding Considerations

When a PLR detects failure of the protected node then rather than

swapping an incoming label with a label that the PLR received from the protected node, the PLR swaps the incoming label with the label that the PLR receives from the MP, and then pushes the label associated with the bypass LSP to that MP.

To minimize micro-loop during the IGP global convergence PLR may continue to use the bypass LSP during network convergence by adding small delay before switching to a new path.

7. Synergy with node protection in mLDP

Both the bypass LSPs and tLDP sessions described in this document could also be used for the purpose of mLDP node protection, as described in [draft-ietf-mpls-mldp-node-protection].

8. Security Considerations

The same security considerations apply as those for the base LDP specification, as described in [RFC5036].

9. IANA Considerations

This document introduces no new IANA Considerations.

10. Acknowledgements

We are indebted to Yakov Rekhter for many discussions on this topic. We like to thank Hannes Gredler, Aman Kapoor, Minto Jeyananth, Eric Rosen, Vladimir Blazhkun and Loa Andersson for through review of this document.

11. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3209] D. Awduche, et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC3209, Decembet 2001.
- [RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.
- [draft-ietf-mpls-app-aware-tldp] Esale, S., et al., "Application-aware Targeted LDP", draft-esale-mpls-app-aware-tldp, work in progress.

12. Informative References

[RFC7715], IJ. Wijnands, et al., "Multipoint LDP (mLDP) Node Protection", RFC7715, January 2016.

Authors' Addresses

Santosh Esale
Juniper Networks
EMail: sesale@juniper.net

Raveendra Torvi
Juniper Networks
EMail: rtorvi@juniper.net

Luyuan Fang
Microsoft
Email: lufang@microsoft.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: November 21, 2018

Santosh Esale
Kireeti Kompella
Juniper Networks
May 20, 2018

LDP Extensions for RMR
draft-esale-mpls-ldp-rmr-extensions-02

Abstract

This document describes LDP extensions to signal Resilient MPLS Ring (RMR) Label Switched Paths (LSPs). An RMR LSP is a multipoint to point LSP signaled using LDP (Label Distribution Protocol). RMR Architecture document - draft-ietf-mpls-rmr-02 - describes why and how MPLS should be used in ring topologies.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
2.	Terminology	3
3.	Protocol extensions	4
3.1.	Ring LSP Capability	4
3.2.	Ring FEC Element	4
4.	Ring Procedures	6
4.1	Upstream LSR	6
4.2	Ring Label Mapping Procedures	7
4.2.1	Definitions	7
4.2.2	Preliminary	7
4.2.3	Egress LSR	7
4.2.4	Ingress and Transit LSR	8
4.3	Equal Cost Multipath (ECMP)	8
4.4	Protection	8
5.	LSP Hierarchy	9
6.	Ring LSPs	9
7.	Security Considerations	11
8.	IANA Considerations	11
9.	Acknowledgments	11
10.	Contributors	11
11.	References	12
11.1	Normative References	12
11.2	Informative References	12
	Authors' Addresses	13

1 Introduction

This document describes LDP extensions to signal resilient MPLS ring (RMR) label switched paths (LSPs). An RMR LSP is a multipoint to point LSP signaled using LDP (Label Distribution Protocol). Architecture document of RMR - draft-ietf-mpls-rmr-02 - describes why and how MPLS should be used in ring topologies.

The ring is either auto-discovered or configured using IGP protocol such as OSPF or ISIS. IGP extensions for RMR is described in a companion documents. After the ring discovery, each ring node acting as egress constructs and signals a clockwise (CW) and anti-clockwise (AC) ring FEC towards AC and CW direction respectively. Each transit node that receives the RMR FEC signals this LSP further in same direction using RMR link state database. In addition, it also adds a transit and ingress route for this LSP. Once the signaling is complete, every node in a ring should have two counter rotating LSPs in CW and AC direction to reach every other node on the ring.

2. Terminology

A ring consists of a subset of n nodes $\{R_i, 0 \leq i < n\}$. The direction from node R_i to R_{i+1} is defined as "clockwise" (CW) and the reverse direction is defined as "anti-clockwise" (AC). As there may be several rings in a graph, each ring is numbered with a distinct ring ID (RID).

The following terminology is used for ring LSPs:

Ring ID (RID): A non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network. A node may belong to multiple rings.

Ring node: A member of a ring. Note that a device may belong to several rings.

Node index: A logical numbering of nodes in a ring, from zero upto one less than the ring size. Used purely for exposition in this document.

Ring neighbors: Nodes whose indices differ by one (modulo ring size).

Ring links: Links that connect ring neighbors.

Express links: Links that connect non-neighboring ring nodes.

MP2P LSP: Each LSP in the ring is a multipoint to point LSP such that LSP can have multiple ingress nodes and one egress node.

3. Protocol extensions

This section describes LDP extensions to signal RMR LSP in a ring.

3.1. Ring LSP Capability

RMR LSPs support for a LSR is advertised using LDP capabilities as defined in [RFC5561]. An implementation that supports the RMR procedures specified in this document MUST add the procedures pertaining to Capability Parameters for Initialization messages.

A new optional capability parameter TLV, RMR Capability, is defined. Following is the format of the RMR Capability Parameter:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|U|F| RMR Capability (TBD) | Length (= 1) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|S| Reserved |
+---+---+---+---+---+

```

As described in [RFC5561]

U: set to 1. Ignore, if not known.

F: Set to 0. Do not forward.

S: MUST be set to 1 to advertise the RMR Capability TLV.

The RMR Capability TLV MUST be advertised in the LDP Initialization message. If the peer has not advertised the RMR capability, then label messages pertaining to RMR FEC Element MUST NOT be sent to the peer.

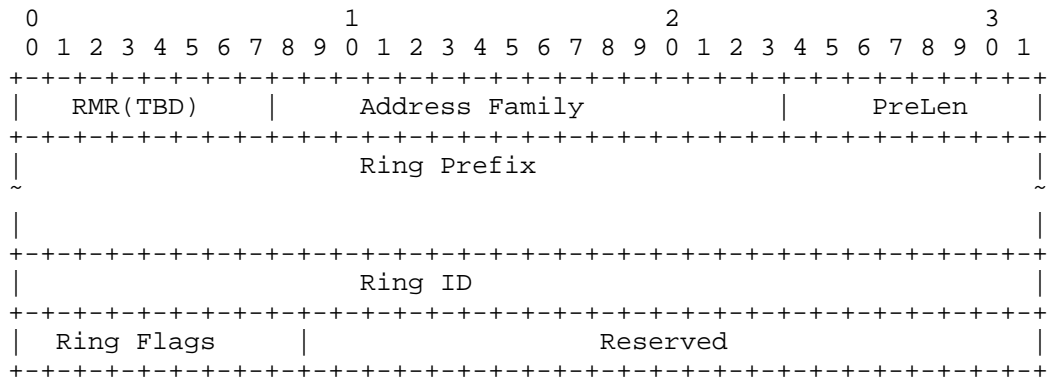
3.2. Ring FEC Element

In order to setup RMR LSP in clockwise and anti-clockwise direction for every ring node, this document defines new protocol entity, the RMR FEC Element, to be used as a FEC Element in the FEC TLV.

The RMR FEC Element consists of the ring address, ring identifier and ring flags which depicts ring direction. The combination of ring address, ring identifier and ring flags uniquely identifies a ring

LSP within the MPLS network.

The RMR FEC Element value encoding is as follows:



FEC Type

One octet quantity containing a value from FEC Type Name Space that encodes the fec type for a RMR LDP LSP.

Address Family

Two octet quantity containing a value from ADDRESS FAMILY NUMBERS in [ASSIGNED_AF] that encodes the address family for the address prefix in the Prefix field.

PreLen

One octet unsigned integer containing the length in bits of the address prefix that follows. A length of zero indicates a prefix that matches all addresses (the default destination); in this case, the Prefix itself is zero octets).

Prefix

An address prefix encoded according to the Address Family field, whose length, in bits, was specified in the PreLen field, padded to a byte boundary.

Ring ID (RID)

A four-octet non-zero number that identifies a ring; this is unique in some scope of a Service Provider's network.

Ring Flags

0	1	2	3	4	5	6	7

```

|RF | Reserved |
+-----+
The value of ring flags (RF) is defined as follows:
1: Clockwise (CW) FEC
2: Anti-clockwise (AC) FEC

```

4. Ring Procedures

This section describes LDP procedures to signal RMR LSP in a ring.

4.1 Upstream LSR

Upstream LSR for RMR LSP is selected as follows:

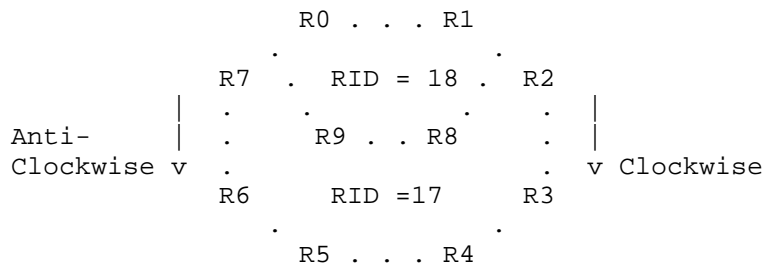


Figure 1: Two Rings with 10 nodes

Consider a MPLS ring with 10 nodes. During the discovery of this ring, IGP populates its link state database with ring information. After the discovery, there are just two paths - one in clockwise direction and other in anti-clockwise direction - for every ring neighbor on a specific ring. For instance, the following table shows router R0's path for ring 17 and 18 depicted in figure 1.

RID	CW neighbor	AC neighbor
17	R1	R7
18	R1	R9

Figure 2: R0's RMR upstream signaling table

IGP informs LDP that a new MPLS ring, RID 17, is discovered. A LDP transit LSR uses this information to establish RMR LSPs. For instance, suppose R5 receives a FEC with prefix R0, RID 17 and ring flags AC. R5 knows that its clockwise path is R6 and anti-clockwise

path is R4 to reach R0 and that the label map arrived from router R4 for a anti-clockwise LSP. Therefore, R5 selects the upstream session for this LSP as R6.

4.2 Ring Label Mapping Procedures

The procedures in the subsequent sections are organized by the role that a node plays to establish a ring LSP. Each node is egress for its own prefixes and transit for every prefix received with a Label Mapping message. Every transit node is also a ingress for that LSP.

4.2.1 Definitions

This section defines the notations for initiation, decoding, processing and propagation of RMR FEC Element.

1. RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$: a FEC Element with egress prefix P, RID R and clockwise direction C or anti-clockwise direction A.
2. RMR Label Mapping $\langle P, R, C, L \rangle$ or $\langle P, R, A, L \rangle$: a Label Mapping message with a FEC TLV with a single RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$ and Label TLV with label L. Label L MUST be allocated from the per-platform label space of the LSR sending the Label Mapping message. The use of the interface label space is outside the scope of this document.
3. RMR Label Withdraw $\langle P, R, C, L \rangle$ or $\langle P, R, A, L \rangle$: a Label Withdraw message with a FEC TLV with a single RMR FEC Element $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$ and Label TLV with label L.
4. RMR LSP $\langle P, R, C \rangle$ or $\langle P, R, A \rangle$: A RMR LSP with egress prefix P, Ring ID R and clockwise direction C or anti-clockwise direction A.

4.2.2 Preliminary

A node X wishing to participate in LDP RMR signaling SHOULD negotiate the RMR capability with all its neighbors. When the IGP informs X of its RMR neighbors A and C for RID R, it MUST check that A and C have also negotiated the RMR capability with X. If these conditions are not satisfied, X cannot participate in signaling for ring R. This applies for all roles that X may play: ingress, transit and egress.

4.2.3 Egress LSR

Every ring node initiates two counter-rotating LSPs that egress on that node. After the IGP discovers the ring, LDP constructs the clockwise RMR FEC $\langle P, R, C \rangle$ and sends it in a Label Mapping message to anti-clockwise neighbor. Similarly, LDP constructs a anti-clockwise RMR FEC $\langle P, R, A \rangle$ and sends it in a Label Mapping message

to clockwise neighbor. This SHOULD establish a clockwise and anti-clockwise LSP - in terms of data traffic - in the clockwise and anti-clockwise direction respectively.

Furthermore, if a label other than implicit or explicit null is advertised for a LSP, LDP SHOULD add a pop route for this label in the Incoming Label Map (ILM) MPLS table.

When the node is no longer part of the ring, it SHOULD tear down its egress LSPs - CW and AC - by sending a label withdraw message.

4.2.4 Ingress and Transit LSR

When a transit LSR R5 depicted in figure 1 receives a label map message with RMR FEC Element <R0, 17, A, L1> from a downstream LDP session to R4, it SHOULD verify that R4 is indeed its anticlockwise neighbor for ring 17. If not, it SHOULD stop decoding the FEC TLV, abort processing the message containing the TLV, send an "Unknown FEC" Notification message to its LDP peer R4 signaling an error and close the session.

If the LSR encounters no other error while parsing the RMR FEC element, it allocates a Label L2 and determines an upstream LDP session as R6 using the algorithm described in section 'Upstream LSR'. It also programs a MPLS ILM table with label route L2 swapped to L1 and Ingress tunnel table with prefix R0 with label push L1 on all the LDP interfaces to R4, and sends the RMR FEC Element <R0, 17, A, L2> to R6.

If a session to the anti-clockwise neighbor for RID 17 depicted in Figure 2, namely R6, does not exist, the RMR FEC Element <R0, 17, A, L2> SHOULD NOT be propagated further. Similarly, when the upstream session changes because of ring topology change, transit LSR should send a label withdraw for RMR FEC Element <R0, 17, A, L2> to older upstream session R6 before sending Label Mapping message with RMR FEC Element <R0, 17, A, L2> to a new upstream session.

4.3 Equal Cost Multipath (ECMP)

A transit and ingress LSR of RMR LSP uses all the links between itself and downstream LSR to program transit and ingress route. Thus, ECMP works automatically for a LDP RMR LSP. A vendor could provide exception when necessary to this behavior by disabling certain ring links for RMR LSPs.

4.4 Protection

RMR uses the two counter-rotating LSPs to protect the other. Say that R5 wants to protect the LSP to R0 for RID 17. R5 receives RMR FEC Element <R0, 17, A, L1> from R4 and sends RMR FEC Element <R0, 17, A, L2> to R6. Then the primary path for the AC LSP is to swap L1 with L2 with next hop R4. Also, R5 receives RMR FEC Element <R0, 17, C, L3> from R6 and sends RMR FEC Element <R0, 17, C, L4> to R4. The primary path for the CW LSP is to swap L3 with L4. The protection path for the AC LSP is to swap L1 with L4 with next hop R6, thus sending the traffic back where it came from, but with a different label. The protection path for the CW LSP is to swap L3 with L2 with next hop R4.

5. LSP Hierarchy

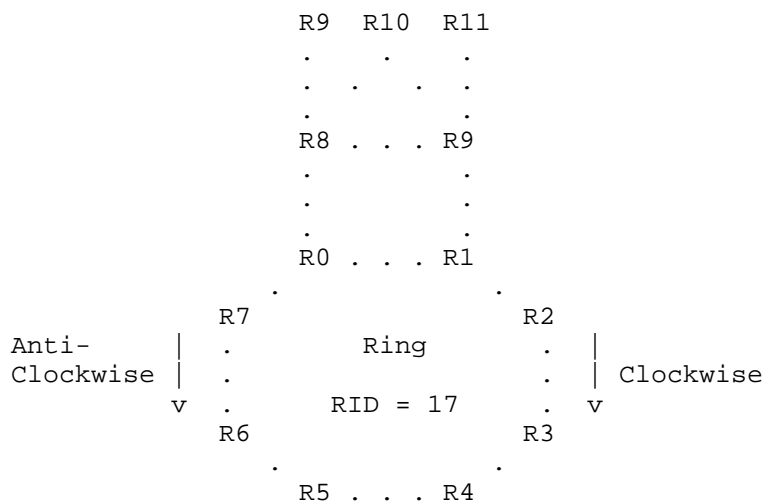


Figure 3: Ring 17 with rest of the Network

Suppose R5 needs to reach R10. Only RMR LSPs are setup inside the ring 17. Additionally, whenever services on R5 need to reach R10, R5 dynamically establishes a tLDP session to ring 17 master node R0 and R1. Further, suppose it only learns IPv4 and IPv6 FECs only over this session using [draft-ietf-mpls-app-aware-tldp-05]. Thus, in order to reach R10, R5 uses top label as RMR LSP label to R0 or R1 and bottom label as R10's FEC label received over tLDP session of R0 or R1 respectively.

6. Ring LSPs

An RMR LSP consists of two counter-rotating ring LSPs that start and end at the same node, say R1. As such, this appears to cause a loop, something that is normally to be avoided by LDP [RSVP-TE]. There are some benefits to this. Having a ring LSP allows the anchor node R1 to ping itself and thus verify the end-to-end operation of the LSP. This, in conjunction with link-level OAM, offers a good indication of the operational state of the LSP. [Also, having R1 be the ingress means that R1 can initiate the Path messages for the two ring LSPs. This avoids R1 having to coordinate with its neighbors to signal the LSPs, and simplifies the case where a ring update changes R1's ring neighbors.] The cost of this is a little more signaling and a couple more label entries in the LFIB.

7. Security Considerations

The Capability and RMR FEC procedures described in this document will not introduce any change to LDP Security Considerations section described in [RFC5036].

8. IANA Considerations

This document requires the assignment of a new code point for a Capability Parameter TLVs from the IANA managed LDP registry "TLV Type Name Space", corresponding to the advertisement of the RMR capability. IANA is requested to assign the lowest available value.

Value	Description	Reference
-----	-----	-----
TBD1	RMR capability	[this document]

This document requires the assignment of a new code point for a FEC type from the IANA managed LDP registry "Forwarding Equivalence Class (FEC) Type Name Space". IANA is requested to assign the lowest available value.

Value	Description	Reference
-----	-----	-----
TBD1	RMR FEC type	[this document]

9. Acknowledgments

TODO.

10. Contributors

Raveendra Torvi
Juniper Networks
10 Technology Park Dr
Westford, MA 01886
USA
Email: rtorvi@juniper.net

Ravi Singh
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: ravis@juniper.net

Abhishek Deshmukh
Juniper Networks
10 Technology Park Dr
Westford, MA 01886
USA
Email: adeshmukh@juniper.net

11. References

11.1 Normative References

- [I-D.ietf-mpls-rmr] Kompella, K. and L. Contreras, "Resilient MPLS Rings", draft-ietf-mpls-rmr-03 (work in progress), October 2016.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, October 2007, <<http://www.rfc-editor.org/info/rfc5036>>.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009, <<http://www.rfc-editor.org/info/rfc5561>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

11.2 Informative References

- [RFC6388] IJ. Wijnands, I. Minei, K. Kompella, B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, November 2011, <<http://www.rfc-editor.org/info/rfc6388>>
- [I-D.draft-deshmukh-mpls-rsvp-rmr-extension] A. Deshmukh, K. Kompella, "RSVP Extensions for RMR", draft-deshmukh-mpls-rsvp-rmr-extension-00 (work in progress), July 2016.
- [I-D.draft-kompella-isis-ospf-rmr] K. Kompella, "IGP Extensions for Resilient MPLS Rings", draft-kompella-isis-ospf-rmr-00

(work in progress), October 2016.

[I-D.draft-ietf-mpls-app-aware-tldp] Santosh Esale, Raveendra Torvi, Luay Jalil, U. Chunduri, Kamran Raza, "Application-aware Targeted LDP", draft-ietf-mpls-app-aware-tldp-05 (work in progress), June 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

Authors' Addresses

Santosh Esale
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA
Email: sesale@juniper.net

Kireeti Kompella
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA
Email: kireeti@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

G.Galimberti, Ed.
Cisco
R.Kunze
Deutsche Telekom
D. Hiremagalur, Ed.
G. G.Grammel, Ed.
Juniper
March 13, 2017

A YANG model to manage the optical parameters for in a WDM
network
draft-galimbe-ccamp-iv-yang-02

Abstract

This memo defines a Yang model that translate the information model to support Impairment-Aware (IA) Routing and Wavelength Assignment (RWA) functionality. The information model is defined in draft-ietf-ccamp-wson-iv-info and draft-martinelli-ccamp-wson-iv-encode. This document defines proper encoding and extend to the models defined in draft-lee-ccamp-wson-yang to support Impairment-Aware (IA) Routing and Wavelength Assignment (RWA) functions

The Yang model defined in this memo can be used for Optical Parameters monitoring and/or configuration of the multivendor Endpoints and ROADMs

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. The Internet-Standard Management Framework	3
3. Conventions	4
4. Definition	4
5. Applicability	4
6. Properties	4
7. Overview	4
7.1. Optical Parameters Description	5
7.1.1. Optical path from point Ss to Rs	6
7.1.2. Rs and Ss Configuration	7
7.1.3. Table of Application Codes	7
7.2. Use Cases	7
7.3. Optical Parameters for impairment validation in a WDM network	7
8. Structure of the Yang Module	8
9. Yang Module	8
10. Security Considerations	17
11. IANA Considerations	17
12. Acknowledgements	18
13. Contributors	18
14. References	18
14.1. Normative References	18
14.2. Informative References	20
Appendix A. Change Log	21
Appendix B. Open Issues	21
Authors' Addresses	21

1. Introduction

This memo defines a Yang model that translates the existing mib module defined in draft-ietf-ccamp-wson-iv-info and draft-martinelli-ccamp-wson-iv-encode to provide the network impairment information to an SDN controller. One of the key SDN controller features is to support multi vendor network and support the service calculation and deployment in multilayer topologies, for the DWDM layer it is fundamental the SDN controller is aware of the optical impairments to verify the feasibility of new circuits before their provisioning. Although SDN controller will not apply exhaustive and accurate algorithms and the optical channel feasibility verification may have a degree of unreliability this function can work on a multivendor common set of parameter and algorithms to ensure the operator the best change to set a circuit. This document follows the same impairment definition and applicability of draft-ietf-ccamp-wson-iv-info.

For the optical impairments related to the DWDM Transceiver the draft draft-dharini-ccamp-if-param-yang. Applications are defined in G.698.2 [ITU.G698.2] using optical interface parameters at the single-channel connection points between optical transmitters and the optical multiplexer, as well as between optical receivers and the optical demultiplexer in the DWDM system. This Recommendation uses a methodology which explicitly specify the details of the optical network between reference point Ss and Rs, e.g., the passive and active elements or details of the design.

This draft refers and supports the draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

The building of a yang model describing the optical parameters allows the different vendors and operator to retrieve, provision and exchange information across multi-vendor domains in a standardized way. In addition to the parameters specified in ITU recommendations the Yang models support also the "vendor specific parameters".

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

This memo specifies a Yang model for optical interfaces.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. In the description of OIDs the convention: Set (S) Get (G) and Trap (T) conventions will describe the action allowed by the parameter.

4. Definition

For a detailed definition this draft refers to draft-ietf-ccamp-wson-iv-info.

5. Applicability

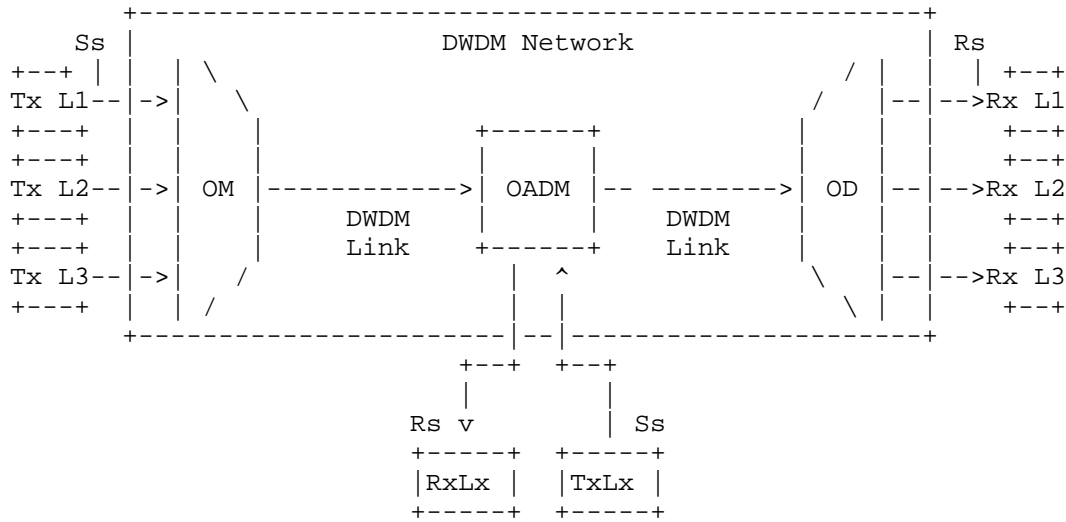
This document targets at Scenario C defined in [RFC6566] section 4.1.1. as approximate impairment estimation. The Approximate concept refers to the fact that this Information Model covers information mainly provided by [ITU.G680] Computational Model. Although the [RFC6566] provides no or little approximation the parameters described in this draft can be applied to the algorithms verifying the circuit feasibility in the new coherent non compensated DWDM networks. In this case the impairments verification can reach a good reliability and accuracy. This draft does not address computational matters but provides all the information suitable to cover most of the full coherent network algorithms, not being exhaustive the information can give a acceptable or even good approximation in terms of connection feasibility. This may not be true for legacy compensated network.

6. Properties

For the signal properties this draft refers to draft-ietf-ccamp-wson-iv-info Ch.2.3 with some extension of the parameters.

7. Overview

Figure 1 shows a set of reference points, for single-channel connection between transmitters (Tx) and receivers (Rx). Here the DWDM network elements include an OM and an OD (which are used as a pair with the opposing element), one or more optical amplifiers and may also include one or more OADMs.



Ss = reference point at the DWDM network element tributary output
Rs = reference point at the DWDM network element tributary input
Lx = Lambda x
OM = Optical Mux
OD = Optical Demux
OADM = Optical Add Drop Mux

from Fig. 5.1/G.698.2

Figure 1: External transponder in WDM networks

7.1. Optical Parameters Description

The link between the external transponders through a WDM network media channels are managed at the edges, i.e. at the transmitters (Tx) and receivers (Rx) attached to the S and R reference points respectively. The set of parameters that could be managed are defined by the "application code" notation

The definitions of the optical parameters are provided below to increase the readability of the document, where the definition is

ended by (R) the parameter can be retrieve with a read, when (W) it can be provisioned by a write, (R,W) can be either read or written.

7.1.1.1. Optical path from point Ss to Rs

The following parameters for the optical path from point S and R are defined in G.698.2 [ITU.G698.2].

Maximum and minimum (residual) chromatic dispersion:

These parameters define the maximum and minimum value of the optical path "end to end chromatic dispersion" (in ps/nm) that the system shall be able to tolerate. (R)

Minimum optical return loss at Ss:

These parameter defines minimum optical return loss (in dB) of the cable plant at the source reference point (Ss), including any connectors (R)

Maximum discrete reflectance between Ss and Rs:

Optical reflectance is defined to be the ratio of the reflected optical power present at a point, to the optical power incident to that point. Control of reflections is discussed extensively in ITU-T Rec. G.957 (R)

Maximum differential group delay:

Differential group delay (DGD) is the time difference between the fractions of a pulse that are transmitted in the two principal states of polarization of an optical signal. For distances greater than several kilometres, and assuming random (strong) polarization mode coupling, DGD in a fibre can be statistically modelled as having a Maxwellian distribution. (R)

Maximum polarization dependent loss:

The polarization dependent loss (PDL) is the difference (in dB) between the maximum and minimum values of the channel insertion loss (or gain) of the black link from point SS to RS due to a variation of the state of polarization (SOP) over all SOPs. (R)

Maximum inter-channel crosstalk:

Inter-channel crosstalk is defined as the ratio of total power in all of the disturbing channels to that in the wanted channel, where the wanted and disturbing channels are at different wavelengths. The parameter specify the isolation of a link conforming to the "black link" approach such that under the worst-case operating conditions the inter-channel crosstalk at any reference point RS is less than the maximum inter-channel crosstalk value (R)

Maximum interferometric crosstalk:

This parameter places a requirement on the isolation of a link conforming to the "black link" approach such that under the worst case operating conditions the interferometric crosstalk at any reference point RS is less than the maximum interferometric crosstalk value. (R)

Maximum optical path OSNR penalty:

The optical path OSNR penalty is defined as the difference between the Lowest OSNR at Rs and Lowest OSNR at Ss that meets the BER requirement (R)

Maximum ripple:

Although is defined in G.698.2 (R).

7.1.2. Rs and Ss Configuration

For the Rs and Ss configuration this draft refers the draft-dharini-ccamp-dwdm-if-param-yang while for the Rs-Ss extended parameters for coherent transmission interfaces refer to draft-dharini-ccamp-dwdm-if-param-yang

7.1.3. Table of Application Codes

For Application Codes configuration this draft refers the draft-dharini-ccamp-dwdm-if-param-yang

7.2. Use Cases

The use cases are described in draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

7.3. Optical Parameters for impairment validation in a WDM network

The ietf-opt-parameters-wdm is an augment to the ?????. It allows the user to get and set the application Optical Parameters of a DWDM network.

```

module: ietf-opt-parameters-wdm
+--rw gain-degrade-high? dbm-t
augment /if:interfaces/if:interface:
+--rw optical-transport
|   +--rw attenuator-value? attenuator-t
|   +--rw offset? decimal64
|   +--rw channel-power-ref? decimal64
|   +--rw tilt-calibration? tilt-t
+--rw channel-t
|   +--rw grid? uint32
|   +--rw channel-spacing? uint32
|   +--rw identifier? uint32
|   +--rw n? uint32
+--rw channel-n-m
|   +--rw grid? uint32
|   +--rw channel-spacing? uint32
|   +--rw n? uint32
|   +--rw m? uint32

```

8. Structure of the Yang Module

ietf-opt-parameters-wdm is a top level model for the support of this feature.

9. Yang Module

The ietf-opt-parameters-wdm is defined as an extension to ietf interfaces ????.

<CODE BEGINS> file "ietf-opt-parameters-wdm.yang"

```

module ietf-opt-parameters-wdm {
  namespace "urn:ietf:params:xml:ns:yang:ietf-opt-parameters-wdm";
  prefix iietf-opt-parameters-wdm;

  import ietf-interfaces {
    prefix if;
  }

  import iana-if-type {
    prefix ianaift;
  }

```

```
organization
  "IETF CCAMP
  Working Group";
```

```
contact
  "WG Web:    <http://tools.ietf.org/wg/ccamp/>
  WG List:    <mailto:ccamp@ietf.org>
```

```
Editor:    Gabriele Galimberti
           <mailto:ggalimbe@cisco.com>;
```

```
description
  "This module contains a collection of YANG definitions for
  collecting and configuring Optical Parameters
  in Optical Networks and calculate the circuit feasibility.
```

```
Copyright (c) 2016 IETF Trust and the persons identified
as authors of the code. All rights reserved.
```

```
Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and
subject to the license terms contained in, the Simplified
BSD License set forth in Section 4.c of the IETF Trust's
Legal Provisions Relating to IETF Documents
(http://trustee.ietf.org/license-info).";
```

```
revision "2016-10-30" {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for Optical Paramenters
    of DWDM Networks
    ";
}
```

```
typedef tilt-t {
  type decimal64 {
    fraction-digits 2;
    range "-5..5";
  }
  description "Tilt Type";
}
```

```
typedef signal-output-power-t {
```

```
        type decimal64 {
            fraction-digits 2;
            range "-10..30";
        }
        description "
            Amplifier Power provisioning ";
    }

    typedef active-channel-t {
        type union {
            type uint8 {
                range "0..200";
            }
        }
        description "
            Number of channels active on a span - and on an amplifier";
    }

    typedef dbm-t {
        type decimal64 {
            fraction-digits 2;
            range "-50..-30 | -10..5 | 10000000";
        }
        description "
            Amplifier Power in dBm ";
    }

    typedef attenuator-t {
        type decimal64 {
            fraction-digits 2;
            range "-15..-5";
        }
        description "
            Attenuation value (attenuator) applied after the Amplifier";
    }

    augment "/if:interfaces/if:interface" {
        when "if:type = 'ianaift:opticalTransport'" {
            description "Specific optical-transport Interface Data";
        }
        description "Specific optical-transport Interface Data";
        container optical-transport {
            description "Specific optical-transport Data";

            leaf attenuator-value {
                type attenuator-t;
                description "External attenuator value ";
            }
        }
    }
```

```
    leaf offset {
      type decimal64 {
        fraction-digits 2;
        range "-30..30";
      }
      description "Raman and power amplifiers offset";
    }

    leaf channel-power-ref {
      type decimal64 {
        fraction-digits 2;
        range "-10..15";
      }
      description "Optical power per channel";
    }

    leaf tilt-calibration {
      type tilt-t;
      description "Amplifier Tilt tuning";
    }
  }
}

grouping opwr-threshold-warning-grp {
  description "
    Minimum Optical Power threshold
    - this is used to rise Power alarm ";

  leaf opwr-min {
    type dbm-t;
    units "dBm";
    default -1;
    description "Minimum Power Value";
  }

  leaf opwr-min-clear {
    type dbm-t;
    units "dBm";
    default -1;
    description "threshold to clear Minimum Power value Alarm";
  }

  leaf opwr-max {
    type dbm-t;
    units "dBm";
    default 1;
    description "
      Maximum Optical Power threshold
    "
```

```
        - this is used to rise Power alarm ";
    }
}

grouping gain-degrade-alarm-grp {
    description "
        Low Optical Power gain threshold
        - this is used to rise Power alarm ";

    leaf gain-degrade-low {
        type dbm-t;
        units "dBm";
        default -1;
        description "Low Gain Degrade Value";
    }
}

leaf gain-degrade-high {
    type dbm-t;
    units "dBm";
    default 1;
    description "
        High Optical Power gain threshold
        - this is used to rise Power alarm ";
}
}

grouping power-degrade-high-alarm-grp {
    description "
        High Optical Power gain alarm ";

    leaf gain-degrade-high {
        type dbm-t;
        units "dBm";
        default 1;
        description "Low Gain Degrade Value";
    }
}

grouping power-degrade-low-alarm-grp {
    description "
        Low Optical Power gain alarm ";

    leaf power-degrade-low {
        type dbm-t;
        units "dBm";
        default -1;
        description "High Gain Degrade Value";
    }
}
```



```
    }  
  }  
  
  grouping noise {  
    leaf noise {  
      type decimal64 {  
        fraction-digits 2;  
      }  
      units "dB";  
      description "Noise feasibility - reference ITU-T G.680  
        OSNR added to the signal by the OMS. The noise is intended  
        per channel and is independent of the number of active  
        channels in OMS";  
    }  
    description "Noise feasibility";  
  }  
  
  grouping noise-sigma {  
    leaf noise-sigma {  
      type decimal64 {  
        fraction-digits 2;  
      }  
      units "dB";  
      description "Noise Sigma feasibility - accuracy of the  
        OSNR added to  
        the signal by the OMS";  
    }  
    description "Noise Sigma feasibility";  
  }  
  
  grouping chromatic-dispersion {  
    leaf chromatic-dispersion {  
      type decimal64 {  
        fraction-digits 2;  
      }  
      units "ps/nm";  
      description "Chromatic Dispersion (CD) related to the OMS";  
    }  
    description "Chromatic Dispersion (CD) feasibility";  
  }  
  
  grouping chromatic-dispersion-slope {  
    leaf chromatic-dispersion-slope {  
      type decimal64 {  
        fraction-digits 2;  
      }  
      units "ps/nm^2";  
    }  
  }
```

```
        description "Chromatic Dispersion (CD) Slope related to
            the OMS";
    }
    description "Chromatic Dispersion (CD) Slope feasibility";
}

grouping pmd {
    leaf pmd {
        type decimal64 {
            fraction-digits 2;
        }
        units "ps";
        description "Polarization Mode Dispersion (PMD) related
            to OMS";
    }
    description "Polarization Mode Dispersion (PMD) feasibility";
}

grouping pdl {
    leaf pdl {
        type decimal64 {
            fraction-digits 2;
        }
        units "dB";
        description "Polarization Dependent Loss (PDL) related to
            the OMS";
    }
    description "Polarization Dependent Loss (PDL) feasibility";
}

grouping drop-power {
    leaf drop-power {
        type decimal64 {
            fraction-digits 2;
        }
        units "dBm";
        description "Drop Power value at the DWDM Transceiver RX
            side";
    }
    description "Drop Power feasibility";
}

grouping drop-power-sigma {
    leaf drop-power-sigma {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
    }
}
```

```
        description "Drop Power Sigma value at the DWDM Transceiver
        RX side";
    }
    description "Drop Power Sigma feasibility";
}

grouping ripple {
    leaf ripple {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
        description "Channel Ripple";
    }
    description "Channel Ripple";
}

grouping ch-noise-figure {
    list ch-noise-figure {
        description "Channel signal-spontaneous noise figure";

        leaf input-to-output {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from input port to output port";
        }

        leaf input-to-drop {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from input port to drop port";
        }

        leaf add-to-output {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from add port to output port";
        }
    }
    description "Channel signal-spontaneous noise figure";
}
```

```
    grouping dgd {
    leaf dgd {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
        description "differential group delay";
    }
    description "differential group delay";
}

grouping ch-isolation {
list ch-isolation {
    description "adjacent and not adjacent channel isolation";

leaf ad-ch-isol {
    type decimal64 {
        fraction-digits 2;
    }
    units "dB";
    description "adjacent channel isolation";
}

leaf no-ad-ch-iso {
    type decimal64 {
        fraction-digits 2;
    }
    units "dB";
    description "non adjacent channel isolation";
}
}
description "djacent and not adjacent channel isolation";
}

grouping ch-extinction {
leaf cer {
    type decimal64 {
        fraction-digits 2;
    }
    units "db";
    description "channel extinction";
}
description "channel extinction";
}

grouping att-coefficient {
leaf att {
    type decimal64 {
```

```

        fraction-digits 2;
    }
    units "db";
    description "Attenuation coefficient (for a fibre segment)";
}
description "Attenuation coefficient (for a fibre segment)";
}

}

}

```

<CODE ENDS>

10. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operation and content.

11. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces:ietf-ext-xponder-wdm-if

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
prefix: ietf-ext-xponder-wdm-if reference: RFC XXXX
```

12. Acknowledgements

Marco Cardani.

13. Contributors

Dean Bogdanovic
Juniper Networks
Westford
U.S.A.
email deanb@juniper.net

Walid Wakim
Cisco
9501 Technology Blvd
ROSEMONT, ILLINOIS 60018
UNITED STATES
email wwakim@cisco.com

Marco Cardani
Cisco
vis S.Maria Molgora, 46b
20871 - Vimercate
Monza Brianza
Italy
email mcardani@cisco.com

Giovanni Martinelli
Cisco
vis S.Maria Molgora, 46b
20871 - Vimercate
Monza Brianza
Italy
email giomarti@cisco.com

14. References

14.1. Normative References

- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<http://www.rfc-editor.org/info/rfc2863>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<http://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<http://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<http://www.rfc-editor.org/info/rfc2580>>.
- [RFC3591] Lam, H-K., Stewart, M., and A. Huynh, "Definitions of Managed Objects for the Optical Interface Type", RFC 3591, DOI 10.17487/RFC3591, September 2003, <<http://www.rfc-editor.org/info/rfc3591>>.
- [RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <<http://www.rfc-editor.org/info/rfc6205>>.
- [ITU.G698.2]
International Telecommunications Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T Recommendation G.698.2, November 2009.
- [ITU.G709]
International Telecommunications Union, "Interface for the Optical Transport Network (OTN)", ITU-T Recommendation G.709, March 2003.
- [ITU.G872]
International Telecommunications Union, "Architecture of optical transport networks", ITU-T Recommendation G.872, November 2001.
- [ITU.G798]
International Telecommunications Union, "Characteristics of optical transport network hierarchy equipment functional blocks", ITU-T Recommendation G.798, October 2010.

- [ITU.G874]
International Telecommunications Union, "Management aspects of optical transport network elements", ITU-T Recommendation G.874, July 2010.
- [ITU.G874.1]
International Telecommunications Union, "Optical transport network (OTN): Protocol-neutral management information model for the network element view", ITU-T Recommendation G.874.1, January 2002.
- [ITU.G959.1]
International Telecommunications Union, "Optical transport network physical layer interfaces", ITU-T Recommendation G.959.1, November 2009.
- [ITU.G826]
International Telecommunications Union, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T Recommendation G.826, November 2009.
- [ITU.G8201]
International Telecommunications Union, "Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)", ITU-T Recommendation G.8201, April 2011.
- [ITU.G694.1]
International Telecommunications Union, "Spectral grids for WDM applications: DWDM frequency grid", ITU-T Recommendation G.694.1, June 2002.
- [ITU.G7710]
International Telecommunications Union, "Common equipment management function requirements", ITU-T Recommendation G.7710, May 2008.

14.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<http://www.rfc-editor.org/info/rfc3410>>.

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<http://www.rfc-editor.org/info/rfc2629>>.
- [RFC4181] Heard, C., Ed., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, DOI 10.17487/RFC4181, September 2005, <<http://www.rfc-editor.org/info/rfc4181>>.
- [I-D.ietf-ccamp-dwdm-if-mng-ctrl-fwk]
Kunze, R., Grammel, G., Beller, D., and G. Galimberti, "A framework for Management and Control of DWDM optical interface parameters", draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk-00 (work in progress), April 2016.
- [RFC4054] Strand, J., Ed. and A. Chiu, Ed., "Impairments and Other Constraints on Optical Layer Routing", RFC 4054, DOI 10.17487/RFC4054, May 2005, <<http://www.rfc-editor.org/info/rfc4054>>.

Appendix A. Change Log

This optional section should be removed before the internet draft is submitted to the IESG for publication as an RFC.

Note to RFC Editor: please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Gabriele Galimberti (editor)
Cisco
Via Santa Maria Molgora, 48 c
20871 - Vimercate
Italy

Phone: +390392091462
Email: ggalimbe@cisco.com

Ruediger Kunze
Deutsche Telekom
Dddd, xx
Berlin
Germany

Phone: +49xxxxxxxxxx
Email: RKunze@telekom.de

Dharini Hiremagalur (editor)
Juniper
1194 N Mathilda Avenue
Sunnyvale - 94089 California
USA

Email: dharinih@juniper.net

Gert Grammel (editor)
Juniper
Oskar-Schlemmer Str. 15
80807 Muenchen
Germany

Phone: +49 1725186386
Email: ggrammel@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 2, 2022

G.Galimberti, Ed.
Cisco
R. Kunze, Ed.
Deutsche Telekom
D. Hiremagalur, Ed.
G. Grammel, Ed.
Juniper
July 1, 2021

A YANG model to manage the optical parameters for in a WDM network
draft-galimbe-ccamp-iv-yang-12

Abstract

This memo defines a Yang model that translate the information model to support Impairment-Aware (IA) Routing and Wavelength Assignment (RWA) functionality. The information model is defined in draft-ietf-ccamp-wson-iv-info and draft-martinelli-ccamp-wson-iv-encode. This document defines proper encoding and extend to the models defined in draft-lee-ccamp-wson-yang to support Impairment-Aware (IA) Routing and Wavelength Assignment (RWA) functions

The Yang model defined in this memo can be used for Optical Parameters monitoring and/or configuration of the multivendor Endpoints and ROADMs. The use of this model does not guarantee interworking of transceivers over a DWDM. Optical path feasibility and interoperability has to be determined by means outside the scope of this document. The purpose of this model is to program interface parameters to consistently configure the mode of operation of transceivers.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 2, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. The Internet-Standard Management Framework	3
3. Conventions	4
4. Definition	4
5. Applicability	4
6. Properties	4
7. Overview	4
7.1. Optical Parameters Description	5
7.1.1. Optical path from point Ss to Rs	6
7.1.2. Rs and Ss Configuration	7
7.1.3. Table of Application Codes	7
7.2. Use Cases	7
7.3. Optical Parameters for impairment validation in a WDM network	7
8. Structure of the Yang Module	8
9. Yang Module	9
10. Security Considerations	20
11. IANA Considerations	20
12. Acknowledgements	21
13. Contributors	21
14. References	22
14.1. Normative References	22
14.2. Informative References	24
Appendix A. Change Log	24

Appendix B. Open Issues	24
Authors' Addresses	24

1. Introduction

This memo defines a Yang model that translates the existing mib module defined in draft-ietf-ccamp-wson-iv-info and draft-martinelli-ccamp-wson-iv-encode to provide the network impairment information to an SDN controller. One of the key SDN controller features is to support multivendor network and support the service calculation and deployment in multilayer topologies, for the DWDM layer it is fundamental that the SDN controller is aware of the optical impairments to verify the feasibility of new circuits before their provisioning. Although SDN controller will not apply exhaustive and accurate algorithms and the optical channel feasibility verification may have a degree of unreliability this function can work on a multivendor common set of parameter and algorithms to ensure the operator the best change to set a circuit. This document follows the same impairment definition and applicability of draft-ietf-ccamp-wson-iv-info.

The optical impairments related to the DWDM Transceiver are described by draft draft-dharini-ccamp-if-param-yang. Applications are defined in G.698.2 [ITU.G698.2] using optical interface parameters at the single-channel connection points between optical transmitters and the optical multiplexer, as well as between optical receivers and the optical demultiplexer in the DWDM system. This Recommendation uses a methodology which explicitly specify the details of the optical network between reference point Ss and Rs, e.g., the passive and active elements or details of the design.

The building of a yang model describing the optical parameters allows the different vendors and operator to retrieve, provision and exchange information across multi-vendor domains in a standardized way. In addition to the parameters specified in ITU recommendations the Yang models support also the "vendor specific parameters".

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

This memo specifies a Yang model for optical interfaces.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]. In the description of OIDs the convention: Set (S) Get (G) and Trap (T) conventions will describe the action allowed by the parameter.

4. Definition

For a detailed definition this draft refers to draft-ietf-ccamp-wson-iv-info.

5. Applicability

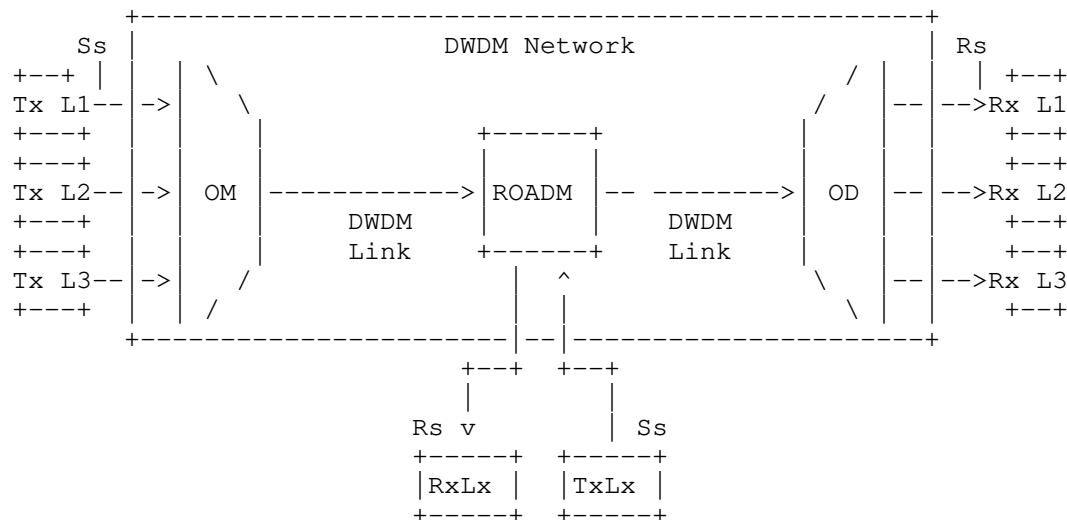
This document targets at Scenario C defined in [RFC6566] section 4.1.1. as approximate impairment estimation. The Approximate concept refers to the fact that this Information Model covers information mainly provided by [ITU.G680] Computational Model. Although the [RFC6566] provides no or little approximation the parameters described in this draft can be applied to the algorithms verifying the circuit feasibility in the new coherent non compensated DWDM networks. In this case the impairments verification can reach a good reliability and accuracy. This draft does not address computational matters but provides all the information suitable to cover most of the full coherent network algorithms, not being exhaustive the information can give an acceptable or even good approximation in terms of connection feasibility. This may not be true for legacy compensated networks.

6. Properties

For the signal properties this draft refers to draft-ietf-ccamp-wson-iv-info Ch.2.3 with some extension of the parameters.

7. Overview

Figure 1 shows a set of reference points, for single-channel connection between transmitters (Tx) and receivers (Rx). Here the DWDM network elements include an OM and an OD (which are used as a pair with the opposing element), one or more optical amplifiers and may also include one or more OADMs.



Ss = reference point at the DWDM network element tributary output
Rs = reference point at the DWDM network element tributary input
Lx = Lambda x
OM = Optical Mux
OD = Optical Demux
ROADM = Reconfigurable Optical Add Drop Mux

from Fig. 5.1/G.698.2

Figure 1: External transponder in WDM networks

7.1. Optical Parameters Description

The link between the external transponders through a WDM network media channels are managed at the edges, i.e. at the transmitters (Tx) and receivers (Rx) attached to the S and R reference points respectively. The set of parameters that could be managed are defined by the "application code" notation

The definitions of the optical parameters are provided below to increase the readability of the document, where the definition is

ended by (R) the parameter can be retrieve with a read, when (W) it can be provisioned by a write, (R,W) can be either read or written.

7.1.1. Optical path from point Ss to Rs

The following parameters for the optical path from point S and R are defined in G.698.2 [ITU.G698.2].

Maximum and minimum (residual) chromatic dispersion:

These parameters define the maximum and minimum value of the optical path "end to end chromatic dispersion" (in ps/nm) that the system shall be able to tolerate. (R)

Minimum optical return loss at Ss:

These parameter defines minimum optical return loss (in dB) of the cable plant at the source reference point (Ss), including any connectors (R)

Maximum discrete reflectance between Ss and Rs:

Optical reflectance is defined to be the ratio of the reflected optical power present at a point, to the optical power incident to that point. Control of reflections is discussed extensively in ITU-T Rec. G.957 (R)

Maximum differential group delay:

Differential group delay (DGD) is the time difference between the fractions of a pulse that are transmitted in the two principal states of polarization of an optical signal. For distances greater than several kilometers, and assuming random (strong) polarization mode coupling, DGD in a fiber can be statistically modelled as having a Maxwellian distribution. (R)

Maximum polarization dependent loss:

The polarization dependent loss (PDL) is the difference (in dB) between the maximum and minimum values of the channel insertion loss (or gain) of the black link from point SS to RS due to a variation of the state of polarization (SOP) over all SOPs. (R)

Maximum inter-channel crosstalk:

Inter-channel crosstalk is defined as the ratio of total power in all of the disturbing channels to that in the wanted channel, where the wanted and disturbing channels are at different wavelengths. The parameter specifies the isolation of a link conforming to the "black link" approach such that under the worst-case operating conditions the inter-channel crosstalk at any reference point RS is less than the maximum inter-channel crosstalk value (R)

Maximum interferometric crosstalk:

This parameter places a requirement on the isolation of a link conforming to the "black link" approach such that under the worst case operating conditions the interferometric crosstalk at any reference point RS is less than the maximum interferometric crosstalk value. (R)

Maximum optical path OSNR penalty:

The optical path OSNR penalty is defined as the difference between the Lowest OSNR at Rs and Lowest OSNR at Ss that meets the BER requirement (R)

Maximum ripple:

Although is defined in G.698.2 (R).

7.1.2. Rs and Ss Configuration

For the Rs and Ss configuration this draft refers the draft-dharini-ccamp-dwdm-if-param-yang while for the Rs-Ss extended parameters for coherent transmission interfaces refer to draft-dharini-ccamp-dwdm-if-param-yang

7.1.3. Table of Application Codes

For Application Codes configuration this draft refers the draft-dharini-ccamp-dwdm-if-param-yang

7.2. Use Cases

The use cases are described in draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

7.3. Optical Parameters for impairment validation in a WDM network

The ietf-opt-parameters-wdm is an augment to the ?????. It allows the user to get and set the application Optical Parameters of a DWDM network.

```

module: ietf-opt-parameters-wdm
  augment /if:interfaces/if:interface:
    +--rw optical-transport
    |   +--rw attenuator-value?      attenuator-t
    |   +--rw offset?                decimal64
    |   +--rw channel-power-ref?     decimal64
    |   +--rw tilt-calibration?      tilt-t
    +--rw opwr-threshold-warning
    |   +--rw opwr-min?              dbm-t
    |   +--rw opwr-min-clear?       dbm-t

```

```

|   +---rw opwr-max?          dbm-t
+---rw gain-degrade-alarm
|   +---rw gain-degrade-low?   dbm-t
|   +---rw gain-degrade-high?  dbm-t
+---rw power-degrade-high-alarm
|   +---rw gain-degrade-high?   dbm-t
+---rw power-degrade-low-alarm
|   +---ro power-degrade-low?   dbm-t
+---rw noise
|   +---rw noise?    decimal64
+---rw
|   +---rw noise?    decimal64
+---rw chromatic-dispersion
|   +---rw noise-sigma?    decimal64
+---rw chromatic-dispersion-slope
|   +---rw chromatic-dispersion-slope?    decimal64
+---rw pmd
|   +---rw pmd?    decimal64
+---rw pdl
|   +---rw pdl?    decimal64
+---rw drop-power
|   +---rw drop-power?    decimal64
+---rw drop-power-sigma
|   +---rw noise?    decimal64
+---rw ripple
|   +---rw drop-power-sigma?    decimal64
+---ro ch-noise-figure
|   +---ro ch-noise-figure* [ch-noise-fig]
|       +---ro ch-noise-fig          ch-noise-figure-point
|       +---ro input-to-output?    decimal64
|       +---ro input-to-drop?      decimal64
|       +---ro add-to-output?      decimal64
+---rw dgd
|   +---rw dgd?    decimal64
+---ro ch-isolation
|   +---ro ch-isolation* [ch-isolat]
|       +---ro ch-isolat          ch-isolation-cross
|       +---ro ad-ch-isol?      decimal64
|       +---ro no-ad-ch-iso?    decimal64
+---rw ch-extinction
|   +---rw cer?    decimal64

```

8. Structure of the Yang Module

ietf-opt-parameters-wdm is a top level model for the support of this feature.

9. Yang Module

The ietf-opt-parameters-wdm is defined as an extension to ietf interfaces ????.

<CODE BEGINS> file "ietf-opt-parameters-wdm.yang"

```
module ietf-opt-parameters-wdm {  
  namespace "urn:ietf:params:xml:ns:yang:ietf-opt-parameters-wdm";  
  prefix iietf-opt-parameters-wdm;
```

```
    import ietf-interfaces {  
      prefix if;  
    }
```

```
    import iana-if-type {  
      prefix ianaift;  
    }
```

```
    organization  
      "IETF CCAMP  
      Working Group";
```

```
    contact  
      "WG Web:   <http://tools.ietf.org/wg/ccamp/>  
      WG List:  <mailto:ccamp@ietf.org>
```

```
      Editor:   Gabriele Galimberti  
               <mailto:ggalimbe@cisco.com>;
```

```
    description  
      "This module contains a collection of YANG definitions for  
      collecting and configuring Optical Parameters  
      in Optical Networks and calculate the circuit feasibility.
```

```
      Copyright (c) 2016 IETF Trust and the persons identified  
      as authors of the code. All rights reserved.
```

```
      Redistribution and use in source and binary forms, with or  
      without modification, is permitted pursuant to, and  
      subject to the license terms contained in, the Simplified  
      BSD License set forth in Section 4.c of the IETF Trust's  
      Legal Provisions Relating to IETF Documents
```

```
(http://trustee.ietf.org/license-info).";
```

```
revision "2021-06-28" {  
    description  
        "Revision 1.3";  
    reference  
        "";  
}
```

```
revision "2020-03-13" {  
    description  
        "Revision 1.2";  
    reference  
        "";  
}
```

```
revision "2018-10-22" {  
    description  
        "Revision 1.1";  
    reference  
        "";  
}
```

```
revision "2018-03-06" {  
    description  
        "Revision 1.0";  
    reference  
        "";  
}
```

```
revision "2016-10-30" {  
    description  
        "Initial revision.";  
    reference  
        "RFC XXXX: A YANG Data Model for Optical Parameters  
        of DWDM Networks  
        ";  
}
```

```
typedef tilt-t {  
    type decimal64 {  
        fraction-digits 2;  
        range "-5..5";  
    }  
    description "Tilt Type";  
}
```

```
}

typedef signal-output-power-t {
    type decimal64 {
        fraction-digits 2;
        range "-10..30";
    }
    description "
        Amplifier Power provisioning ";
}

typedef active-channel-t {
    type union {
        type uint8 {
            range "0..200";
        }
    }
    description "
        Number of channels active on a span - and on an amplifier";
}

typedef dbm-t {
    type decimal64 {
        fraction-digits 2;
        range "-50..-30 | -10..5 | 100000000";
    }
    description "
        Amplifier Power in dBm ";
}

typedef attenuator-t {
    type decimal64 {
        fraction-digits 2;
        range "-15..-5";
    }
    description "
        Attenuation value (attenuator) applied after the Amplifier";
}

typedef ch-noise-figure-point {
    type decimal64 {
        fraction-digits 2;
        range "-15..-5";
    }
    description "
        Amplifier noise figure of point power";
}
```

```
typedef ch-isolation-cross {
  type decimal64 {
    fraction-digits 2;
    range "-15..-5";
  }
  description "
cross channel isolation value";
}

grouping opwr-threshold-warning-grp {
  description "
    Minimum Optical Power threshold
    - this is used to rise Power alarm ";

  leaf opwr-min {
    type dbm-t;
    units "dBm";
    default -1;
    description "Minimum Power Value";
  }

  leaf opwr-min-clear {
    type dbm-t;
    units "dBm";
    default -1;
    description "threshold to clear Minimum Power value Alarm";
  }

  leaf opwr-max {
    type dbm-t;
    units "dBm";
    default 1;
    description "
      Maximum Optical Power threshold
      - this is used to rise Power alarm ";
  }
}

grouping gain-degrade-alarm-grp {
  description "
    Low Optical Power gain threshold
    - this is used to rise Power alarm ";

  leaf gain-degrade-low {
    type dbm-t;
    units "dBm";
    default -1;
    description "Low Gain Degrade Value";
  }
}
```

```
    }

    leaf gain-degrade-high {
      type dbm-t;
      units "dBm";
      default 1;
      description "
        High Optical Power gain threshold
        - this is used to rise Power alarm ";
    }
  }

  grouping power-degrade-high-alarm-grp {
    description "
      High Optical Power gain alarm ";

    leaf gain-degrade-high {
      type dbm-t;
      units "dBm";
      default 1;
      description "Low Gain Degrade Value";
    }
  }

  grouping power-degrade-low-alarm-grp {
    description "
      Low Optical Power gain alarm ";

    leaf power-degrade-low {
      type dbm-t;
      units "dBm";
      default -1;
      config false;
      description "High Gain Degrade Value";
    }
  }

  grouping noise-grp {
    description "Noise feasibility";
    leaf noise {
      type decimal64 {
        fraction-digits 2;
      }
      units "dB";
      description "Noise feasibility - reference ITU-T G.680
        OSNR added to the signal by the OMS. The noise is intended
        per channel and is independent of the number of active
```

```
        channels in OMS";
    }
}

grouping noise-sigma-grp {
    description "Noise sigma feasibility";
    leaf noise-sigma {
        type decimal64 {
            fraction-digits 2;
        }
        units "dB";
        description "Noise Sigma feasibility - accuracy of the
            OSNR added to
            the signal by the OMS";
    }
}

grouping chromatic-dispersion-grp {
    description "Chromatic Dispersion";
    leaf chromatic-dispersion {
        type decimal64 {
            fraction-digits 2;
        }
        units "ps/nm";
        description "Chromatic Dispersion (CD) related to the OMS";
    }
}

grouping chromatic-dispersion-slope-grp {
    description "Chromatic Dispersion slope";
    leaf chromatic-dispersion-slope {
        type decimal64 {
            fraction-digits 2;
        }
        units "ps/nm^2";
        description "Chromatic Dispersion (CD) Slope related to
            the OMS";
    }
}

grouping pmd-grp {
    description "Polarization Mode Dispersion";
    leaf pmd {
        type decimal64 {
            fraction-digits 2;
        }
        units "ps";
    }
}
```



```
        description "Polarization Mode Dispersion (PMD) related
            to OMS";
    }
}

grouping pdl-grp {
    description "Polarization Dependent Loss";
    leaf pdl {
        type decimal64 {
            fraction-digits 2;
        }
        units "dB";
        description "Polarization Dependent Loss (PDL) related to
            the OMS";
    }
}

grouping drop-power-grp {
    description "Drop power at DWDM if RX feasibility";
    leaf drop-power {
        type decimal64 {
            fraction-digits 2;
        }
        units "dBm";
        description "Drop Power value at the DWDM Transceiver RX
            side";
    }
}

grouping drop-power-sigma-grp {
    description "Drop power sigma at DWDM if RX feasibility ";
    leaf drop-power-sigma {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
        description "Drop Power Sigma value at the DWDM Transceiver
            RX side";
    }
}

grouping ripple-grp {
    description "Channel Ripple";
    leaf ripple {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
    }
}
```

```
        description "Channel Ripple";
    }
}

grouping ch-noise-figure-grp {
    list ch-noise-figure {
        key "ch-noise-fig";
        description "Channel signal-spontaneous noise figure";

        leaf ch-noise-fig {
            type ch-noise-figure-point;
            description "Channel signal-spontaneous noise
            figure point";
        }

        leaf input-to-output {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from input port to output port";
        }

        leaf input-to-drop {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from input port to drop port";
        }

        leaf add-to-output {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "from add port to output port";
        }
    }
    description "Channel signal-spontaneous noise figure";
}

grouping dgd-grp {
    description "Differential Group Delay";
    leaf dgd {
        type decimal64 {
            fraction-digits 2;
        }
    }
}
```

```
        units "db";
        description "differential group delay";
    }
}

grouping ch-isolation-grp {
    list ch-isolation {
        key "ch-isolat";
        description "adjacent and not adjacent channel isolation";

        leaf ch-isolat {
            type ch-isolation-cross;
            description "channel isolation from adjacent";
        }

        leaf ad-ch-isol {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "adjacent channel isolation";
        }

        leaf no-ad-ch-iso {
            type decimal64 {
                fraction-digits 2;
            }
            units "dB";
            description "non adjacent channel isolation";
        }
    }
    description "djacent and not adjacent channel isolation";
}

grouping ch-extinction-grp {
    description "Channel Extinsion";
    leaf cer {
        type decimal64 {
            fraction-digits 2;
        }
        units "db";
        description "channel extinction";
    }
}

grouping att-coefficient-grp {
    description "Attenuation coefficient (for a fibre segment)";
    leaf att {
```

```
    type decimal64 {
      fraction-digits 2;
    }
    units "db";
    description "Attenuation coefficient (for a fibre segment)";
  }
}

augment "/if:interfaces/if:interface" {
  when "if:type = 'ianaift:opticalTransport'" {
    description "Specific optical-transport Interface Data";
  }
  description "Specific optical-transport Interface Data";
  container optical-transport {
    description "Specific optical-transport Data";

    leaf attenuator-value {
      type attenuator-t;
      description "External attenuator value ";
    }

    leaf offset {
      type decimal64 {
        fraction-digits 2;
        range "-30..30";
      }
      description "Raman and power amplifiers offset";
    }

    leaf channel-power-ref {
      type decimal64 {
        fraction-digits 2;
        range "-10..15";
      }
      description "Optical power per channel";
    }

    leaf tilt-calibration {
      type tilt-t;
      description "Amplifier Tilt tuning";
    }
  }
  container opwr-threshold-warning {
    description "Optical power threshold warning";
    uses opwr-threshold-warning-grp;
  }
  container gain-degrade-alarm {
```

```
        description "Gain degrade alarm";
        uses gain-degrade-alarm-grp;
    }
    container power-degrade-high-alarm {
        description "Power degrade high aparm";
        uses power-degrade-high-alarm-grp;
    }
    container power-degrade-low-alarm {
        description "Power degrade low alarm";
        uses power-degrade-low-alarm-grp;
    }
    container noise {
        description "Channel Noise feasibility";
        uses noise-grp;
    }
    container noise-sigma {
        description "Channel Noise sigma feasibility";
        uses noise-grp;
    }
    container chromatic-dispersion {
        description "Chromatic Dispersion";
        uses noise-sigma-grp;
    }
    container chromatic-dispersion-slope {
        description "Chromatic Dispersion slope";
        uses chromatic-dispersion-slope-grp;
    }
    container pmd {
        description "Polarization Mode Dispersion";
        uses pmd-grp;
    }
    container pdl {
        description "Polarization Dependent Loss";
        uses pdl-grp;
    }
    container drop-power {
        description "Drop power at DWDM if RX feasibility";
        uses drop-power-grp;
    }
    container drop-power-sigma {
        description "Drop power sigma at DWDM if RX feasibility";
        uses noise-grp;
    }
    container ripple {
        description "Channel Ripple";
        uses drop-power-sigma-grp;
    }
    container ch-noise-figure {
```

```
        config false;
        description "Channel signal-spontaneous noise figure";
        uses ch-noise-figure-grp;
    }
    container dgd {
        description "Differential Group Delay";
        uses dgd-grp;
    }
    container ch-isolation {
        config false;
        description "adjacent and not adjacent channel isolation";
        uses ch-isolation-grp;
    }
    container ch-extinction {
        description "Channel Extinction";
        uses ch-extinction-grp;
    }
}
}
}
```

<CODE ENDS>

10. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operation and content.

11. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces:ietf-ext-xponder-wdm-if

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

This document registers a YANG module in the YANG Module Names registry [RFC6020].

prefix: ietf-ext-xponder-wdm-if reference: RFC XXXX

12. Acknowledgements

Marco Cardani.

13. Contributors

Dean Bogdanovic

Westford
U.S.A.
email

Walid Wakim

Cisco
9501 Technology Blvd
ROSEMONT, ILLINOIS 60018
UNITED STATES
email wwakim@cisco.com

Marco Cardani

Cisco
vis S.Maria Molgora, 48c
20871 - Vimercate
Monza Brianza
Italy
email mcardani@cisco.com

Giovanni Martinelli

Cisco
vis S.Maria Molgora, 48c
20871 - Vimercate
Monza Brianza
Italy
email giomarti@cisco.com

14. References

14.1. Normative References

- [ITU.G694.1]
International Telecommunications Union, "Spectral grids for WDM applications: DWDM frequency grid", ITU-T Recommendation G.694.1, February 2012.
- [ITU.G698.2]
International Telecommunications Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T Recommendation G.698.2, November 2009.
- [ITU.G709]
International Telecommunications Union, "Interface for the Optical Transport Network (OTN)", ITU-T Recommendation G.709, June 2016.
- [ITU.G7710]
International Telecommunications Union, "Common equipment management function requirements", ITU-T Recommendation G.7710, August 2017.
- [ITU.G798]
International Telecommunications Union, "Characteristics of optical transport network hierarchy equipment functional blocks", ITU-T Recommendation G.798, December 2017.
- [ITU.G8201]
International Telecommunications Union, "Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)", ITU-T Recommendation G.8201, April 2011.
- [ITU.G826]
International Telecommunications Union, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T Recommendation G.826, December 2002.
- [ITU.G872]
International Telecommunications Union, "Architecture of optical transport networks", ITU-T Recommendation G.872, January 2017.

- [ITU.G874] International Telecommunications Union, "Management aspects of optical transport network elements", ITU-T Recommendation G.874, August 2017.
- [ITU.G874.1] International Telecommunications Union, "Optical transport network (OTN): Protocol-neutral management information model for the network element view", ITU-T Recommendation G.874.1, November 2016.
- [ITU.G959.1] International Telecommunications Union, "Optical transport network physical layer interfaces", ITU-T Recommendation G.959.1, July 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<https://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<https://www.rfc-editor.org/info/rfc2580>>.
- [RFC2863] McCloghrie, K. and F. Kastenholtz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.
- [RFC3591] Lam, H-K., Stewart, M., and A. Huynh, "Definitions of Managed Objects for the Optical Interface Type", RFC 3591, DOI 10.17487/RFC3591, September 2003, <<https://www.rfc-editor.org/info/rfc3591>>.

- [RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <<https://www.rfc-editor.org/info/rfc6205>>.

14.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<https://www.rfc-editor.org/info/rfc2629>>.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<https://www.rfc-editor.org/info/rfc3410>>.
- [RFC4054] Strand, J., Ed. and A. Chiu, Ed., "Impairments and Other Constraints on Optical Layer Routing", RFC 4054, DOI 10.17487/RFC4054, May 2005, <<https://www.rfc-editor.org/info/rfc4054>>.
- [RFC4181] Heard, C., Ed., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, DOI 10.17487/RFC4181, September 2005, <<https://www.rfc-editor.org/info/rfc4181>>.

Appendix A. Change Log

This optional section should be removed before the internet draft is submitted to the IESG for publication as an RFC.

Note to RFC Editor: please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Gabriele Galimberti (editor)
Cisco
Via Santa Maria Molgora, 48 c
20871 - Vimercate
Italy

Phone: +390392091462
Email: ggalimbe@cisco.com

Ruediger Kunze (editor)
Deutsche Telekom
Winterfeldtstr. 21-27
10781 Berlin
Germany

Phone: +491702275321
Email: RKunze@telekom.de

Dharini Hiremagalur (editor)
Juniper
1194 N Mathilda Avenue
Sunnyvale - 94089 California
USA

Email: dharinih@juniper.net

Gert Grammel (editor)
Juniper
Oskar-Schlemmer Str. 15
80807 Muenchen
Germany

Phone: +49 1725186386
Email: ggrammel@juniper.net

CCAMP Working Group

Internet Draft

Intended status: Standard Track

Y. Lee (Editor)

D. Dhody

X. Zhang

A. Guo

Huawei

V. Lopez

Telefonica

D. King

U. of Lancaster

B. Yoon

ETRI

R.Vilalta

CTTC

Expires: August 21, 2017

February 21, 2017

A Yang Data Model for WSON Optical Networks

draft-ietf-ccamp-wson-yang-05.txt

Abstract

This document provides a YANG data model for the routing and wavelength assignment (RWA) TE topology in wavelength switched optical networks (WSONs).

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on August 21, 2017.

Copyright Notice

Lee, et al.

Expires August 2017

[Page 1]

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	2
2. YANG Model (Tree Structure).....	3
3. WSON-RWA YANG Model.....	4
4. Security Considerations.....	11
5. IANA Considerations.....	11
6. Acknowledgments.....	11
7. References.....	12
7.1. Normative References.....	12
7.2. Informative References.....	12
8. Contributors.....	12
Authors' Addresses.....	12

1. Introduction

This document provides a YANG data model for the routing and wavelength assignment (RWA) Traffic Engineering (TE) topology in wavelength switched optical networks (WSONs). The YANG model described in this document is a WSON technology-specific Yang model based on the information model developed in [RFC7446] and the two encoding documents [RFC7581] and [RFC7579] that developed protocol independent encodings based on [RFC7446]. This document augments the the generic TE topology draft [TE-TOPO].

What is not in scope of this document is both impairment-aware WSON and flex-grid.

2. YANG Model (Tree Structure)

```

module: ietf-wson-topology
  augment /nd:networks/nd:network/nd:network-types:
    +--rw wson-topology!
  augment /nd:networks/nd:network/nd:node/tet:te/tet:config/tet:te-node-
attributes/tet:connectivity-matrix:
    +--rw matrix-interface* [in-port-id]
      +--rw in-port-id      wson-interface-ref
      +--rw out-port-id?    wson-interface-ref
  augment /nd:networks/nd:network/nd:node/tet:te/tet:state/tet:te-node-
attributes/tet:connectivity-matrix:
    +--ro matrix-interface* [in-port-id]
      +--ro in-port-id      wson-interface-ref
      +--ro out-port-id?    wson-interface-ref
  augment /nd:networks/nd:network/lnk:link/tet:te/tet:config/tet:te-link-attri-
butes:
    +--rw channel-max?          int32
    +--rw default-frequency?    decimal64
    +--rw channel-spacing?      decimal64
    +--rw wavelength-available-bitmap* binary
  augment /nd:networks/nd:network/lnk:link/tet:te/tet:state/tet:te-link-attri-
butes:
    +--ro channel-max?          int32
    +--ro default-frequency?    decimal64
    +--ro channel-spacing?      decimal64
    +--ro wavelength-available-bitmap* binary
  augment /nd:networks/nd:network/nd:node/tet:te/tet:config/tet:te-node-attri-
butes:
    +--rw wson-node
      | +--rw device-type?    devicetype
      | +--rw dir?            directionality
      | +--rw interfaces* [name]
      |   +--rw name          string
      |   +--rw port-number?  uint32
      |   +--rw input-port?   boolean
      |   +--rw output-port?  boolean
      |   +--rw description?  string
    +--rw resource-pool* [resource-pool-id]
      +--rw resource-pool-id  uint32
      +--rw pool-state?       boolean
      +--rw matrix-interface* [in-port-id]
        +--rw in-port-id      wson-interface-ref
        +--rw out-port-id?    wson-interface-ref
  augment /nd:networks/nd:network/nd:node/tet:te/tet:state/tet:te-node-attribute
s:
    +--ro wson-node
      | +--ro device-type?    devicetype
      | +--ro dir?            directionality
      | +--ro interfaces* [name]
      |   +--ro name          string

```

```
|      +--ro port-number?    uint32
|      +--ro input-port?     boolean
|      +--ro output-port?    boolean
|      +--ro description?    string
+--ro resource-pool* [resource-pool-id]
    +--ro resource-pool-id    uint32
    +--ro pool-state?        boolean
    +--ro matrix-interface* [in-port-id]
        +--ro in-port-id     wson-interface-ref
        +--ro out-port-id?   wson-interface-ref
```

3. WSON-RWA YANG Model

```
<CODE BEGINS> file "ietf-wson-topology@2017-02-21.yang"

module ietf-wson-topology {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-wson-topology";

  prefix "wson";

  import ietf-network {
    prefix "nd";
  }

  import ietf-network-topology {
    prefix "lnk";
  }

  import ietf-inet-types {
    prefix "inet";
  }
}
```

```
import ietf-te-topology {
prefix "tet";
}

organization
  "IETF CCAMP Working Group";

contact
  "Editor:   Young Lee   <leeyoung@huawei.com>";

description
  "This module contains a collection of YANG definitions for
  RWA WSON.

  Copyright (c) 2016 IETF Trust and the persons identified as
  authors of the code.  All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD
  License set forth in Section 4.c of the IETF Trust's Legal
  Provisions Relating to IETF Documents
  (http://trustee.ietf.org/license-info).";

revision 2017-02-21 {
  description
    "version 5.";

  reference
    "RFC XXX: A Yang Data Model for WSON Optical Networks ";
}

typedef wson-topology-id {
  type inet:uri;
  description
    "The WSON Topology ID";
}

typedef wson-node-id {
  type inet:ip-address;
  description
    "The WSON Node ID";
}

typedef devicetype {
  type enumeration {
    enum adm {
      value 1;
    }
  }
}
```



```
        description
            "Device is ADM";
    }

    enum roadm {
        value 2;
        description
            "Device is ROAMD/OXC";
    }
}
description
    "device type: fixed (ADM) or switched (ROADM/OXC)";
}

typedef directionality {
    type enumeration {
        enum bidir {
            value 0;
            description
                "bi-directional";
        }
        enum input {
            value 1;
            description
                "input direction";
        }
        enum output {
            value 2;
            description
                "output direction";
        }
    }
}
description
    "The directionality of link set";
}

typedef wson-interface-ref {
    type leafref {
        path "/nd:networks/nd:network/nd:node/tet:te/tet:config"
        + "/tet:te-node-attributes/wson:wson-node/"
        + "wson:interfaces/wson:name";
    }
}
description
    "This type is used by data models that need to
    reference WSON interface.";
}
```

```
grouping wson-topology-type {
  description "wson-topology type";
  container wson-topology {
    presence "indicates a topology of wson";
    description
      "Container to identify wson topology type";
  }
}

grouping wson-node-attributes {
  description "wson node attributes";
  container wson-node {
    description "WSOON node attrttributes.";
    leaf device-type {
      type devicetype;
      description
        "device type: fixed (ADM) or switched
        (ROADM/OXC)";
    }
    leaf dir {
      type directionality;
      description
        "bi-directionality or input or output
        of link set";
    }
    list interfaces {
      key "name";
      unique "port-number"; // TODO Puerto y TP ID
      description "List of interfaces contained in the node";
      uses node-interface;
    }
  }
}

grouping node-interface {
  description "node interface definition";
  leaf name {
    type string;
    description "Interface name";
  }
  leaf port-number {
    type uint32;
    description "Number of the port used by the interface";
  }
  leaf input-port {
    type boolean;
    description "Determines if the port is an input port";
  }
  leaf output-port {
```

```

        type boolean;
        description
            "Determines if the port is an output port";
    }
    leaf description {
        type string;
        description "Description of the interface";
    }
}

grouping available-wavelength {
    description "describe available wavelengths";
    leaf-list wavelength-available-bitmap {
        type binary;
        description
            "array of bits (i.e., bitmap) that indicates
            if a wavelength is available or not on each
            channel.";
    }
}

grouping wson-link-attributes {
    description "Set of WSON link attributes";
    leaf channel-max {
        type int32;
        description "Maximum Number of OCh channels available
by the node";
    }
    leaf default-frequency {
        type decimal64 {
            fraction-digits 5;
        }
        units THz;
        default 193.1;
        description "Default Central Frequency";
    }
    leaf channel-spacing {
        type decimal64 {
            fraction-digits 5;
        }
        units GHz;
        description "This is fixed channel spacing for WSON,
e.g, 12.5, 25, 50, 100, ..";
    }
}

grouping wson-connectivity-matrix {
    description "wson connectivity matrix";
    list matrix-interface {

```

```
        key "in-port-id";

        description
            "matrix-interface describes input-ports
            and out-ports around a connectivity
            matrix";

        leaf in-port-id {
            type wson-interface-ref;
            description
                "The reference to in-port";
        }

        leaf out-port-id {
            type wson-interface-ref;
            description
                "The reference to out-port";
        }
    }
}

grouping resource-pool-attributes {
    description "resource pool describes regeneration or wave converter"
;
    list resource-pool {
        key "resource-pool-id";
        description
            "The resource pool list";

        leaf resource-pool-id {
            type uint32;
            description
                "The resource pool ID";
        }

        leaf pool-state {
            type boolean;
            description
                "TRUE is state UP; FALSE is state down";
        }
    }

    uses wson-connectivity-matrix;
}

augment "/nd:networks/nd:network/nd:network-types" {
    description "wson-topology augmented";
    uses wson-topology-type;
}
```

```
    augment "/nd:networks/nd:network/nd:node/tet:te/tet:config"
      + "/tet:te-node-attributes/tet:connectivity-matrix" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON connectivity
matrix.";
        }
        description "WSON connectivity matrix config augmentation";
        uses wson-connectivity-matrix;
      }

    augment "/nd:networks/nd:network/nd:node/tet:te/tet:state"
      + "/tet:te-node-attributes/tet:connectivity-matrix" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON connectivity
matrix.";
        }
        description "WSON connectivity matrix state augmentation";
        uses wson-connectivity-matrix;
      }

    augment "/nd:networks/nd:network/lnk:link/tet:te/tet:config"
      + "/tet:te-link-attributes" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON.";
        }
        description "WSON Link augmentation.";

        uses wson-link-attributes;
        uses available-wavelength;
      }

    augment "/nd:networks/nd:network/lnk:link/tet:te/tet:state"
      + "/tet:te-link-attributes" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON.";
        }
        description "WSON Link augmentation.";

        uses wson-link-attributes;
        uses available-wavelength;
```

```
    }

    augment "/nd:networks/nd:network/nd:node/tet:te/tet:config"
      + "/tet:te-node-attributes" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON.";
        }
        description "WSON Node augmentation.";

        uses wson-node-attributes;
        uses resource-pool-attributes;
      }

    augment "/nd:networks/nd:network/nd:node/tet:te/tet:state"
      + "/tet:te-node-attributes" {
        when "/nd:networks/nd:network/nd:network-types"
          + "/wson-topology" {
          description
            "This augment is only valid for WSON.";
        }
        description "WSON Node augmentation.";

        uses wson-node-attributes;
        uses resource-pool-attributes;
      }
  }
}

<CODE ENDS>
```

4. Security Considerations

TDB

5. IANA Considerations

TDB

6. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

7. References

7.1. Normative References

[TE-TOPO] X. Liu, et al., "YANG Data Model for TE Topologies", work in progress: draft-ietf-teas-yang-te-topo.

7.2. Informative References

[RFC7446] Y. Lee, G. Bernstein, D. Li, W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.

[RFC7579] G. Bernstein, Y. Lee, D. Li, W. Imajuku, "General Network Element Constraint Encoding for GMPLS Controlled Networks", RFC 7579, June 2015.

[RFC7581] G. Bernstein, Y. Lee, D. Li, W. Imajuku, "Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", RFC 7581, June 2015.

8. Contributors

Authors' Addresses

Young Lee (ed.)
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023
USA

Phone: (469) 277-5838
Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies India Pvt. Ltd,
Near EPIP Industrial Area, Kundalahalli Village, Whitefield,
Bangalore - 560 037 [H1-2A-245]

Email: dhruv.dhody@huawei.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com

Aihua Guo
Huawei Technologies
Email: aihuaguo@huawei.com

Victor Lopez
Telefonica
Email: victor.lopezalvarez@telefonica.com

Daniel King
University of Lancaster
Email: d.king@lancaster.ac.uk

Bin Yeong Yoon
ETRI
218 Gaijeongro, Yuseong-gu
Daejeon, Korea
Email: byyun@etri.re.kr

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

CCAMP Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 2, 2021

H. Zheng
Huawei Technologies
Y. Lee
Samsung
A. Guo
Futurewei
V. Lopez
Telefonica
D. King
University of Lancaster
December 29, 2020

A YANG Data Model for WSON (Wavelength Switched Optical Networks)
draft-ietf-ccamp-wson-yang-28

Abstract

This document provides a YANG data model for the routing and wavelength assignment (RWA) TE topology in wavelength switched optical networks (WSONs). The YANG data model defined in this document conforms to the Network Management Datastore Architecture (NMDA).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 2, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology and Notations	3
1.2. Tree Diagram	3
1.3. Prefix in Data Node Names	4
2. YANG Model (Tree Structure) for WSON topology	4
3. The YANG Code for WSON topology	24
4. Security Considerations	58
5. IANA Considerations	59
6. Contributors	59
7. References	60
7.1. Normative References	60
7.2. Informative References	61
Authors' Addresses	62

1. Introduction

This document provides a YANG data model for the routing and wavelength assignment (RWA) Traffic Engineering (TE) topology in transparent wavelength switched optical networks (WSONs). The YANG model described in this document is a WSON technology-specific YANG model based on the information model developed in [RFC7446] and the two encoding documents [RFC7581] and [RFC7579] that developed protocol independent encodings based on [RFC7446].

[ITU-Tg6982] defines amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces. The YANG data model defined in this document refers to the standard application mode defined in [ITU-Tg6982].

What is not in scope of this document is both impairment-aware optical networks and flexi-grid. Refer to [I-D.ietf-ccamp-optical-impairment-topology-yang] for impairment-aware optical network topology model and [I-D.ietf-ccamp-flexigrid-yang] for flexi-grid optical network topology model.

Additionally, transponders and resource blocks (e.g., 3R Regeneration) models are not in the scope of this document.

This document defines one YANG model: ietf-wson-topology (Section 3). This document augments the generic TE topology draft [RFC8795].

There are multiple applications for the yang data model defined in this document. For example, nodes within the network can use the data model to capture their understanding of the overall WSON topology and expose it to a controller. A controller can further propagate the topology to other controllers. The YANG model is used by NETCONF [RFC6020], [RFC8341] or a RESTCONF [RFC8040] protocol. The YANG data model defined in this document conforms to the Network Management Datastore Architecture [RFC8342].

1.1. Terminology and Notations

Refer to [RFC7446] and [RFC7581] for the key terms used in this document. The following terms are defined in [RFC7950] and are not redefined here:

- o client
- o server
- o augment
- o data model
- o data node

The following terms are defined in [RFC6241] and are not redefined here:

- o configuration data
- o state data

The terminology for describing YANG data models is found in [RFC7950].

1.2. Tree Diagram

A simplified graphical representation of the data model is used in Section 2 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. Prefix in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in the following table.

Prefix	YANG module	Reference
10-types	ietf-layer0-types	[ietf-ccamp-layer0-types]
wson	ietf-wson-topology	[RFCXXXX]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
tet	ietf-te-topology	[RFC8795]

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. YANG Model (Tree Structure) for WSON topology

```

module: ietf-wson-topology
  augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
    +--rw wson-topology!
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes:
      +--rw wson-node!
      +--rw is-reconfigurable-node?  boolean
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:label-restrictions/tet:label-restriction:
      +--rw grid-type?  identityref
      +--rw priority?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:from/tet:label-restrictions
    /tet:label-restriction:
      +--rw grid-type?  identityref
      +--rw priority?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:to/tet:label-restrictions
    /tet:label-restriction:
      +--rw grid-type?  identityref

```

```

    +---rw priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:label-restrictions/tet:label-restriction:
    +---ro grid-type?    identityref
    +---ro priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:from/tet:label-restrictions
    /tet:label-restriction:
    +---ro grid-type?    identityref
    +---ro priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:to/tet:label-restrictions
    /tet:label-restriction:
    +---ro grid-type?    identityref
    +---ro priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities/tet:label-restrictions
    /tet:label-restriction:
    +---rw grid-type?    identityref
    +---rw priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities
    /tet:local-link-connectivity/tet:label-restrictions
    /tet:label-restriction:
    +---rw grid-type?    identityref
    +---rw priority?      uint8
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:label-restrictions
    /tet:label-restriction:
    +---rw grid-type?    identityref
    +---rw priority?      uint8
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:information-source-entry/tet:label-restrictions
    /tet:label-restriction:
    +---ro grid-type?    identityref
    +---ro priority?      uint8
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:label-restrictions
    /tet:label-restriction:
    +---rw grid-type?    identityref
    +---rw priority?      uint8
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices

```

```

        /tet:label-restrictions/tet:label-restriction
        /tet:label-start/tet:te-label/tet:technology:
+---: (wson)
+---rw (grid-type)?
+---: (dwdm)
|   +---rw dwdm-n?    10-types:dwdm-n
+---: (cwdm)
+---rw cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
        /tet:te-node-attributes/tet:connectivity-matrices
        /tet:label-restrictions/tet:label-restriction
        /tet:label-end/tet:te-label/tet:technology:
+---: (wson)
+---rw (grid-type)?
+---: (dwdm)
|   +---rw dwdm-n?    10-types:dwdm-n
+---: (cwdm)
+---rw cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
        /tet:te-node-attributes/tet:connectivity-matrices
        /tet:label-restrictions/tet:label-restriction
        /tet:label-step/tet:technology:
+---: (wson)
+---rw (l0-grid-type)?
+---: (dwdm)
|   +---rw wson-dwdm-channel-spacing?    identityref
+---: (cwdm)
+---rw wson-cwdm-channel-spacing?    identityref
augment /nw:networks/nw:network/nw:node/tet:te
        /tet:te-node-attributes/tet:connectivity-matrices
        /tet:underlay/tet:primary-path/tet:path-element/tet:type
        /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---: (wson)
+---rw (grid-type)?
+---: (dwdm)
|   +---rw (single-or-super-channel)?
|       +---: (single)
|       |   +---rw dwdm-n?                    10-types:dwdm-n
|       +---: (super)
|       +---rw subcarrier-dwdm-n*    10-types:dwdm-n
+---: (cwdm)
+---rw cwdm-n?                    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
        /tet:te-node-attributes/tet:connectivity-matrices
        /tet:underlay/tet:backup-path/tet:path-element/tet:type
        /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---: (wson)
+---rw (grid-type)?

```

```

    +---:(dwdm)
    |   +---rw (single-or-super-channel)?
    |       +---:(single)
    |           |   +---rw dwdm-n?           10-types:dwdm-n
    |       +---:(super)
    |           +---rw subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:optimizations/tet:algorithm/tet:metric
    /tet:optimization-metric
    /tet:explicit-route-exclude-objects
    /tet:route-object-exclude-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw (single-or-super-channel)?
|       +---:(single)
|           |   +---rw dwdm-n?           10-types:dwdm-n
|       +---:(super)
|           +---rw subcarrier-dwdm-n*    10-types:dwdm-n
+---:(cwdm)
|   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:optimizations/tet:algorithm/tet:metric
    /tet:optimization-metric
    /tet:explicit-route-include-objects
    /tet:route-object-include-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw (single-or-super-channel)?
|       +---:(single)
|           |   +---rw dwdm-n?           10-types:dwdm-n
|       +---:(super)
|           +---rw subcarrier-dwdm-n*    10-types:dwdm-n
+---:(cwdm)
|   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:path-properties/tet:path-route-objects
    /tet:path-route-object/tet:type/tet:label/tet:label-hop
    /tet:te-label/tet:technology:
+---:(wson)

```



```

    +---ro (grid-type)?
      +---:(dwdm)
        | +---ro (single-or-super-channel)?
        |   +---:(single)
        |     | +---ro dwdm-n?          10-types:dwdm-n
        |     +---:(super)
        |       +---ro subcarrier-dwdm-n* 10-types:dwdm-n
      +---:(cwdm)
        +---ro cwdm-n?          10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:from/tet:label-restrictions
  /tet:label-restriction/tet:label-start/tet:te-label
  /tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      | +---rw dwdm-n?  10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:from/tet:label-restrictions
  /tet:label-restriction/tet:label-end/tet:te-label
  /tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      | +---rw dwdm-n?  10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:from/tet:label-restrictions
  /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
  +---rw (l0-grid-type)?
    +---:(dwdm)
      | +---rw wson-dwdm-channel-spacing?  identityref
    +---:(cwdm)
      +---rw wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:to/tet:label-restrictions
  /tet:label-restriction/tet:label-start/tet:te-label
  /tet:technology:
+---:(wson)
  +---rw (grid-type)?

```

```

      +---:(dwdm)
      |   +---rw dwdm-n?   10-types:dwdm-n
      +---:(cwdm)
      |   +---rw cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
      /tet:te-node-attributes/tet:connectivity-matrices
      /tet:connectivity-matrix/tet:to/tet:label-restrictions
      /tet:label-restriction/tet:label-end/tet:te-label
      /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?   10-types:dwdm-n
+---:(cwdm)
|   +---rw cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
      /tet:te-node-attributes/tet:connectivity-matrices
      /tet:connectivity-matrix/tet:to/tet:label-restrictions
      /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
+---rw (l0-grid-type)?
+---:(dwdm)
|   +---rw wson-dwdm-channel-spacing?   identityref
+---:(cwdm)
|   +---rw wson-cwdm-channel-spacing?   identityref
augment /nw:networks/nw:network/nw:node/tet:te
      /tet:te-node-attributes/tet:connectivity-matrices
      /tet:connectivity-matrix/tet:underlay/tet:primary-path
      /tet:path-element/tet:type/tet:label/tet:label-hop
      /tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw (single-or-super-channel)?
|   |   +---:(single)
|   |   |   +---rw dwdm-n?   10-types:dwdm-n
|   |   +---:(super)
|   |   |   +---rw subcarrier-dwdm-n*   10-types:dwdm-n
+---:(cwdm)
|   +---rw cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
      /tet:te-node-attributes/tet:connectivity-matrices
      /tet:connectivity-matrix/tet:underlay/tet:backup-path
      /tet:path-element/tet:type/tet:label/tet:label-hop
      /tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)

```

```

    +---rw (single-or-super-channel)?
        +---:(single)
            | +---rw dwdm-n?                10-types:dwdm-n
        +---:(super)
            +---rw subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
        +---rw cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:optimizations/tet:algorithm
    /tet:metric/tet:optimization-metric
    /tet:explicit-route-exclude-objects
    /tet:route-object-exclude-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
        +---:(dwdm)
            | +---rw (single-or-super-channel)?
                +---:(single)
                    | +---rw dwdm-n?                10-types:dwdm-n
                +---:(super)
                    +---rw subcarrier-dwdm-n*    10-types:dwdm-n
            +---:(cwdm)
                +---rw cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:optimizations/tet:algorithm
    /tet:metric/tet:optimization-metric
    /tet:explicit-route-include-objects
    /tet:route-object-include-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
        +---:(dwdm)
            | +---rw (single-or-super-channel)?
                +---:(single)
                    | +---rw dwdm-n?                10-types:dwdm-n
                +---:(super)
                    +---rw subcarrier-dwdm-n*    10-types:dwdm-n
            +---:(cwdm)
                +---rw cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:path-properties
    /tet:path-route-objects/tet:path-route-object/tet:type
    /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?

```

```

    +---:(dwdm)
    |   +---ro (single-or-super-channel)?
    |   |   +---:(single)
    |   |   |   +---ro dwdm-n?          10-types:dwdm-n
    |   |   +---:(super)
    |   |   |   +---ro subcarrier-dwdm-n* 10-types:dwdm-n
    +---:(cwdm)
    |   +---ro cwdm-n?          10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:label-restrictions/tet:label-restriction
    /tet:label-start/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |   +---ro dwdm-n?  10-types:dwdm-n
    +---:(cwdm)
    |   +---ro cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:label-restrictions/tet:label-restriction
    /tet:label-end/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |   +---ro dwdm-n?  10-types:dwdm-n
    +---:(cwdm)
    |   +---ro cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:label-restrictions/tet:label-restriction
    /tet:label-step/tet:technology:
+---:(wson)
    +---ro (l0-grid-type)?
    +---:(dwdm)
    |   +---ro wson-dwdm-channel-spacing?  identityref
    +---:(cwdm)
    |   +---ro wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:underlay/tet:primary-path/tet:path-element/tet:type
    /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |   +---ro (single-or-super-channel)?
    |   |   +---:(single)
    |   |   |   +---ro dwdm-n?          10-types:dwdm-n

```

```

    |         +---:(super)
    |         +---ro subcarrier-dwdm-n*    10-types:dwdm-n
+---:(cwdm)
    |         +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:underlay/tet:backup-path/tet:path-element/tet:type
    /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro (single-or-super-channel)?
    |         +---:(single)
    |         |         +---ro dwdm-n?                10-types:dwdm-n
    |         +---:(super)
    |         +---ro subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |         +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:optimizations/tet:algorithm/tet:metric
    /tet:optimization-metric
    /tet:explicit-route-exclude-objects
    /tet:route-object-exclude-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro (single-or-super-channel)?
    |         +---:(single)
    |         |         +---ro dwdm-n?                10-types:dwdm-n
    |         +---:(super)
    |         +---ro subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |         +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:optimizations/tet:algorithm/tet:metric
    /tet:optimization-metric
    /tet:explicit-route-include-objects
    /tet:route-object-include-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro (single-or-super-channel)?
    |         +---:(single)
    |         |         +---ro dwdm-n?                10-types:dwdm-n

```

```

    |         +---:(super)
    |         +---ro subcarrier-dwdm-n*    10-types:dwdm-n
+---:(cwdm)
    |         +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:path-properties/tet:path-route-objects
    /tet:path-route-object/tet:type/tet:label/tet:label-hop
    /tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro (single-or-super-channel)?
    |         +---:(single)
    |         |         +---ro dwdm-n?                10-types:dwdm-n
    |         +---:(super)
    |         +---ro subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |         +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:from/tet:label-restrictions
    /tet:label-restriction/tet:label-start/tet:te-label
    /tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro dwdm-n?    10-types:dwdm-n
    +---:(cwdm)
    |         +---ro cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:from/tet:label-restrictions
    /tet:label-restriction/tet:label-end/tet:te-label
    /tet:technology:
+---:(wson)
    +---ro (grid-type)?
    +---:(dwdm)
    |         +---ro dwdm-n?    10-types:dwdm-n
    +---:(cwdm)
    |         +---ro cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:from/tet:label-restrictions
    /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
    +---ro (l0-grid-type)?
    +---:(dwdm)

```

```

    | +--ro wson-dwdm-channel-spacing?  identityref
    +--:(cwdm)
      +--ro wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:to/tet:label-restrictions
  /tet:label-restriction/tet:label-start/tet:te-label
  /tet:technology:
+--:(wson)
  +--ro (grid-type)?
  +--:(dwdm)
    | +--ro dwdm-n?  10-types:dwdm-n
    +--:(cwdm)
      +--ro cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:to/tet:label-restrictions
  /tet:label-restriction/tet:label-end/tet:te-label
  /tet:technology:
+--:(wson)
  +--ro (grid-type)?
  +--:(dwdm)
    | +--ro dwdm-n?  10-types:dwdm-n
    +--:(cwdm)
      +--ro cwdm-n?  10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:to/tet:label-restrictions
  /tet:label-restriction/tet:label-step/tet:technology:
+--:(wson)
  +--ro (l0-grid-type)?
  +--:(dwdm)
    | +--ro wson-dwdm-channel-spacing?  identityref
    +--:(cwdm)
      +--ro wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:underlay/tet:primary-path
  /tet:path-element/tet:type/tet:label/tet:label-hop
  /tet:te-label/tet:technology:
+--:(wson)
  +--ro (grid-type)?
  +--:(dwdm)
    | +--ro (single-or-super-channel)?
    |   +--:(single)
    |   | +--ro dwdm-n?  10-types:dwdm-n
    |   +--:(super)
    |   | +--ro subcarrier-dwdm-n*  10-types:dwdm-n

```

```

      +---:(cwdm)
        +---ro cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:underlay/tet:backup-path
  /tet:path-element/tet:type/tet:label/tet:label-hop
  /tet:te-label/tet:technology:
+---:(wson)
  +---ro (grid-type)?
    +---:(dwdm)
      +---ro (single-or-super-channel)?
        +---:(single)
          | +---ro dwdm-n?                               10-types:dwdm-n
          +---:(super)
            +---ro subcarrier-dwdm-n*   10-types:dwdm-n
    +---:(cwdm)
      +---ro cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:optimizations/tet:algorithm
  /tet:metric/tet:optimization-metric
  /tet:explicit-route-exclude-objects
  /tet:route-object-exclude-object/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---ro (grid-type)?
    +---:(dwdm)
      +---ro (single-or-super-channel)?
        +---:(single)
          | +---ro dwdm-n?                               10-types:dwdm-n
          +---:(super)
            +---ro subcarrier-dwdm-n*   10-types:dwdm-n
    +---:(cwdm)
      +---ro cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix/tet:optimizations/tet:algorithm
  /tet:metric/tet:optimization-metric
  /tet:explicit-route-include-objects
  /tet:route-object-include-object/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---ro (grid-type)?
    +---:(dwdm)
      +---ro (single-or-super-channel)?
        +---:(single)
          | +---ro dwdm-n?                               10-types:dwdm-n
          +---:(super)

```



```

    |          +---ro subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |          +---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix/tet:path-properties
    /tet:path-route-objects/tet:path-route-object/tet:type
    /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
+---ro (grid-type)?
+---:(dwdm)
|   +---ro (single-or-super-channel)?
|   |   +---:(single)
|   |   |   +---ro dwdm-n?                10-types:dwdm-n
|   |   +---:(super)
|   |   |   +---ro subcarrier-dwdm-n*    10-types:dwdm-n
+---:(cwdm)
+---ro cwdm-n?                10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities/tet:label-restrictions
    /tet:label-restriction/tet:label-start/tet:te-label
    /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?    10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities/tet:label-restrictions
    /tet:label-restriction/tet:label-end/tet:te-label
    /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?    10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?    10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities/tet:label-restrictions
    /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
+---rw (l0-grid-type)?
+---:(dwdm)
|   +---rw wson-dwdm-channel-spacing?    identityref

```

```

      +---:(cwdm)
        +---rw wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities/tet:underlay
  /tet:primary-path/tet:path-element/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      +---rw (single-or-super-channel)?
        +---:(single)
          | +---rw dwdm-n? 10-types:dwdm-n
        +---:(super)
          +---rw subcarrier-dwdm-n* 10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n? 10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities/tet:underlay
  /tet:backup-path/tet:path-element/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      +---rw (single-or-super-channel)?
        +---:(single)
          | +---rw dwdm-n? 10-types:dwdm-n
        +---:(super)
          +---rw subcarrier-dwdm-n* 10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n? 10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities/tet:optimizations
  /tet:algorithm/tet:metric/tet:optimization-metric
  /tet:explicit-route-exclude-objects
  /tet:route-object-exclude-object/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      +---rw (single-or-super-channel)?
        +---:(single)
          | +---rw dwdm-n? 10-types:dwdm-n
        +---:(super)
          +---rw subcarrier-dwdm-n* 10-types:dwdm-n
    +---:(cwdm)

```

```

      +---rw cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities/tet:optimizations
  /tet:algorithm/tet:metric/tet:optimization-metric
  /tet:explicit-route-include-objects
  /tet:route-object-include-object/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
  +---:(dwdm)
  |   +---rw (single-or-super-channel)?
  |   |   +---:(single)
  |   |   |   +---rw dwdm-n?                       10-types:dwdm-n
  |   |   +---:(super)
  |   |   |   +---rw subcarrier-dwdm-n*           10-types:dwdm-n
  |   +---:(cwdm)
  |   +---rw cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities/tet:path-properties
  /tet:path-route-objects/tet:path-route-object/tet:type
  /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---ro (grid-type)?
  +---:(dwdm)
  |   +---ro (single-or-super-channel)?
  |   |   +---:(single)
  |   |   |   +---ro dwdm-n?                       10-types:dwdm-n
  |   |   +---:(super)
  |   |   |   +---ro subcarrier-dwdm-n*           10-types:dwdm-n
  |   +---:(cwdm)
  |   +---ro cwdm-n?                               10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities
  /tet:local-link-connectivity/tet:label-restrictions
  /tet:label-restriction/tet:label-start/tet:te-label
  /tet:technology:
+---:(wson)
  +---rw (grid-type)?
  +---:(dwdm)
  |   +---rw dwdm-n?   10-types:dwdm-n
  +---:(cwdm)
  |   +---rw cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities

```

```

        /tet:local-link-connectivity/tet:label-restrictions
        /tet:label-restriction/tet:label-end/tet:te-label
        /tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      | +---rw dwdm-n?      10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n?      10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities
  /tet:local-link-connectivity/tet:label-restrictions
  /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
  +---rw (l0-grid-type)?
    +---:(dwdm)
      | +---rw wson-dwdm-channel-spacing?  identityref
    +---:(cwdm)
      +---rw wson-cwdm-channel-spacing?  identityref
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities
  /tet:local-link-connectivity/tet:underlay
  /tet:primary-path/tet:path-element/tet:type/tet:label
  /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      | +---rw (single-or-super-channel)?
      |   +---:(single)
      |   | +---rw dwdm-n?      10-types:dwdm-n
      |   +---:(super)
      |   +---rw subcarrier-dwdm-n*  10-types:dwdm-n
    +---:(cwdm)
      +---rw cwdm-n?      10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities
  /tet:local-link-connectivity/tet:underlay/tet:backup-path
  /tet:path-element/tet:type/tet:label/tet:label-hop
  /tet:te-label/tet:technology:
+---:(wson)
  +---rw (grid-type)?
    +---:(dwdm)
      | +---rw (single-or-super-channel)?
      |   +---:(single)
      |   | +---rw dwdm-n?      10-types:dwdm-n

```

```

    |         +---:(super)
    |         |         +---rw subcarrier-dwdm-n*      10-types:dwdm-n
    +---:(cwdm)
    |         +---rw cwdm-n?                          10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities
    /tet:local-link-connectivity/tet:optimizations
    /tet:algorithm/tet:metric/tet:optimization-metric
    /tet:explicit-route-exclude-objects
    /tet:route-object-exclude-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
    +---:(dwdm)
    |         +---rw (single-or-super-channel)?
    |         |         +---:(single)
    |         |         |         +---rw dwdm-n?      10-types:dwdm-n
    |         |         +---:(super)
    |         |         +---rw subcarrier-dwdm-n*      10-types:dwdm-n
    +---:(cwdm)
    |         +---rw cwdm-n?                          10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities
    /tet:local-link-connectivity/tet:optimizations
    /tet:algorithm/tet:metric/tet:optimization-metric
    /tet:explicit-route-include-objects
    /tet:route-object-include-object/tet:type/tet:label
    /tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
    +---:(dwdm)
    |         +---rw (single-or-super-channel)?
    |         |         +---:(single)
    |         |         |         +---rw dwdm-n?      10-types:dwdm-n
    |         |         +---:(super)
    |         |         +---rw subcarrier-dwdm-n*      10-types:dwdm-n
    +---:(cwdm)
    |         +---rw cwdm-n?                          10-types:cwdm-n
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:tunnel-termination-point
    /tet:local-link-connectivities
    /tet:local-link-connectivity/tet:path-properties
    /tet:path-route-objects/tet:path-route-object/tet:type
    /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(wson)
    +---ro (grid-type)?

```

```

    +---:(dwdm)
    |   +---ro (single-or-super-channel)?
    |       +---:(single)
    |           |   +---ro dwdm-n?           10-types:dwdm-n
    |       +---:(super)
    |           +---ro subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |   +---ro cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:underlay/tet:primary-path
    /tet:path-element/tet:type/tet:label/tet:label-hop
    /tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
    +---:(dwdm)
    |   +---rw (single-or-super-channel)?
    |       +---:(single)
    |           |   +---rw dwdm-n?           10-types:dwdm-n
    |       +---:(super)
    |           +---rw subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:underlay/tet:backup-path
    /tet:path-element/tet:type/tet:label/tet:label-hop
    /tet:te-label/tet:technology:
+---:(wson)
    +---rw (grid-type)?
    +---:(dwdm)
    |   +---rw (single-or-super-channel)?
    |       +---:(single)
    |           |   +---rw dwdm-n?           10-types:dwdm-n
    |       +---:(super)
    |           +---rw subcarrier-dwdm-n*    10-types:dwdm-n
    +---:(cwdm)
    |   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:label-restrictions
    /tet:label-restriction/tet:label-start/tet:te-label
    /tet:technology:
+---:(wson)
    +---rw (grid-type)?
    +---:(dwdm)
    |   +---rw dwdm-n?           10-types:dwdm-n
    +---:(cwdm)
    |   +---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:label-restrictions

```

```

        /tet:label-restriction/tet:label-end/tet:te-label
        /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?   10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
        /tet:te-link-attributes/tet:label-restrictions
        /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
+---rw (l0-grid-type)?
+---:(dwdm)
|   +---rw wson-dwdm-channel-spacing?   identityref
+---:(cwdm)
+---rw wson-cwdm-channel-spacing?   identityref
augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:label-restrictions
        /tet:label-restriction/tet:label-start/tet:te-label
        /tet:technology:
+---:(wson)
+---ro (grid-type)?
+---:(dwdm)
|   +---ro dwdm-n?   10-types:dwdm-n
+---:(cwdm)
+---ro cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:label-restrictions
        /tet:label-restriction/tet:label-end/tet:te-label
        /tet:technology:
+---:(wson)
+---ro (grid-type)?
+---:(dwdm)
|   +---ro dwdm-n?   10-types:dwdm-n
+---:(cwdm)
+---ro cwdm-n?   10-types:cwdm-n
augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:label-restrictions
        /tet:label-restriction/tet:label-step/tet:technology:
+---:(wson)
+---ro (l0-grid-type)?
+---:(dwdm)
|   +---ro wson-dwdm-channel-spacing?   identityref
+---:(cwdm)
+---ro wson-cwdm-channel-spacing?   identityref
augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:underlay/tet:primary-path

```

```

        /tet:path-element/tet:type/tet:label/tet:label-hop
        /tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw (single-or-super-channel)?
|   |   +---:(single)
|   |   |   +---rw dwdm-n?           10-types:dwdm-n
|   |   +---:(super)
|   |       +---rw subcarrier-dwdm-n* 10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:underlay/tet:backup-path
        /tet:path-element/tet:type/tet:label/tet:label-hop
        /tet:te-label/tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw (single-or-super-channel)?
|   |   +---:(single)
|   |   |   +---rw dwdm-n?           10-types:dwdm-n
|   |   +---:(super)
|   |       +---rw subcarrier-dwdm-n* 10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?           10-types:cwdm-n
augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:label-restrictions
        /tet:label-restriction/tet:label-start/tet:te-label
        /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?       10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?          10-types:cwdm-n
augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:label-restrictions
        /tet:label-restriction/tet:label-end/tet:te-label
        /tet:technology:
+---:(wson)
+---rw (grid-type)?
+---:(dwdm)
|   +---rw dwdm-n?       10-types:dwdm-n
+---:(cwdm)
+---rw cwdm-n?          10-types:cwdm-n
augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:label-restrictions

```



```

        /tet:label-restriction/tet:label-step/tet:technology:
+--:(wson)
  +--rw (l0-grid-type)?
    +--:(dwdm)
    |   +--rw wson-dwdm-channel-spacing?   identityref
    +--:(cwdm)
    |   +--rw wson-cwdm-channel-spacing?   identityref

```

3. The YANG Code for WSON topology

```

<CODE BEGINS> file "ietf-wson-topology@2020-10-16.yang"
module ietf-wson-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-wson-topology";
  prefix "wson";

  import ietf-network {
    prefix "nw";
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  import ietf-network-topology {
    prefix "nt";
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  import ietf-te-topology {
    prefix "tet";
    reference
      "RFC 8795: YANG Data Model for
      Traffic Engineering (TE) Topologies";
  }

  import ietf-layer0-types {
    prefix "l0-types";
    reference
      "RFC XXXX: A YANG Data Model for Layer 0 Types";
  }

  /* Note: The RFC Editor will replace XXXX with the number assigned
     to the RFC once draft-ietf-ccamp-layer0-types becomes an RFC.*/

  organization

```

```
"IETF CCAMP Working Group";

contact
  "WG Web: <http://tools.ietf.org/wg/ccamp/>
  WG List: <mailto:ccamp@ietf.org>

  Editor: Haomian Zheng
          <mailto:zhenghaomian@huawei.com>

  Editor: Young Lee
          <younglee.tx@gmail.com>

  Editor: Aihua Guo
          <aihuaguo.ietf@gmail.com>
  Editor: Victor Lopez
          <victor.lopezalvarez@telefonica.com>

  Editor: Daniel King
          <d.king@lancaster.ac.uk>";

description
  "This module provides a YANG data model for the routing and
  wavelength assignment (RWA) Traffic Engineering (TE)
  topology in wavelength switched optical networks (WSOYs).
  The YANG model described in this document is a WSOY
  technology-specific YANG model augmenting the generic TE
  topology module (ietf-te-topology) based on the information
  model developed in RFC 7446 and the two encoding documents
  RFC 7579 and RFC 7581.

  Copyright (c) 2020 IETF Trust and the persons identified
  as authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with
  or without modification, is permitted pursuant to, and
  subject to the license terms contained in, the Simplified
  BSD License set forth in Section 4.c of the IETF Trust's
  Legal Provisions Relating to IETF Documents
  (http://trustee.ietf.org/license-info).

  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

revision 2020-10-16 {
  description
    "Initial Version";
  reference
    "RFC XXXX: A YANG Data Model for WSOY (Wavelength Switched
```

```

        Optical Networks)";
    // RFC Ed.: replace XXXX with actual RFC number, update date
    // information and remove this note
}

/*
 * Data nodes
 */

augment "/nw:networks/nw:network/nw:network-types"
  + "/tet:te-topology" {
  description
    "Augment network types to define WSON topology type.";
  container wson-topology {
    presence
      "Its presence identifies the WSON topology type.";
    description
      "Introduce new network type for WSON topology.";
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te"
  + "/tet:te-node-attributes" {
  when "/nw:networks/nw:network/nw:network-types"
    + "/tet:te-topology/wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
      WSON topology type.";
  }
  description "Augment TE node attributes.";
  container wson-node {
    presence "The TE node is a WSON node.";
    description "WSON node attributes";
    leaf is-reconfigurable-node {
      type boolean;
      default true;
      description
        "Indicates whether the WSON node is reconfigurable:
        - true: the node is reconfigurable, i.e.,
          it is representing a ROADM node;
        - false: the node is not reconfigurable, i.e.,
          it is representing a FOADM node.";
    }
  }
}

/*
 * Augment TE label range information

```

```
*/  
augment "/nw:networks/nw:network/nw:node/tet:te/"  
    + "tet:te-node-attributes/tet:connectivity-matrices/"  
    + "tet:label-restrictions/tet:label-restriction" {  
when "../.../../.../../nw:network-types/tet:te-topology/"  
    + "wson:wson-topology" {  
description  
    "Augmentation parameters apply only for networks with  
        WSON topology type.";  
}  
description  
    "Augment TE label range information for the TE node  
        connectivity matrices.";  
uses l0-types:l0-label-range-info;  
}  
  
augment "/nw:networks/nw:network/nw:node/tet:te/"  
    + "tet:te-node-attributes/tet:connectivity-matrices/"  
    + "tet:connectivity-matrix/tet:from/"  
    + "tet:label-restrictions/tet:label-restriction" {  
when "../.../../.../../.../../nw:network-types/tet:te-topology/"  
    + "wson:wson-topology" {  
description  
    "Augmentation parameters apply only for networks with  
        WSON topology type.";  
}  
description  
    "Augment TE label range information for the source LTP  
        of the connectivity matrix entry.";  
uses l0-types:l0-label-range-info;  
}  
  
augment "/nw:networks/nw:network/nw:node/tet:te/"  
    + "tet:te-node-attributes/tet:connectivity-matrices/"  
    + "tet:connectivity-matrix/tet:to/"  
    + "tet:label-restrictions/tet:label-restriction" {  
when "../.../../.../../.../../nw:network-types/tet:te-topology/"  
    + "wson:wson-topology" {  
description  
    "Augmentation parameters apply only for networks with  
        WSON topology type.";  
}  
description  
    "Augment TE label range information for the destination LTP  
        of the connectivity matrix entry.";  
uses l0-types:l0-label-range-info;  
}
```

```

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/"
  + "tet:connectivity-matrices/tet:label-restrictions/"
  + "tet:label-restriction" {
when "../../../../../../../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the TE node
  connectivity matrices information source.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:from/tet:label-restrictions/tet:label-restriction" {
when "../../../../../../../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the source LTP
  of the connectivity matrix entry information source.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:to/tet:label-restrictions/tet:label-restriction" {
when "../../../../../../../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the destination LTP
  of the connectivity matrix entry information source.";
uses l0-types:l0-label-range-info;
}

```

```

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:label-restrictions/tet:label-restriction" {
when "../..../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the TTP
  Local Link Connectivities.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:local-link-connectivity/"
  + "tet:label-restrictions/tet:label-restriction" {
when "../..../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the TTP
  Local Link Connectivity entry.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:label-restrictions/tet:label-restriction" {
when "../..../nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range information for the TE link.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/nw:network/nt:link/tet:te/"

```

```
+ "tet:information-source-entry/"
+ "tet:label-restrictions/tet:label-restriction" {
when "../.../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
      WSON topology type.";
}
description
    "Augment TE label range information for the TE link
      information source.";
uses l0-types:l0-label-range-info;
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction" {
description
    "Augment TE label range information for the TE link template.";
uses l0-types:l0-label-range-info;
}

/*
 * Augment TE label
 */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:label-restrictions/tet:label-restriction/"
+ "tet:label-start/"
+ "tet:te-label/tet:technology" {
when "../.../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
      WSON topology type.";
}
description
    "Augment TE label range start for the TE node
      connectivity matrices";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:label-restrictions/"
```

```
+ "tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../../../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range end for the TE node
        connectivity matrices";
case wson {
uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:label-restrictions/"
+ "tet:label-restriction/tet:label-step/"
+ "tet:technology" {
when "../../../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range step for the TE node
        connectivity matrices";
case wson {
uses l0-types:wson-label-step;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:underlay/tet:primary-path/tet:path-element/"
+ "tet:type/tet:label/tet:label-hop/"
+ "tet:te-label/tet:technology" {
when "../../../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
```



```

    "Augment TE label hop for the underlay primary path of the
      TE node connectivity matrices";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:underlay/tet:backup-path/tet:path-element/"
  + "tet:type/tet:label/tet:label-hop/"
  + "tet:te-label/tet:technology" {
  when "../../../../../../../../../../../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
        WSON topology type.";
  }
  description
    "Augment TE label hop for the underlay backup path of the
      TE node connectivity matrices";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:optimizations/tet:algorithm/tet:metric/"
  + "tet:optimization-metric/"
  + "tet:explicit-route-exclude-objects/"
  + "tet:route-object-exclude-object/"
  + "tet:type/tet:label/tet:label-hop/"
  + "tet:te-label/tet:technology" {
  when "../../../../../../../../../../../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
        WSON topology type.";
  }
  description
    "Augment TE label hop for the explicit route objects excluded
      by the path computation of the TE node connectivity
      matrices";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

```

```

    }
  }

  augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-include-objects/"
    + "tet:route-object-include-object/"
    + "tet:type/tet:label/tet:label-hop/"
    + "tet:te-label/tet:technology" {
  when "../../../../../../../../../../../../../../../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
       WSON topology type.";
  }
  description
    "Augment TE label hop for the explicit route objects included
     by the path computation of the TE node connectivity
     matrices";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:path-properties/tet:path-route-objects/"
  + "tet:path-route-object/tet:type/tet:label/tet:label-hop/"
  + "tet:te-label/tet:technology" {
  when "../../../../../../../../../../../../../../../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
       WSON topology type.";
  }
  description
    "Augment TE label hop for the computed path route objects
     of the TE node connectivity matrices";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"

```

```

    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/tet:from/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-start/"
    + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label range start for the source LTP
    of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/tet:from/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-end/"
    + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label range end for the source LTP
    of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/tet:from/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-step/"
    + "tet:technology" {
when "../.../.../.../.../.../.../.../.../..."

```

```

    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
    }
description
    "Augment TE label range step for the source LTP
    of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-step;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/tet:to/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-start/"
    + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
    }
description
    "Augment TE label range start for the destination LTP
    of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/tet:to/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-end/"
    + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
    }
}

```

```

    description
      "Augment TE label range end for the destination LTP
      of the connectivity matrix entry.";
    case wson {
      uses 10-types:wson-label-start-end;
    }
  }

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/tet:to/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-step/"
  + "tet:technology" {
when "../.../.../.../.../.../.../.../.../"
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range step for the destination LTP
  of the connectivity matrix entry.";
case wson {
  uses 10-types:wson-label-step;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:underlay/tet:primary-path/tet:path-element/"
  + "tet:type/tet:label/tet:label-hop/"
  + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../"
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the underlay primary path
  of the connectivity matrix entry.";
case wson {
  uses 10-types:wson-label-hop;
}
}

```

```

}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:underlay/tet:backup-path/tet:path-element/"
  + "tet:type/tet:label/tet:label-hop/"
  + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the underlay backup path
  of the connectivity matrix entry.";
case wson {
  uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/tet:optimizations/"
  + "tet:algorithm/tet:metric/tet:optimization-metric/"
  + "tet:explicit-route-exclude-objects/"
  + "tet:route-object-exclude-object/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the explicit route objects excluded
  by the path computation of the connectivity matrix entry.";
case wson {
  uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:te-node-attributes/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/tet:optimizations/"

```

```

    + "tet:algorithm/tet:metric/tet:optimization-metric/"
    + "tet:explicit-route-include-objects/"
    + "tet:route-object-include-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSOON topology type.";
}
description
    "Augment TE label hop for the explicit route objects included
    by the path computation of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:path-properties/tet:path-route-objects/"
    + "tet:path-route-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSOON topology type.";
}
description
    "Augment TE label hop for the computed path route objects
    of the connectivity matrix entry.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/"
    + "tet:connectivity-matrices/tet:label-restrictions/"
    + "tet:label-restriction/"
    + "tet:label-start/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {

```

```

        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range start for the TE node connectivity
        matrices information source.";
    case wson {
        uses l0-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/"
    + "tet:connectivity-matrices/tet:label-restrictions/"
    + "tet:label-restriction/"
    + "tet:label-end/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range end for the TE node connectivity
        matrices information source.";
    case wson {
        uses l0-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/"
    + "tet:connectivity-matrices/tet:label-restrictions/"
    + "tet:label-restriction/"
    + "tet:label-step/tet:technology" {
    when "../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range step for the TE node connectivity
        matrices information source.";
    case wson {

```



```

    uses l0-types:wson-label-step;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
  when "../.../.../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
       WSON topology type.";
  }
  description
    "Augment TE label hop for the underlay primary path
     of the TE node connectivity matrices of the information
     source entry.";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
  when "../.../.../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
       WSON topology type.";
  }
  description
    "Augment TE label hop for the underlay backup path
     of the TE node connectivity matrices of the information
     source entry.";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:optimizations/tet:algorithm/tet:metric/"
  + "tet:optimization-metric/"

```

```

    + "tet:explicit-route-exclude-objects/"
    + "tet:route-object-exclude-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the explicit route objects excluded
    by the path computation of the TE node connectivity matrices
    information source.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-include-objects/"
    + "tet:route-object-include-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the explicit route objects included
    by the path computation of the TE node connectivity matrices
    information source.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:path-properties/tet:path-route-objects/"
    + "tet:path-route-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."

```

```

    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the computed path route objects
    of the TE node connectivity matrices information source.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:from/tet:label-restrictions/"
    + "tet:label-restriction/"
    + "tet:label-start/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label range start for the source LTP
    of the connectivity matrix entry information source.";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:from/tet:label-restrictions/"
    + "tet:label-restriction/"
    + "tet:label-end/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description

```

```

        "Augment TE label range end for the source LTP
        of the connectivity matrix entry information source.";
    case wson {
        uses l0-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:from/tet:label-restrictions/"
+ "tet:label-restriction/"
+ "tet:label-step/tet:technology" {
when "../../../../../../../../../../../"
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range step for the source LTP
    of the connectivity matrix entry information source.";
case wson {
    uses l0-types:wson-label-step;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:to/tet:label-restrictions/tet:label-restriction/"
+ "tet:label-start/tet:te-label/tet:technology" {
when "../../../../../../../../../../../"
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range start for the destination LTP
    of the connectivity matrix entry information source.";
case wson {
    uses l0-types:wson-label-start-end;
}
}
}

```

```

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:to/tet:label-restrictions/tet:label-restriction/"
  + "tet:label-end/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range end for the destination LTP
  of the connectivity matrix entry information source.";
case wson {
  uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:to/tet:label-restrictions/tet:label-restriction/"
  + "tet:label-step/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label range step for the destination LTP
  of the connectivity matrix entry information source.";
case wson {
  uses l0-types:wson-label-step;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:information-source-entry/tet:connectivity-matrices/"
  + "tet:connectivity-matrix/"
  + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {

```

```

        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label hop for the underlay primary path
        of the connectivity matrix entry information source.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label hop for the underlay backup path
        of the connectivity matrix entry information source.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-exclude-objects/"
    + "tet:route-object-exclude-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description

```

```

    "Augment TE label hop for the explicit route objects excluded
    by the path computation of the connectivity matrix entry
    information source.";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:optimizations/tet:algorithm/tet:metric/"
+ "tet:optimization-metric/"
+ "tet:explicit-route-include-objects/"
+ "tet:route-object-include-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the explicit route objects included
  by the path computation of the connectivity matrix entry
  information source.";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the computed path route objects
  of the connectivity matrix entry information source.";

```

```

    case wson {
      uses 10-types:wson-label-hop;
    }
  }

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-start/"
  + "tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
      WSON topology type.";
}
description
  "Augment TE label range start for the TTP
    Local Link Connectivities.";
case wson {
  uses 10-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-end/"
  + "tet:te-label/tet:technology"{
when "../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
      WSON topology type.";
}
description
  "Augment TE label range end for the TTP
    Local Link Connectivities.";
case wson {
  uses 10-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"

```



```

    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-step/"
    + "tet:technology"{
when "../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
    }
    description
        "Augment TE label range step for the TTP
        Local Link Connectivities.";
    case wson {
        uses l0-types:wson-label-step;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
    }
    description
        "Augment TE label hop for the underlay primary path
        of the TTP Local Link Connectivities.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../"
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {

```

```

        description
            "Augmentation parameters apply only for networks with
              WSON topology type.";
    }
    description
        "Augment TE label hop for the underlay backup path
          of the TTP Local Link Connectivities.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-exclude-objects/"
    + "tet:route-object-exclude-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
              WSON topology type.";
    }
    description
        "Augment TE label hop for the explicit route objects excluded
          by the path computation of the TTP Local Link
          Connectivities.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-include-objects/"
    + "tet:route-object-include-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description

```

```

        "Augmentation parameters apply only for networks with
        WSON topology type.";
    }
    description
        "Augment TE label hop for the explicit route objects included
        by the path computation of the TTP Local Link
        Connectivities.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:path-properties/tet:path-route-objects/"
    + "tet:path-route-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label hop for the computed path route objects
        of the TTP Local Link Connectivities.";
    case wson {
        uses l0-types:wson-label-hop;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:local-link-connectivity/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-start/tet:te-label/tet:technology" {
    when "../.../.../.../.../.../.../.../..."
        + "nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range start for the TTP

```

```

        Local Link Connectivity entry.";
    case wson {
        uses l0-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/"
+ "tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:label-restrictions/tet:label-restriction/"
+ "tet:label-end/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../..."
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range end for the TTP
    Local Link Connectivity entry.";
case wson {
    uses l0-types:wson-label-start-end;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/"
+ "tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:label-restrictions/tet:label-restriction/"
+ "tet:label-step/tet:technology" {
when "../.../.../.../.../.../.../..."
+ "nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
    description
        "Augmentation parameters apply only for networks with
        WSON topology type.";
}
description
    "Augment TE label range step for the TTP
    Local Link Connectivity entry.";
case wson {
    uses l0-types:wson-label-step;
}
}

```

```

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:local-link-connectivity/"
  + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the underlay primary path
  of the TTP Local Link Connectivity entry.";
case wson {
  uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:local-link-connectivity/"
  + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
  + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
  + "nw:network-types/tet:te-topology/"
  + "wson:wson-topology" {
  description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
  "Augment TE label hop for the underlay backup path
  of the TTP Local Link Connectivity entry.";
case wson {
  uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:local-link-connectivity/"
  + "tet:optimizations/tet:algorithm/tet:metric/"
  + "tet:optimization-metric/"

```

```

    + "tet:explicit-route-exclude-objects/"
    + "tet:route-object-exclude-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the explicit route objects excluded
    by the path computation of the TTP Local Link
    Connectivity entry.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:local-link-connectivity/"
    + "tet:optimizations/tet:algorithm/tet:metric/"
    + "tet:optimization-metric/"
    + "tet:explicit-route-include-objects/"
    + "tet:route-object-include-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the explicit route objects included
    by the path computation of the TTP Local Link
    Connectivity entry.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:tunnel-termination-point/"
    + "tet:local-link-connectivities/"
    + "tet:local-link-connectivity/"

```

```

    + "tet:path-properties/tet:path-route-objects/"
    + "tet:path-route-object/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the computed path route objects
    of the TTP Local Link Connectivity entry.";
case wson {
    uses l0-types:wson-label-hop;
}
}
augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:te-link-attributes/"
    + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description
    "Augment TE label hop for the underlay primary path
    of the TE link.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:te-link-attributes/"
    + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
    + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../.../.../.../.../.../..."
    + "nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
    WSON topology type.";
}
description

```

```

    "Augment TE label hop for the underlay backup path
      of the TE link.";
  case wson {
    uses l0-types:wson-label-hop;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-start/tet:te-label/tet:technology" {
  when "../../../../../../../nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
        WSON topology type.";
  }
  description
    "Augment TE label range start for the TE link.";
  case wson {
    uses l0-types:wson-label-start-end;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-end/tet:te-label/tet:technology" {
  when "../../../../../../../nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description
      "Augmentation parameters apply only for networks with
        WSON topology type.";
  }
  description
    "Augment TE label range end for the TE link.";
  case wson {
    uses l0-types:wson-label-start-end;
  }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:label-restrictions/tet:label-restriction/"
  + "tet:label-step/tet:technology" {
  when "../../../../../../../nw:network-types/tet:te-topology/"
    + "wson:wson-topology" {
    description

```



```

        "Augmentation parameters apply only for networks with
        WSON topology type.";
    }
    description
        "Augment TE label range step for the TE link.";
    case wson {
        uses 10-types:wson-label-step;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:information-source-entry/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-start/tet:te-label/tet:technology" {
    when "../..//../..//../..//nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range start for the TE link
        information source.";
    case wson {
        uses 10-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:information-source-entry/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-end/tet:te-label/tet:technology" {
    when "../..//../..//../..//nw:network-types/tet:te-topology/"
        + "wson:wson-topology" {
        description
            "Augmentation parameters apply only for networks with
            WSON topology type.";
    }
    description
        "Augment TE label range end for the TE link
        information source.";
    case wson {
        uses 10-types:wson-label-start-end;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:information-source-entry/"

```

```

+ "tet:label-restrictions/tet:label-restriction/"
+ "tet:label-step/tet:technology" {
when "../.../nw:network-types/tet:te-topology/"
+ "wson:wson-topology" {
description
    "Augmentation parameters apply only for networks with
      WSON topology type.";
}
description
    "Augment TE label range step for the TE link
      information source.";
case wson {
    uses l0-types:wson-label-step;
}
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
description
    "Augment TE label hop for the underlay primary path
      of the TE link template.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
description
    "Augment TE label hop for the underlay backup path
      of the TE link template.";
case wson {
    uses l0-types:wson-label-hop;
}
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction/"
+ "tet:label-start/tet:te-label/tet:technology" {
description
    "Augment TE label range start for the TE link template.";
case wson {
    uses l0-types:wson-label-start-end;

```

```

    }
  }

  augment "/nw:networks/tet:te/tet:templates/"
    + "tet:link-template/tet:te-link-attributes/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-end/tet:te-label/tet:technology" {
    description
      "Augment TE label range end for the TE link template.";
    case wson {
      uses l0-types:wson-label-start-end;
    }
  }

  augment "/nw:networks/tet:te/tet:templates/"
    + "tet:link-template/tet:te-link-attributes/"
    + "tet:label-restrictions/tet:label-restriction/"
    + "tet:label-step/tet:technology" {
    description
      "Augment TE label range step for the TE link template.";
    case wson {
      uses l0-types:wson-label-step;
    }
  }
}

<CODE ENDS>

```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF Protocol over Secure Shell (SSH) [RFC6242] describes a method for invoking and running NETCONF within a Secure Shell (SSH) session as an SSH subsystem. The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the

default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. Considerations in Section 8 of [RFC8795] are also applicable to their subtrees in the module defined in this document.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Considerations in Section 8 of [RFC8795] are also applicable to their subtrees in the module defined in this document.

5. IANA Considerations

It is proposed to IANA to assign new URIs from the "IETF XML Registry" [RFC3688] as follows:

URI: urn:ietf:params:xml:ns:yang:ietf-wson-topology
Registrant Contact: The IESG
XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC7950] and [RFC6020].

name: ietf-wson-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-wson-topology
prefix: wson
reference: RFC XXXX

6. Contributors

Dhruv Dhody
Huawei Technologies
Email: dhruv.ietf@gmail.com

Bin Yeong Yoon
ETRI
Email: byyun@etri.re.kr

Ricard Vilalta
CTTC
Email: vricard.vilalta@cttc.es

Italo Busi
Huawei Technologies
Email: Italo.Busi@huawei.com

7. References

7.1. Normative References

- [I-D.ietf-ccamp-layer0-types]
Zheng, H., Lee, Y., Guo, A., Lopez, V., and D. King, "A YANG Data Model for Layer 0 Types", draft-ietf-ccamp-layer0-types-08 (work in progress), October 2020.
- [ITU-Tg6982]
International Telecommunication Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T G.698.2, November 2018.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7579] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "General Network Element Constraint Encoding for GMPLS-Controlled Networks", RFC 7579, DOI 10.17487/RFC7579, June 2015, <<https://www.rfc-editor.org/info/rfc7579>>.
- [RFC7581] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", RFC 7581, DOI 10.17487/RFC7581, June 2015, <<https://www.rfc-editor.org/info/rfc7581>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

7.2. Informative References

- [I-D.ietf-ccamp-flexigrid-yang]
Madrid, U., Perdices, D., King, D., Lee, Y., and H. Zheng, "A YANG Data Model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang-08 (work in progress), November 2020.
- [I-D.ietf-ccamp-optical-impairment-topology-yang]
Lee, Y., Auge, L., Lopez, V., Galimberti, G., and D. Beller, "A Yang Data Model for Optical Impairment-aware Topology", draft-ietf-ccamp-optical-impairment-topology-yang-05 (work in progress), November 2020.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC7446] Lee, Y., Ed., Bernstein, G., Ed., Li, D., and W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, DOI 10.17487/RFC7446, February 2015, <<https://www.rfc-editor.org/info/rfc7446>>.

Authors' Addresses

Haomian Zheng
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

Young Lee
Samsung
South Korea

Email: younglee.tx@gmail.com

Aihua Guo
Futurewei

Email: aihuaguo.ietf@gmail.com

Victor Lopez
Telefonica

Email: victor.lopezalvarez@telefonica.com

Daniel King
University of Lancaster

Email: d.king@lancaster.ac.uk

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2021

T. Saad
Juniper Networks
K. Raza
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
October 26, 2020

A YANG Data Model for MPLS Base
draft-ietf-mpls-base-yang-17

Abstract

This document contains a specification of the MPLS base YANG data model. The MPLS base YANG data model serves as a base framework for configuring and managing an MPLS switching subsystem on an MPLS-enabled router. It is expected that other MPLS YANG data models (e.g. MPLS Label Switched Path (LSP) Static, LDP or RSVP-TE YANG models) will augment the MPLS base YANG data model.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 29, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Acronyms and Abbreviations	3
2. MPLS Base Model	4
2.1. Model Overview	4
2.2. Model Organization	4
2.3. Model Design	6
2.4. Model Tree Diagram	8
2.5. Model YANG Module	9
3. IANA Considerations	19
4. Security Considerations	20
5. Acknowledgement	21
6. Appendix A. Data Tree Instance Example	21
7. Contributors	27
8. References	27
8.1. Normative References	27
8.2. Informative References	29
Authors' Addresses	29

1. Introduction

A core routing YANG data model is defined in [RFC8349], and it provides a basis for the development of routing data models for specific Address Families (AFs). Specifically, [RFC8349] defines a model for a generic Routing Information Base (RIB) that is Address-Family (AF) agnostic. [RFC8349] also defines two instances of RIBs based on the generic RIB model for IPv4 and IPv6 AFs.

The MPLS base model that is defined in this document augments the generic RIB model defined in [RFC8349] with additional data that enables MPLS forwarding for the specific destination prefix(es) present in the AF RIB(s) as described in the MPLS architecture document [RFC3031].

The MPLS base model also defines a new instance of the generic RIB YANG data model as defined in [RFC8349] to store native MPLS routes. The native MPLS RIB instance stores route(s) that are not associated with other AF instance RIBs (such as IPv4, or IPv6 instance RIB(s)),

but are enabled for MPLS forwarding. Examples of such native MPLS routes are routes programmed by RSVP on transit MPLS router(s) along the path of a Label Switched Path (LSP). Other example(s) are MPLS routes that cross-connect to specific Layer-2 adjacencies, such as Layer-2 Attachment Circuit(s) (ACs)), or Layer-3 adjacencies, such as Segment-Routing (SR) Adjacency Segments (Adj-SIDs) described in [RFC8402].

The MPLS base YANG data model serves as a basis for future development of MPLS YANG data models covering more-sophisticated MPLS feature(s) and sub-system(s). The main purpose is to provide essential building blocks for other YANG data models involving different control-plane protocols, and MPLS functions.

To this end, it is expected that the MPLS base data model will be augmented by a number of other YANG modules developed at IETF (e.g. by TEAS and MPLS working groups).

The YANG module in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

1.1. Terminology

The terminology for describing YANG data models is found in [RFC7950].

1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

RIB: Routing Information Base

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

ILM: Incoming Label Map

2. MPLS Base Model

This document describes the 'ietf-mpls' YANG module that provides base components of the MPLS data model. It is expected that other MPLS YANG modules will augment 'ietf-mpls' YANG module for other MPLS extension to provision Label Switched Paths (LSPs) (e.g. MPLS Static, MPLS LDP or MPLS RSVP-TE LSP(s)).

2.1. Model Overview

This document models MPLS labeled routes as an augmentation of the generic routing RIB data model as defined in [RFC8349]. For example, IP prefix routes (e.g. routes stored in IPv4 or IPv6 RIBs) are augmented to carry additional data to enable it for MPLS forwarding.

This document also defines a new instance of the generic RIB defined in [RFC8349] to store native MPLS route(s) (described further in Section 2.3) by extending the identity 'address-family' defined in [RFC8349] with a new "mpls" identity as suggested in Section 3 of [RFC8349].

2.2. Model Organization

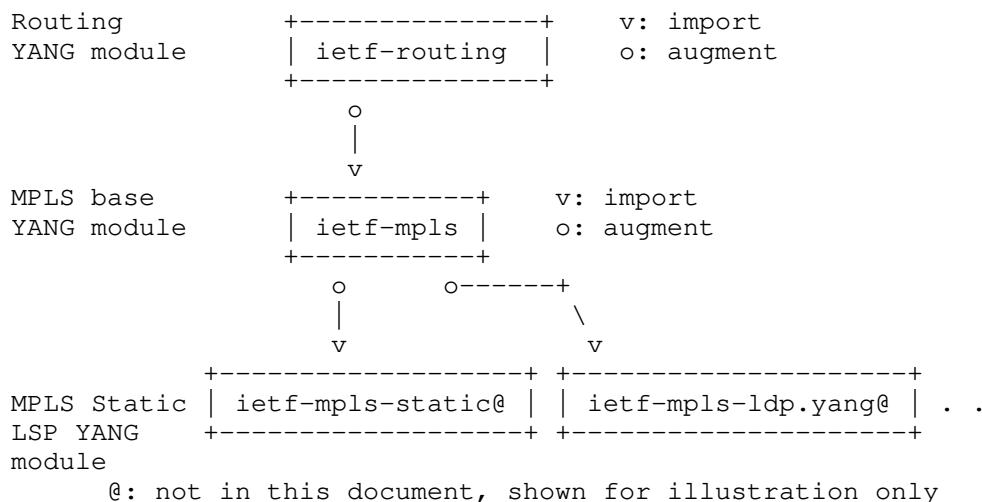


Figure 1: Relationship between MPLS modules

The 'ietf-mpls' YANG module defines the following identities:

mpls:

This identity extends the 'address-family' identity for RIB instance(s) identity as defined in [RFC8349] to represent the native MPLS RIB instance.

label-block-alloc-mode:

A base YANG identity for supported label block allocation mode(s).

The 'ietf-mpls' YANG module contains the following high-level types and groupings:

mpls-operations-type:

An enumeration type that represents support for possible MPLS operation types (impose-and-forward, pop-and-forward, pop-impose-and-forward, and pop-and-lookup)

nhlfe-role:

An enumeration type that represents the role of the NHLFE entry.

nhlfe-single-contents:

A YANG grouping that describes single Next Hop Label Forwarding Entry (NHLFE) and its associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is specific to the case when a single next-hop is associated with the route.

The NHLFE is used when forwarding labeled packet. It contains the following information:

1. the packet's next hop. For 'nhlfe-single-contents' only a single next hop is expected, while for 'nhlfe-multiple-contents' multiple next hops are possible.
2. the operation to perform on the packet's label stack; this can be one of the following operations: a) replace the label at the top of the label stack with one or more specified new label b) pop the label stack c) replace the label at the top of the label stack with a specified new label, and then push one or more specified new labels onto the label stack. d) push one or more label(s) on an unlabeled packet

It may also contain:

- d) the data link encapsulation to use when transmitting the packet
- e) the way to encode the label stack when transmitting the packet
- f) any other information needed in order to properly dispose of the packet.

nhlfe-multiple-contents:

A YANG grouping that describes a set of NHLFE(s) and their associated parameters as described in the MPLS architecture document [RFC3031]. This grouping is used when multiple next-hops are associated with the route.

interfaces-mpls:

A YANG grouping that describes the list of MPLS enabled interfaces on a device.

label-blocks:

A YANG grouping that describes the list of assigned MPLS label blocks and their properties.

rib-mpls-properties:

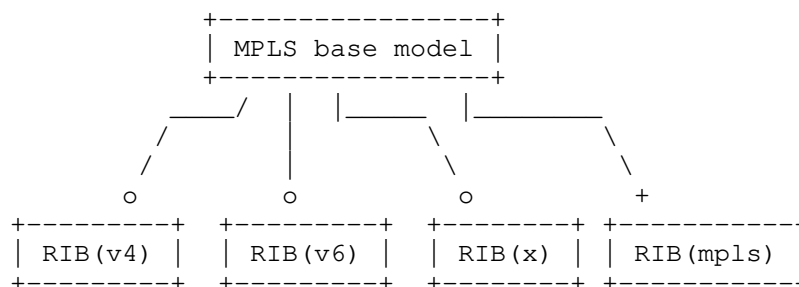
A YANG grouping for the augmentation of the generic RIB with MPLS label forwarding data as defined in [RFC3031].

rib-active-route-mpls-input:

A YANG grouping for the augmentation to the 'active-route' RPC that is specific to the MPLS RIB instance.

2.3. Model Design

The MPLS routing model is based on the core routing data model defined in [RFC8349]. Figure 2 shows the extensions introduced by the MPLS base model on defined RIB(s).



+: created by the MPLS base model
 o: augmented by the MPLS base model

Figure 2: Relationship between MPLS model and RIB instances

As shown in Figure 2, the MPLS base YANG data model augments defined instance(s) of AF RIB(s) with additional data that enables MPLS forwarding for destination prefix(es) store in such RIB(s). For example, an IPv4 prefix stored in RIB(v4) is augmented to carry a MPLS local label and per next-hop remote label(s) to enable MPLS forwarding for such prefix.

The MPLS base model also creates a separate instance of the generic RIB model defined in [RFC8349] to store MPLS native route(s) that are enabled for MPLS forwarding, but not stored in other AF RIB(s).

Some examples of such native MPLS routes are:

- o routes programmed by RSVP on Label Switched Router(s) (LSRs) along the path of a Label Switched Path (LSP),
- o routes that cross-connect an MPLS local label to a Layer-2, or Layer-3 VRF,
- o routes that cross-connect an MPLS local label to a specific Layer-2 adjacency or interface, such as Layer-2 Attachment Circuit(s) (ACs), or
- o routes that cross-connect an MPLS local label to a Layer-3 adjacency or interface - such as MPLS Segment-Routing (SR) Adjacency Segments (Adj-SIDs), SR MPLS Binding SIDs, etc. as defined in [RFC8402].

2.4. Model Tree Diagram

The MPLS base tree diagram that follows the notation defined in [RFC8340] is shown in Figure 3.

```

module: ietf-mpls
augment /rt:routing:
  +--rw mpls
    +--rw ttl-propagate?          boolean
    +--rw mpls-label-blocks
      +--rw mpls-label-block* [index]
        +--rw index              string
        +--rw start-label?       rt-types:mpls-label
        +--rw end-label?         rt-types:mpls-label
        +--rw block-allocation-mode? identityref
        +--ro inuse-labels-count? yang:gauge32
    +--rw interfaces
      +--rw interface* [name]
        +--rw name                if:interface-ref
        +--rw mpls-enabled?       boolean
        +--rw maximum-labeled-packet? uint32
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route:
    +--ro mpls-enabled?          boolean
    +--ro mpls-local-label?      rt-types:mpls-label
    +--ro destination-prefix?    -> ../mpls-local-label
    +--ro route-context?        string
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
    /rt:next-hop-options/rt:simple-next-hop:
    +--ro mpls-label-stack
      +--ro entry* [id]
        +--ro id                 uint8
        +--ro label?             rt-types:mpls-label
        +--ro ttl?               uint8
        +--ro traffic-class?     uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop
    /rt:next-hop-options/rt:next-hop-list/rt:next-hop-list
    /rt:next-hop:
    +--ro index?                 string
    +--ro backup-index?          string
    +--ro loadshare?             uint16
    +--ro role?                  nhlfe-role
    +--ro mpls-label-stack
      +--ro entry* [id]
        +--ro id                 uint8
        +--ro label?             rt-types:mpls-label
        +--ro ttl?               uint8
        +--ro traffic-class?     uint8
  augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input:

```

```

+---w destination-address?  -> ../mpls-local-label
+---w mpls-local-label?     rt-types:mpls-label
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
    /rt:route/rt:next-hop/rt:next-hop-options
    /rt:simple-next-hop:
+--- mpls-label-stack
    +--- entry* [id]
        +--- id                uint8
        +--- label?            rt-types:mpls-label
        +--- ttl?              uint8
        +--- traffic-class?    uint8
augment /rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output
    /rt:route/rt:next-hop/rt:next-hop-options
    /rt:next-hop-list/rt:next-hop-list/rt:next-hop:
+--- index?                  string
+--- backup-index?          string
+--- loadshare?              uint16
+--- role?                   nhlfe-role
+--- mpls-label-stack
    +--- entry* [id]
        +--- id                uint8
        +--- label?            rt-types:mpls-label
        +--- ttl?              uint8
        +--- traffic-class?    uint8

```

Figure 3: MPLS Base tree diagram

2.5. Model YANG Module

This section describes the 'ietf-mpls' YANG module that provides base components of the MPLS data model. Other YANG module(s) may import and augment the base MPLS module to add feature specific data.

The ietf-mpls YANG module imports the following YANG modules:

- o ietf-routing defined in [RFC8349]
- o ietf-routing-types defined in [RFC8294]
- o ietf-interfaces defined in [RFC8343]

This YANG module also references the following RFCs in defining the types and YANG grouping of the YANG module: [RFC3032], [RFC3031], and [RFC7424].

```

<CODE BEGINS> file "ietf-mpls@2020-10-26.yang"
module ietf-mpls {
  yang-version 1.1;

```



```
namespace "urn:ietf:params:xml:ns:yang:ietf-mpls";

/* Replace with IANA when assigned */

prefix mpls;

import ietf-routing {
  prefix rt;
  reference
    "RFC8349: A YANG Data Model for Routing Management";
}
import ietf-routing-types {
  prefix rt-types;
  reference
    "RFC8294: Common YANG Data Types for the Routing Area";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-interfaces {
  prefix if;
  reference
    "RFC8343: A YANG Data Model for Interface Management";
}

organization
  "IETF MPLS Working Group";
contact
  "WG Web:   <http://tools.ietf.org/wg/mpls/>

  WG List:  <mailto:mpls@ietf.org>

  Editor:   Tarek Saad
            <mailto:tsaad@juniper.net>

  Editor:   Kamran Raza
            <mailto:skraza@cisco.com>

  Editor:   Rakesh Gandhi
            <mailto:rgandhi@cisco.com>

  Editor:   Xufeng Liu
            <mailto: xufeng.liu.ietf@gmail.com>

  Editor:   Vishnu Pavan Beeram
            <mailto:vbeeram@juniper.net>";
```

description

"This YANG module defines the essential components for the management of the MPLS subsystem. The model fully conforms to the Network Management Datastore Architecture (NMDA).

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2020-10-26 {

description

"Initial revision.";

reference

"RFC XXXX: A YANG Data Model for base MPLS";

}

/* Identities */

identity mpls {

base rt:address-family;

description

"This identity represents the MPLS address family.";

}

identity mpls-unicast {

base mpls:mpls;

description

"This identity represents the MPLS unicast address family.";

}

identity label-block-alloc-mode {

description

"Base identity for label-block allocation mode.";

}

```
identity label-block-alloc-mode-manager {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by label manager.";
}

identity label-block-alloc-mode-application {
  base label-block-alloc-mode;
  description
    "Label block allocation on reserved block
    is managed by application.";
}

/**
 * Typedefs
 */

typedef mpls-operations-type {
  type enumeration {
    enum impose-and-forward {
      description
        "Operation impose outgoing label(s) and forward to
        next-hop.";
    }
    enum pop-and-forward {
      description
        "Operation pop incoming label and forward to next-hop.";
    }
    enum pop-impose-and-forward {
      description
        "Operation pop incoming label, impose one or more
        outgoing label(s) and forward to next-hop.";
    }
    enum swap-and-forward {
      description
        "Operation swap incoming label, with outgoing label and
        forward to next-hop.";
    }
    enum pop-and-lookup {
      description
        "Operation pop incoming label and perform a lookup.";
    }
  }
  description
    "MPLS operations types.";
}
```

```
typedef nhlfe-role {
  type enumeration {
    enum primary {
      description
        "Next-hop acts as primary for carrying traffic.";
    }
    enum backup {
      description
        "Next-hop acts as backup.";
    }
    enum primary-and-backup {
      description
        "Next-hop acts as primary and backup simultaneously
        for carry traffic.";
    }
  }
  description
    "The next-hop role.";
}

grouping nhlfe-single-contents {
  description
    "A grouping that describes single Next Hop Label Forwarding
    Entry (NHLFE) and its associated parameters as described in
    the MPLS architecture. This grouping is specific to the case
    when a single next-hop is associated with the route.";
  uses rt-types:mpls-label-stack;
}

grouping nhlfe-multiple-contents {
  description
    "A grouping that describes a set of NHLFE(s) and their
    associated parameters as described in the MPLS architecture.
    This grouping is used when multiple next-hops are associated
    with the route.";
  leaf index {
    type string;
    description
      "A user-specified identifier utilised to uniquely
      reference the next-hop entry in the next-hop list.
      The value of this index has no semantic meaning
      other than for referencing the entry.";
  }
  leaf backup-index {
    type string;
    description
      "A user-specified identifier utilised to uniquely
      reference the backup next-hop entry in the NHLFE list."
  }
}
```

```
        The value of this index has no semantic meaning
        other than for referencing the entry.";
    reference
        "RFC4090 and RFC5714";
}
leaf loadshare {
    type uint16;
    default "1";
    description
        "This value is used to compute a loadshare to perform un-equal
        load balancing when multiple outgoing next-hop(s) are
        specified. A share is computed as a ratio of this number to the
        total under all next-hops(s).";
    reference
        "RFC7424, section 5.4,
        RFC3031, section 3.11 and 3.12.";
}
leaf role {
    type nhlfe-role;
    description
        "NHLFE role.";
}
uses nhlfe-single-contents;
}

grouping interfaces-mpls {
    description
        "List of MPLS interfaces.";
    container interfaces {
        description
            "List of MPLS enabled interaces.";
        list interface {
            key "name";
            description
                "MPLS enabled interface entry.";
            leaf name {
                type if:interface-ref;
                description
                    "A reference to the name of a interface in the system that
                    is to be enabled for MPLS.";
            }
            leaf mpls-enabled {
                type boolean;
                default "false";
                description
                    "'true' if mpls encapsulation is enabled on the interface.
                    'false' if mpls encapsulation is disabled on the
                    interface.";
            }
        }
    }
}
```

```
    }
    leaf maximum-labeled-packet {
        type uint32;
        units "octets";
        description
            "Maximum labeled packet size.";
        reference
            "RFC3032, section 3.2.";
    }
}
}
}

grouping globals {
    description
        "MPLS global configuration grouping.";
    leaf ttl-propagate {
        type boolean;
        default "true";
        description
            "Propagate TTL between IP and MPLS.";
    }
}

grouping label-blocks {
    description
        "Label-block allocation grouping.";
    container mpls-label-blocks {
        description
            "Label-block allocation container.";
        list mpls-label-block {
            key "index";
            description
                "List of MPLS label-blocks.";
            leaf index {
                type string;
                description
                    "A user-specified identifier utilised to uniquely
                     reference an MPLS label block.";
            }
            leaf start-label {
                type rt-types:mpls-label;
                must '. <= ../end-label' {
                    error-message
                        "The start-label must be less than or equal "
                        + "to end-label";
                }
                description

```

```
        "Label-block start.";
    }
    leaf end-label {
        type rt-types:mpls-label;
        must '.. >= ../start-label' {
            error-message
                "The end-label must be greater than or equal "
                + "to start-label";
        }
        description
            "Label-block end.";
    }
    leaf block-allocation-mode {
        type identityref {
            base label-block-alloc-mode;
        }
        description
            "Label-block allocation mode.";
    }
    leaf inuse-labels-count {
        when "derived-from-or-self(../block-allocation-mode, "
            + "'mpls:label-block-alloc-mode-manager')";
        type yang:gauge32;
        config false;
        description
            "Label-block inuse labels count.";
    }
}
}
}

grouping rib-mpls-properties {
    description
        "A grouping of native MPLS RIB properties.";
    leaf destination-prefix {
        type leafref {
            path "../mpls-local-label";
        }
        description
            "MPLS destination prefix.";
    }
    leaf route-context {
        type string;
        description
            "A context associated with the native MPLS route.";
    }
}
```

```
grouping rib-active-route-mpls-input {
  description
    "A grouping applicable to native MPLS RIB 'active-route'
    RPC input augmentation.";
  leaf destination-address {
    type leafref {
      path "../mpls-local-label";
    }
    description
      "MPLS native active route destination.";
  }
  leaf mpls-local-label {
    type rt-types:mpls-label;
    description
      "MPLS local label.";
  }
}

augment "/rt:routing" {
  description
    "MPLS augmentation.";
  container mpls {
    description
      "MPLS container, to be used as an augmentation target node
      other MPLS sub-features config, e.g. MPLS static LSP, MPLS
      LDP LSPs, and Traffic Engineering MPLS LSP Tunnels, etc.";
    uses globals;
    uses label-blocks;
    uses interfaces-mpls;
  }
}

/* MPLS routes augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route" {
  description
    "This augmentation is applicable to all MPLS routes.";
  leaf mpls-enabled {
    type boolean;
    default "false";
    description
      "Indicates whether MPLS is enabled for this route.";
  }
  leaf mpls-local-label {
    when "../mpls-enabled = 'true'";
    type rt-types:mpls-label;
    description
      "MPLS local label associated with the route.";
  }
}
```



```
    }
    uses rib-mpls-properties {
      /* MPLS AF augmentation to native MPLS RIB */
      when "derived-from-or-self(..../rt:address-family, "
        + "'mpls:mpls')" {
        description
          "This augment is valid only for routes of native MPLS
          RIB.";
      }
    }
  }
}

/* MPLS simple-next-hop augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
  description
    "Augment 'simple-next-hop' case in IP unicast routes.";
  uses nhlfe-single-contents {
    when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route"
      + "/mpls:mpls-enabled = 'true'";
  }
}

/* MPLS next-hop-list augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/"
  + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
  + "rt:next-hop-list/rt:next-hop" {
  description
    "This leaf augments the 'next-hop-list' case of IP unicast
    routes.";
  uses nhlfe-multiple-contents {
    when "/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route"
      + "/mpls:mpls-enabled = 'true'";
  }
}

/* MPLS RPC input augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input" {
  description
    "Input MPLS augmentation for the 'active-route' action
    statement.";
  uses rib-active-route-mpls-input {
    /* MPLS AF augmentation to native MPLS RIB */
    when "derived-from-or-self(..../rt:address-family, "
      + "'mpls:mpls')" {

```

```

        description
            "This augment is valid only for routes of native MPLS
            RIB.";
    }
}

/* MPLS RPC output augmentation */

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
    + "rt:output/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:simple-next-hop" {
    description
        "Output MPLS augmentation for the 'active-route' action
        statement.";
    uses nhlfe-single-contents;
}

augment "/rt:routing/rt:ribs/rt:rib/rt:active-route/"
    + "rt:output/rt:route/"
    + "rt:next-hop/rt:next-hop-options/rt:next-hop-list/"
    + "rt:next-hop-list/rt:next-hop" {
    description
        "Output MPLS augmentation for the 'active-route' action
        statement.";
    uses nhlfe-multiple-contents;
}
}
<CODE ENDS>

```

Figure 4: MPLS base YANG module.

3. IANA Considerations

This document registers the following URIs in the 'ns' sub-registry of the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls
 Registrant Contact: The MPLS WG of the IETF.
 XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
name:      ietf-mpls
namespace: urn:ietf:params:xml:ns:yang:ietf-mpls
prefix:    mpls
// RFC Ed.: replace XXXX with RFC number and remove this note
reference: RFCXXXX
```

4. Security Considerations

The YANG module specified in this document define a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/rt:routing/mpls:mpls/mpls:label-blocks"`: there are data nodes under this path that are writeable such as 'start-label' and 'end-label'. Write operations to those data nodes may cause disruptive action to existing traffic.

Some of the readable data nodes in these YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/rt:routing/rt:ribs/rt:rib/rt:routes/rt:route/rt:next-hop/rt:next-hop-options/rt:next-hop-list/rt:next-hop-list/rt:next-hop"` and `"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route/rt:next-hop/rt:next-hop-options/rt:simple-next-hop"`: these two paths are augmented by additional MPLS leaf(s) defined in this model. Access to this information may disclose the next-hop or path per prefix and/or other information.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:input"` and `"/rt:routing/rt:ribs/rt:rib/rt:active-route/rt:output/rt:route"`: these two paths are augmented by additional MPLS data node(s) that are defined in this model. Access to those path(s) may disclose information about per prefix route and/or other information and that may be further used for further attack(s).

The security considerations spelled out in [RFC3031] and [RFC3032] apply for this document as well.

5. Acknowledgement

The authors would like to thank Xia Chen for her contributions to the early revisions of this document.

6. Appendix A. Data Tree Instance Example

A simple network setup is shown in Figure 5. R1 runs the ISIS routing protocol, and learns reachability about two IPv4 prefixes: P1: 198.51.100.1/32 and P2: 198.51.100.1/32, and two IPv6 prefixes P3: 2001:db8:0:10::1/64 and P4: 2001:db8:0:10::1/64. We also assume that R1 learns about local and remote MPLS label bindings for each prefix using ISIS (e.g. using Segment-Routing (SR) extensions).

```

State on R1:
=====
      IPv4 Prefix          MPLS Label
P1: 198.51.100.1/32      16001
P2: 198.51.100.2/32      16002

      IPv6 Prefix          MPLS Label
P3: 2001:db8:0:10::1/64  16003
P4: 2001:db8:0:10::2/64  16004

RSVP MPLS LSPv4-Tunnel:
Source:      198.51.100.3
Destination: 198.51.100.4
Tunnel-ID:   10
LSP-ID:      1

                        192.0.2.5/30
                        2001:db8:0:1::1/64
                        eth0
                        +---+
                        /
          +-----+
          |  R1  |
          +-----+
            \
            +---+
            eth1
            192.0.2.13/30
            2001:db8:0:2::1/64

```

Figure 5: Example of network configuration.

The instance data tree could then be as follows:

```

{
  "ietf-routing:routing":{
    "ribs":{
      "rib":[
        {
          "name":"RIB-V4",
          "address-family":
            "ietf-ipv4-unicast-routing:v4ur:ipv4-unicast",
          "routes":{
            "route":[
              {
                "next-hop":{
                  "outgoing-interface":"eth0",
                  "ietf-mpls:mpls-label-stack":{

```

```
    "entry":[
      {
        "id":1,
        "label":16001,
        "ttl":255
      }
    ]
  },
  "ietf-ipv4-unicast-routing:next-hop-address":
    "192.0.2.5"
},
"source-protocol":"isis:isis",
"ietf-mpls:mpls-enabled":true,
"ietf-mpls:mpls-local-label":16001,
"ietf-ipv4-unicast-routing:destination-prefix":
"198.51.100.1/32",
"ietf-mpls:route-context":"SID-IDX:1"
},
{
  "next-hop":{
    "next-hop-list":{
      "next-hop":[
        {
          "outgoing-interface":"eth0",
          "ietf-mpls:index":"1",
          "ietf-mpls:backup-index":"2",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16002,
                "ttl":255
              }
            ]
          }
        },
        {
          "outgoing-interface":"eth1",
          "ietf-mpls:index":"2",
          "ietf-mpls:backup-index":"1",
          "ietf-mpls:role":"primary-and-backup",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16002,
```

```

        "ttl":255
      }
    ]
  },
  "ietf-ipv4-unicast-routing:address":"192.0.2.13"
}
]
}
},
"source-protocol":"isis:isis",
"ietf-mpls:mpls-enabled":true,
"ietf-mpls:mpls-local-label":16002,
"ietf-ipv4-unicast-routing:destination-prefix":
"198.51.100.2/32",
"ietf-mpls:route-context":"SID-IDX:2"
}
]
}
},
{
  "name":"RIB-V6",
  "address-family":
  "ietf-ipv6-unicast-routing:v6ur:ipv6-unicast",
  "routes":{
    "route":[
      {
        "next-hop":{
          "outgoing-interface":"eth0",
          "ietf-mpls:mpls-label-stack":{
            "entry":[
              {
                "id":1,
                "label":16003,
                "ttl":255
              }
            ]
          },
          "ietf-ipv6-unicast-routing:next-hop-address":
            "2001:db8:0:1::1"
        },
        "source-protocol":"isis:isis",
        "ietf-mpls:mpls-enabled":true,
        "ietf-mpls:mpls-local-label":16001,
        "ietf-ipv6-unicast-routing:destination-prefix":
        "2001:db8:0:10::1/6",
        "ietf-mpls:route-context":"SID-IDX:1"
      },
      {

```

```

    "next-hop":{
      "next-hop-list":{
        "next-hop":[
          {
            "outgoing-interface":"eth0",
            "ietf-mpls:index":"1",
            "ietf-mpls:backup-index":"2",
            "ietf-mpls:role":"primary-and-backup",
            "ietf-mpls:mpls-label-stack":{
              "entry":[
                {
                  "id":1,
                  "label":16004,
                  "ttl":255
                }
              ]
            },
            "ietf-ipv6-unicast-routing:address":
              "2001:db8:0:1::1"
          },
          {
            "outgoing-interface":"eth1",
            "ietf-mpls:index":"2",
            "ietf-mpls:backup-index":"1",
            "ietf-mpls:role":"primary-and-backup",
            "ietf-mpls:mpls-label-stack":{
              "entry":[
                {
                  "id":1,
                  "label":16004,
                  "ttl":255
                }
              ]
            },
            "ietf-ipv6-unicast-routing:address":
              "2001:db8:0:2::1"
          }
        ]
      }
    },
    "source-protocol":"isis:isis",
    "ietf-mpls:mpls-enabled":true,
    "ietf-mpls:mpls-local-label":16004,
    "ietf-ipv6-unicast-routing:destination-prefix":
      "2001:db8:0:10::2/64",
    "ietf-mpls:route-context":"SID-IDX:2"
  }
]

```



```

    }
  },
  {
    "name": "RIB-MPLS",
    "address-family": "ietf-mpls:mpls:mpls",
    "routes": {
      "route": [
        {
          "next-hop": {
            "outgoing-interface": "eth0",
            "ietf-mpls:mpls-label-stack": {
              "entry": [
                {
                  "id": 1,
                  "label": 24002,
                  "ttl": 255
                }
              ]
            }
          },
          "ietf-ipv4-unicast-routing:next-hop-address":
            "192.0.2.5"
        },
        "source-protocol": "rsvp:rsvp",
        "ietf-mpls:mpls-enabled": true,
        "ietf-mpls:mpls-local-label": 24001,
        "ietf-mpls:destination-prefix": "24001",
        "ietf-mpls:route-context":
          "RSVP Src:198.51.100.3,Dst:198.51.100.4,T:10,L:1"
      ]
    }
  }
}

]
},
"ietf-mpls:mpls": {
  "mpls-label-blocks": {
    "mpls-label-block": [
      {
        "index": "mpls-srgb-label-block",
        "start-label": 16000,
        "end-label": 16500,
        "block-allocation-mode": "mpls:label-block-alloc-mode-manager"
      }
    ]
  }
},
"interfaces": {
  "interface": [
    {

```

```
        "name":"eth0",
        "mpls-enabled":true,
        "maximum-labeled-packet":1488
      },
      {
        "name":"eth1",
        "mpls-enabled":true,
        "maximum-labeled-packet":1488
      }
    ]
  }
}
```

Figure 6: Foo bar.

7. Contributors

Igor Bryskin
Huawei Technologies
email: i_bryskin@yahoo.com

Himanshu Shah
Ciena
email: hshah@ciena.com

8. References

8.1. Normative References

- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

8.2. Informative References

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC7424] Krishnan, R., Yong, L., Ghanwani, A., So, N., and B. Khasnabish, "Mechanisms for Optimizing Link Aggregation Group (LAG) and Equal-Cost Multipath (ECMP) Component Link Utilization in Networks", RFC 7424, DOI 10.17487/RFC7424, January 2015, <<https://www.rfc-editor.org/info/rfc7424>>.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Kamran Raza
Cisco Systems Inc

Email: skraza@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 19, 2017

K. Raza
R. Asati
Cisco Systems, Inc.

X. Liu
Ericsson

S. Esale
Juniper Networks

X. Chen
Huawei Technologies

H. Shah
Ciena Corporation

August 18, 2016

YANG Data Model for MPLS LDP and mLDP
draft-ietf-mpls-ldp-mldp-yang-00

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP) and Multipoint LDP (mLDP).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 19, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Specification of Requirements	3
3. LDP YANG Model	3
3.1. Overview	4
3.2. Configuration	7
3.2.1. Configuration Hierarchy	11
3.2.2. All-VRFs Configuration	14
3.3. Operational State	14
3.3.1. Derived States	21
3.4. Notifications	26
3.5. Actions	26
4. mLDP YANG Model	27
4.1. Overview	27
4.2. Configuration	28
4.2.1. Configuration Hierarchy	28
4.2.2. mldp container	30
4.2.3. Leveraging LDP containers	31
4.2.4. YANG tree	31
4.3. Operational State	33
4.3.1. Derived states	38
4.4. Notifications	42
4.5. Actions	43
5. Open Items	43
6. YANG Specification	43
7. Security Considerations	110
8. IANA Considerations	110
9. Acknowledgments	110
10. References	110
10.1. Normative References	110
10.2. Informative References	113
Appendix A. Additional Contributors	113

Authors' Addresses	113
--------------------	-----

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is one of the network management protocols that defines mechanisms to manage network devices. YANG [RFC6020] is a modular language that represents data structures in an XML tree format, and is used as a data modelling language for the NETCONF.

This document introduces a YANG data model for MPLS Label Distribution Protocol (LDP) [RFC5036] and Multipoint LDP (mLDP) [RFC6388]. For LDP, it also covers LDP IPv6 [RFC7552] and LDP capabilities [RFC5561].

The data model is defined for following constructs that are used for managing the protocol:

- o Configuration
- o Operational State
- o Executables (Actions)
- o Notifications

This document is organized to define the data model for each of the above constructs (configuration, state, action, and notifications) in the sequence as listed earlier. Given that mLDP is tightly coupled with LDP, mLDP data model is defined under LDP tree and in the same sequence as listed above.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In this document, the word "IP" is used to refer to both IPv4 and IPv6, unless otherwise explicitly stated. For example, "IP address family" means and be read as "IPv4 and/or IPv6 address family"

3. LDP YANG Model

3.1. Overview

This document defines a new module named "ietf-mpls-ldp" for LDP/mLDP data model where this module augments /rt:routing/rt:control-plane-protocols that is defined in [I-D.ietf-netmod-routing-cfg].

There are four main containers in "ietf-mpls-ldp" module as follows:

- o Read-Write parameters for configuration (Discussed in Section 3.2)
- o Read-only parameters for operational state (Discussed in Section 3.3)
- o Notifications for events (Discussed in Section 3.4)
- o RPCs for executing commands to perform some action (Discussed in Section 3.5)

For the configuration and state data, this model follows the similar approach described in [I-D.openconfig-netmod-opstate] to represent the configuration (intended state) and operational (applied and derived) state. This means that for every configuration (rw) item, there is an associated (ro) item under "state" container to represent the applied state. Furthermore, protocol derived state is also kept under "state" tree corresponding to the protocol area (discovery, peer etc.). [Ed note: This document will be (re-)aligned with [I-D.openconfig-netmod-opstate] once that specification is adopted as a WG document]

Following diagram depicts high level LDP yang tree organization and hierarchy:

```

module: ietf-mpls-ldp
  +-- rw routing
    +-- rw control-plane-protocols
      +-- rw mpls-ldp
        +-- rw global
          +-- rw config
            +-- rw ...
          +-- ro state
            +-- ro ...
          .
        +-- rw ...
      +-- rw ...
    ...

rpcs:
  +-- x mpls-ldp-rpc
  +-- x . . . . .

notifications:
  +--- n mpls-ldp-notif
  +--- n ...

```

Figure 1

Before going into data model details, it is important to take note of the following points:

- o This module aims to address only the core LDP/mLDP parameters as per RFC specification, as well as some widely used and deployed non-RFC features (such as label policies, session authentication etc). Any vendor specific feature should be defined in a vendor-specific augmentation of this model.
- o Multi-topology LDP [RFC7307] and Multi-topology mLDP [I-D.iwijnand-mpls-mldp-multi-topology] are beyond the scope of this document.
- o This module does not cover any applications running on top of LDP and mLDP, nor does it cover any OAM procedures for LDP and mLDP.
- o This model is a VPN Forwarding and Routing (VRF)-centric model. It is important to note that [RFC4364] defines VRF tables and default forwarding tables as different, however from a yang modelling perspective this introduces unnecessary complications,

hence we are treating the default forwarding table as just another VRF.

- o A "network-instance" as defined in [I-D.rtgyangdt-rtgwg-ni-model] refers to a VRF instance (both default and non-default) within the scope of this model.
- o This model supports two address-families, namely "ipv4" and "ipv6".
- o This model assumes platform-wide label space (i.e. label space Id of zero). However, when Upstream Label assignment [RFC6389] is in use, an upstream assigned label is looked up in a Context-Specific label space as defined in [RFC5331].
- o The label and peer policies (including filters) are defined using a prefix-list. When used for a peer policy, the prefix refers to the LSR Id of the peer. The prefix-list is referenced from routing-policy model as defined in [I-D.ietf-rtgwg-policy-model].
- o The use of grouping (templates) for bundling and grouping the configuration items is not employed in current revision, and is a subject for consideration in future.
- o This model uses the terms LDP "neighbor"/"adjacency", "session", and "peer" with the following semantics:
 - * Neighbor/Adjacency: An LDP enabled LSR that is discovered through LDP discovery mechanisms.
 - * Session: An LDP neighbor with whom a TCP connection has been established.
 - * Peer: An LDP session which has successfully progressed beyond its initialization phase and is either already exchanging the bindings or is ready to do so.

It is to be noted that LDP Graceful Restart mechanisms defined in [RFC3478] allow keeping the exchanged bindings for some time after a session goes down with a peer. We call such a state -- i.e. keeping peer bindings without established or recovered peering -- a "stale" peer. When used in this document, the above terms will refer strictly to the semantics and definitions defined for them.

A graphical representation of LDP YANG data model is presented in Figure 3, Figure 5, Figure 11, and Figure 12. Whereas, the actual model definition in YANG is captured in Section 6.

While presenting the YANG tree view and actual .yang specification, this document assumes the reader is familiar with the concepts of YANG modeling, its presentation and its compilation.

3.2. Configuration

This specification defines the configuration parameters for base LDP as specified in [RFC5036] and LDP IPv6 [RFC7552]. Moreover, it incorporates provisions to enable LDP Capabilities [RFC5561], and defines some of the most significant and commonly used capabilities such as Typed Wildcard FEC [RFC5918], End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

This specification supports VRF-centric configuration. For implementations that support protocol-centric configuration, with provision for inheritance and items that apply to all vrfs, we recommend an augmentation of this model such that any protocol-centric or all-vrf configuration is defined under their designated containers within the standard network-instance (please see Section 3.2.2)

This model augments /rt:routing/rt:control-plane-protocols that is defined in [I-D.ietf-netmod-routing-cfg]. For LDP interfaces, this model refers the MPLS interface as defined under MPLS base specification [I-D.saad-mpls-base-yang]. Furthermore, as mentioned earlier, the configuration tree presents read-write intended configuration leave/items as well as read-only state of the applied configuration. The former is listed under "config" container and latter under "state" container.

Following is high-level configuration organization for LDP/mLDP:

```

module: ietf-mpls-ldp
  +-- routing
  +-- control-plane-protocols
    +-- mpls-ldp
      +-- global
        +-- ...
        +-- ...
        +-- address-family* [afi]
          +-- . . .
          +-- . . .
        +-- discovery
          +-- . . .
      +-- peers
        +-- ...
        +-- ...

```

Figure 2

Given the configuration hierarchy, the model allows inheritance such that an item in a child tree is able to derive value from a similar or related item in one of the parent. For instance, hello holdtime can be configured per-VRF or per-VRF-interface, thus allowing inheritance as well flexibility to override with a different value at any child level.

Following is a simplified graphical representation of the data model for LDP configuration

```

+--rw mpls-ldp!
  +--rw global
    |
    | +--rw config
    | |
    | | +--rw capability
    | | |
    | | | +--rw end-of-lib {capability-end-of-lib}?
    | | | | +--rw enable?    boolean
    | | | +--rw typed-wildcard-fec {capability-typed-wildcard-fec}?
    | | | | +--rw enable?    boolean
    | | | +--rw upstream-label-assignment {capability-upstream-label-assign
ment}?
    | | | | +--rw enable?    boolean
    | | | +--rw graceful-restart
    | | | | +--rw enable?          boolean
    | | | | +--rw helper-enable?   boolean {graceful-restart-helper-mod
e}?
    | | | +--rw reconnect-time?    uint16
    | | | +--rw recovery-time?     uint16
    | | | +--rw forwarding-holdtime? uint16
    | | +--rw igp-synchronization-delay? uint16
    | +--rw lsr-id?                yang:dotted-quad

```

```

|   +--rw address-family* [afi]
|   |   +--rw afi          ldp-address-family
|   |   +--rw config
|   |   |   +--rw enable?          boolean
|   |   |   +--rw label-policy
|   |   |   |   +--rw independent-mode
|   |   |   |   |   +--rw assign {policy-label-assignment-config}?
|   |   |   |   |   |   +--rw (prefix-option)?
|   |   |   |   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   |   |   |   |   |   +--rw host-routes-only?      boolean
|   |   |   |   +--rw advertise
|   |   |   |   |   +--rw explicit-null
|   |   |   |   |   |   +--rw enable?          boolean
|   |   |   |   |   |   +--rw prefix-list?      prefix-list-ref
|   |   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   |   +--rw accept
|   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   +--rw ordered-mode {policy-ordered-label-config}?
|   |   |   +--rw egress-lsr
|   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   |   +--rw advertise
|   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   |   +--rw accept
|   |   |   |   +--rw prefix-list?          prefix-list-ref
|   |   +--rw ipv4
|   |   |   +--rw transport-address?          inet:ipv4-address
|   |   +--rw ipv6
|   |   |   +--rw transport-address?          inet:ipv6-address
|   +--rw discovery
|   |   +--rw interfaces
|   |   |   +--rw config
|   |   |   |   +--rw hello-holdtime?          uint16
|   |   |   |   +--rw hello-interval?          uint16
|   |   |   +--rw interface* [interface]
|   |   |   |   +--rw interface                  mpls-interface-ref
|   |   |   |   +--rw config
|   |   |   |   |   +--rw hello-holdtime?          uint16
|   |   |   |   |   +--rw hello-interval?          uint16
|   |   |   |   |   +--rw igp-synchronization-delay? uint16 {per-interface-ti
mer-config}?
|   |   |   +--rw address-family* [afi]
|   |   |   |   +--rw afi          ldp-address-family
|   |   |   |   +--rw config
|   |   |   |   |   +--rw enable?          boolean
|   |   |   |   |   +--rw ipv4
|   |   |   |   |   |   +--rw transport-address?          union
|   |   |   |   |   +--rw ipv6
|   |   |   |   |   |   +--rw transport-address?          union
|   |   +--rw targeted

```

```

+--rw config
|   +--rw hello-holdtime?    uint16
|   +--rw hello-interval?    uint16
|   +--rw hello-accept {policy-extended-discovery-config}?
|       +--rw enable?        boolean
|       +--rw neighbor-list? neighbor-list-ref
+--rw address-family* [afi]
|   +--rw afi                ldp-address-family
|   +--rw ipv4
|       +--rw target* [adjacent-address]
|           +--rw adjacent-address    inet:ipv4-address
|           +--rw config
|               +--rw enable?        boolean
|               +--rw local-address?  inet:ipv4-address
|   +--rw ipv6
|       +--rw target* [adjacent-address]
|           +--rw adjacent-address    inet:ipv6-address
|           +--rw config
|               +--rw enable?        boolean
|               +--rw local-address?  inet:ipv6-address
+--rw forwarding-nexthop {forwarding-nexthop-config}?
|   +--rw interfaces
|       +--rw interface* [interface]
|           +--rw interface          mpls-interface-ref
|           +--rw address-family* [afi]
|               +--rw afi            ldp-address-family
|               +--rw config
|                   +--rw ldp-disable? boolean
+--rw label-policy
|   +--rw independent-mode
|       +--rw assign {policy-label-assignment-config}?
|           +--rw (prefix-option)?
|               +--rw prefix-list?    prefix-list-ref
|               +--rw host-routes-only? boolean
|       +--rw advertise
|           +--rw explicit-null
|               +--rw enable?        boolean
|               +--rw prefix-list?    prefix-list-ref
|           +--rw prefix-list?        prefix-list-ref
|       +--rw accept
|           +--rw prefix-list?        prefix-list-ref
+--rw ordered-mode {policy-ordered-label-config}?
|   +--rw egress-lsr
|       +--rw prefix-list?    prefix-list-ref
|   +--rw advertise
|       +--rw prefix-list?    prefix-list-ref
|   +--rw accept
|       +--rw prefix-list?    prefix-list-ref

```

```

+--rw peers
  +--rw config
    |   +--rw session-authentication-md5-password?  string
    |   +--rw session-ka-holdtime?                  uint16
    |   +--rw session-ka-interval?                  uint16
    |   +--rw session-downstream-on-demand {session-downstream-on-demand-con
fig}?
    |   |   +--rw enable?          boolean
    |   |   +--rw peer-list?      peer-list-ref
    +--rw peer* [lsr-id]
      +--rw lsr-id      yang:dotted-quad
      +--rw config
        +--rw admin-down?          boolean
        +--rw capability
        +--rw label-policy
          |   +--rw advertise
          |   |   +--rw prefix-list?  prefix-list-ref
          |   +--rw accept
          |   |   +--rw prefix-list?  prefix-list-ref
        +--rw session-authentication-md5-password?  string
        +--rw graceful-restart
          |   +--rw enable?          boolean
          |   +--rw reconnect-time?  uint16
          |   +--rw recovery-time?   uint16
        +--rw session-ka-holdtime?          uint16
        +--rw session-ka-interval?          uint16
        +--rw address-family
          +--rw ipv4
            |   +--rw label-policy
            |   |   +--rw advertise
            |   |   |   +--rw prefix-list?  prefix-list-ref
            |   |   +--rw accept
            |   |   |   +--rw prefix-list?  prefix-list-ref
          +--rw ipv6
            +--rw label-policy
              +--rw advertise
              |   +--rw prefix-list?  prefix-list-ref
              +--rw accept
              |   +--rw prefix-list?  prefix-list-ref

```

Figure 3

3.2.1. Configuration Hierarchy

The LDP configuration container is logically divided into following high-level config areas:


```
Per-VRF parameters
  o Global parameters
  o Per-address-family parameters
  o LDP Capabilities parameters
  o Hello Discovery parameters
    - interfaces
      - Per-interface:
        Global
        Per-address-family
    - targeted
      - Per-target
  o Peer parameters
    - Global
    - Per-peer
      Per-address-family
      Capabilities parameters
  o Forwarding parameters
```

Figure 4

Following subsections briefly explain these configuration areas.

3.2.1.1. Per-VRF parameters

LDP module resides under an network-instance and the scope of any LDP configuration defined under this tree is per network-instance (per-VRF). This configuration is further divided into sub categories as follows.

3.2.1.1.1. Per-VRF global parameters

There are configuration items that are available directly under a VRF instance and do not fall under any other sub tree. Example of such a parameter is LDP LSR id that is typically configured per VRF. To keep legacy LDP features and applications working in an LDP IPv4 networks with this model, this document recommends an operator to pick a routable IPv4 unicast address as an LSR Id.

3.2.1.1.2. Per-VRF Capabilities parameters

This container falls under global tree and holds the LDP capabilities that are to be enabled for certain features. By default, an LDP capability is disabled unless explicitly enabled. These capabilities are typically used to negotiate with LDP peer(s) the support/non-support related to a feature and its parameters. The scope of a capability enabled under this container applies to all LDP peers in the given VRF instance. There is also a peer level capability

container that is provided to override a capability that is enabled/specified at VRF level.

3.2.1.1.3. Per-VRF Per-Address-Family parameters

Any LDP configuration parameter related to IP address family (AF) whose scope is VRF wide is configured under this tree. The examples of per-AF parameters include enabling LDP for an address family, prefix-list based label policies, and LDP transport address.

3.2.1.1.4. Per-VRF Hello Discovery parameters

This container is used to hold LDP configuration related to Hello and discovery process for both basic (link) and extended (targeted) discovery.

The "interfaces" is a container to configure parameters related to VRF interfaces. There are parameters that apply to all interfaces (such as hello timers), as well as parameters that can be configured per-interface. Hence, an interface list is defined under "interfaces" container. The model defines parameters to configure per-interface non AF related items, as well as per-interface per-AF items. The example of former is interface hello timers, and example of latter is enabling hellos for a given AF under an interface.

The "targeted" container under a VRF instance allows to configure LDP targeted discovery related parameters. Within this container, the "target" list provides a mean to configure multiple target addresses to perform extended discovery to a specific destination target, as well as to fine-tune the per-target parameters.

3.2.1.1.5. Per-VRF Peer parameters

This container is used to hold LDP configuration related to LDP sessions and peers under a VRF instance. This container allows to configure parameters that either apply on VRF's all peers or a subset (peer-list) of VRF peers. The example of such parameters include authentication password, session KA timers etc. Moreover, the model also allows per-peer parameter tuning by specifying a "peer" list under the "peers" container. A peer is uniquely identified using its LSR Id and hence LSR Id is the key for peer list

Like per-interface parameters, some per-peer parameters are AF-agnostic (i.e. either non AF related or apply to both IP address families), and some that belong to an AF. The example of former is per-peer session password configuration, whereas the example of latter is prefix-list based label policies (inbound and outbound) that apply to a given peer.

3.2.1.1.6. Per-VRF Forwarding parameters

This container is used to hold configuration used to control LDP forwarding behavior under a VRF instance. One example of a configuration under this container is when a user wishes to enable neighbor discovery on an interface but wishes to disable use of the same interface as forwarding nexthop. This example configuration makes sense only when there are more than one LDP enabled interfaces towards the neighbor.

3.2.2. All-VRFs Configuration

[Ed note: TODO]

3.3. Operational State

Operational state of LDP can be queried and obtained from read-only state containers that fall under the same tree (/rt:routing/rt:control-plane-protocols/) as the configuration.

Please note this state tree refers both the configuration "applied" state as well as the "derived" state related to the protocol. [Ed note: This is where this model differs presently from [I-D.openconfig-netmod-opstate] and subject to alignment in later revisions]

Following is a simplified graphical representation of the data model for LDP operational state.

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--ro state
        +--ro capability
          +--ro end-of-lib {capability-end-of-lib}?
            | +--ro enable?    boolean
          +--ro typed-wildcard-fec {capability-typed-wildcard-fec}?
            | +--ro enable?    boolean
          +--ro upstream-label-assignment {capability-upstream-label-assignment}?
            | +--ro enable?    boolean
          +--ro graceful-restart
            +--ro enable?      boolean
            +--ro helper-enable? boolean {graceful-restart-helper-mod
e}?
          +--ro reconnect-time?    uint16
          +--ro recovery-time?     uint16
          +--ro forwarding-holdtime? uint16

```

```

+---ro igp-synchronization-delay?      uint16
+---ro lsr-id?                         yang:dotted-quad
+---rw address-family* [afi]
+---rw afi                             ldp-address-family
+---ro state
+---ro enable?                         boolean
+---ro label-policy
+---ro independent-mode
+---ro assign {policy-label-assignment-config}?
+---ro (prefix-option)?
+---:(prefix-list)
+---ro prefix-list?                    prefix-list-ref
+---:(host-routes-only)
+---ro host-routes-only?              boolean
+---ro advertise
+---ro explicit-null
+---ro enable?                         boolean
+---ro prefix-list?                    prefix-list-ref
+---ro prefix-list?                    prefix-list-ref
+---ro accept
+---ro prefix-list?                    prefix-list-ref
+---ro ordered-mode {policy-ordered-label-config}?
+---ro egress-lsr
+---ro prefix-list?                    prefix-list-ref
+---ro advertise
+---ro prefix-list?                    prefix-list-ref
+---ro accept
+---ro prefix-list?                    prefix-list-ref
+---ro ipv4
+---ro transport-address?              inet:ipv4-address
+---ro bindings
+---ro address* [address]
+---ro address                         inet:ipv4-address
+---ro advertisement-type?             advertised-received
+---ro peer?                           leafref
+---ro fec-label* [fec]
+---ro fec                             inet:ipv4-prefix
+---ro peer* [peer advertisement-type]
+---ro peer                             leafref
+---ro advertisement-type              advertised-received
+---ro label?                          mpls:mpls-label
+---ro used-in-forwarding?             boolean
+---ro ipv6
+---ro transport-address?              inet:ipv6-address
+---ro binding
+---ro address* [address]
+---ro address                         inet:ipv6-address
+---ro advertisement-type?             advertised-received

```

```

|   |--ro peer? leafref
|--ro fec-label* [fec]
|   |--ro fec inet:ipv6-prefix
|--ro peer* [peer advertisement-type]
|   |--ro peer leafref
|   |--ro advertisement-type advertised-received
|   |--ro label? mpls:mpls-label
|   |--ro used-in-forwarding? boolean
|--rw discovery
|   |--rw interfaces
|       |--ro state
|           |--ro hello-holdtime? uint16
|           |--ro hello-interval? uint16
|--rw interface* [interface]
|   |--ro state
|       |--ro hello-holdtime? uint16
|       |--ro hello-interval? uint16
|       |--ro igp-synchronization-delay? uint16 {per-interface-ti
mer-config}?
|           |--ro next-hello? uint16
|--rw address-family* [afi]
|   |--rw afi ldp-address-family
|       |--ro state
|           |--ro enable? boolean
|           |--ro ipv4
|               |--ro transport-address? union
|               |--ro hello-adjacencies* [adjacent-address]
|                   |--ro adjacent-address inet:ipv4-address
|                   |--ro flag* identityref
|                   |--ro hello-holdtime
|                       |--ro adjacent? uint16
|                       |--ro negotiated? uint16
|                       |--ro remaining? uint16
|                   |--ro next-hello? uint16
|               |--ro statistics
|                   |--ro discontinuity-time yang:date-and-time
|                   |--ro hello-received? yang:counter64
|                   |--ro hello-dropped? yang:counter64
|               |--ro peer? leafref
|--ro ipv6
|   |--ro transport-address? union
|   |--ro hello-adjacencies* [adjacent-address]
|       |--ro adjacent-address inet:ipv6-address
|       |--ro flag* identityref
|       |--ro hello-holdtime
|           |--ro adjacent? uint16
|           |--ro negotiated? uint16
|           |--ro remaining? uint16
|       |--ro next-hello? uint16

```

```

    +--ro statistics
      |   |--ro discontinuity-time       yang:date-and-time
      |   |--ro hello-received?         yang:counter64
      |   |--ro hello-dropped?          yang:counter64
      |   |--ro peer?                    leafref
+--rw targeted
  +--ro state
    |   |--ro hello-holdtime?           uint16
    |   |--ro hello-interval?          uint16
    |   |--ro hello-accept {policy-extended-discovery-config}?
    |     |--ro enable?                  boolean
    |     |--ro neighbor-list?          neighbor-list-ref
+--rw address-family* [afi]
  +--rw afi                ldp-address-family
  +--ro state
    |   |--ro ipv4
    |     |--ro hello-adjacencies* [local-address adjacent-address
    ]
        |--ro local-address            inet:ipv4-address
        |--ro adjacent-address          inet:ipv4-address
        |--ro flag*                     identityref
        |--ro hello-holdtime
          |   |--ro adjacent?            uint16
          |   |--ro negotiated?          uint16
          |   |--ro remaining?           uint16
        |--ro next-hello?               uint16
        |--ro statistics
          |   |--ro discontinuity-time     yang:date-and-time
          |   |--ro hello-received?        yang:counter64
          |   |--ro hello-dropped?         yang:counter64
          |   |--ro peer?                 leafref
        +--ro ipv6
          |--ro hello-adjacencies* [local-address adjacent-address
          ]
              |--ro local-address          inet:ipv6-address
              |--ro adjacent-address        inet:ipv6-address
              |--ro flag*                   identityref
              |--ro hello-holdtime
                |   |--ro adjacent?         uint16
                |   |--ro negotiated?       uint16
                |   |--ro remaining?        uint16
              |--ro next-hello?             uint16
              |--ro statistics
                |   |--ro discontinuity-time   yang:date-and-time
                |   |--ro hello-received?     yang:counter64
                |   |--ro hello-dropped?      yang:counter64
                |   |--ro peer?              leafref
              +--rw ipv4
                |--rw target* [adjacent-address]
                |--rw adjacent-address      inet:ipv4-address

```

```

|         |         |         +---ro state                boolean
|         |         |         +---ro enable?              boolean
|         |         |         +---ro local-address?       inet:ipv4-address
+---rw ipv6
|         |         |         +---rw target* [adjacent-address]
|         |         |         +---rw adjacent-address     inet:ipv6-address
|         |         |         +---ro state
|         |         |         +---ro enable?              boolean
|         |         |         +---ro local-address?       inet:ipv6-address
+---rw forwarding-nexthop {forwarding-nexthop-config}?
|         |         |         +---rw interfaces
|         |         |         |         +---rw interface* [interface]
|         |         |         |         +---rw interface   mpls-interface-ref
|         |         |         |         +---rw address-family* [afi]
|         |         |         |         +---rw afi          ldp-address-family
|         |         |         |         +---ro state
|         |         |         |         +---ro ldp-disable?    boolean
+---rw peers
|         |         |         +---ro state
|         |         |         |         +---ro session-authentication-md5-password? string
|         |         |         |         +---ro session-ka-holdtime?           uint16
|         |         |         |         +---ro session-ka-interval?           uint16
|         |         |         |         +---ro session-downstream-on-demand {session-downstream-on-demand-con
fig)?
|         |         |         |         +---ro enable?            boolean
|         |         |         |         +---ro peer-list?        peer-list-ref
+---rw peer* [lsr-id]
|         |         |         |         +---rw lsr-id      yang:dotted-quad
|         |         |         |         +---ro state
|         |         |         |         |         +---ro admin-down?                    boolean
|         |         |         |         |         +---ro capability
|         |         |         |         |         +---ro label-policy
|         |         |         |         |         |         +---ro advertise
|         |         |         |         |         |         |         +---ro prefix-list?    prefix-list-ref
|         |         |         |         |         |         +---ro accept
|         |         |         |         |         |         |         +---ro prefix-list?    prefix-list-ref
+---ro session-authentication-md5-password? string
+---ro graceful-restart
|         |         |         |         |         +---ro enable?            boolean
|         |         |         |         |         +---ro reconnect-time?    uint16
|         |         |         |         |         +---ro recovery-time?     uint16
+---ro session-ka-holdtime?                uint16
+---ro session-ka-interval?                uint16
+---ro address-family
|         |         |         |         +---ro ipv4
|         |         |         |         |         +---ro label-policy
|         |         |         |         |         |         +---ro advertise
|         |         |         |         |         |         |         +---ro prefix-list?    prefix-list-ref
|         |         |         |         |         |         +---ro accept

```

```

|         +--ro prefix-list?    prefix-list-ref
+--ro hello-adjacencies* [local-address adjacent-address]
|   +--ro local-address          inet:ipv4-address
|   +--ro adjacent-address       inet:ipv4-address
|   +--ro flag*                  identityref
|   +--ro hello-holdtime
|   |   +--ro adjacent?          uint16
|   |   +--ro negotiated?       uint16
|   |   +--ro remaining?        uint16
|   +--ro next-hello?            uint16
|   +--ro statistics
|   |   +--ro discontinuity-time  yang:date-and-time
|   |   +--ro hello-received?    yang:counter64
|   |   +--ro hello-dropped?     yang:counter64
|   +--ro interface?            mpls-interface-ref
+--ro ipv6
|   +--ro label-policy
|   |   +--ro advertise
|   |   |   +--ro prefix-list?    prefix-list-ref
|   |   +--ro accept
|   |   |   +--ro prefix-list?    prefix-list-ref
|   +--ro hello-adjacencies* [local-address adjacent-address]
|   |   +--ro local-address       inet:ipv6-address
|   |   +--ro adjacent-address    inet:ipv6-address
|   |   +--ro flag*               identityref
|   |   +--ro hello-holdtime
|   |   |   +--ro adjacent?        uint16
|   |   |   +--ro negotiated?      uint16
|   |   |   +--ro remaining?       uint16
|   |   +--ro next-hello?         uint16
|   |   +--ro statistics
|   |   |   +--ro discontinuity-time  yang:date-and-time
|   |   |   +--ro hello-received?    yang:counter64
|   |   |   +--ro hello-dropped?     yang:counter64
|   |   +--ro interface?          mpls-interface-ref
+--ro label-advertisement-mode
|   +--ro local?                  label-adv-mode
|   +--ro peer?                   label-adv-mode
|   +--ro negotiated?             label-adv-mode
+--ro next-keep-alive?            uint16
+--ro peer-ldp-id?                yang:dotted-quad
+--ro received-peer-state
|   +--ro graceful-restart
|   |   +--ro enable?             boolean
|   |   +--ro reconnect-time?     uint16
|   |   +--ro recovery-time?      uint16
+--ro capability
+--ro end-of-lib

```



```

|      |  +--ro enable?    boolean
|      +--ro typed-wildcard-fec
|      |  +--ro enable?    boolean
|      +--ro upstream-label-assignment
|      +--ro enable?    boolean
+--ro session-holdtime
|  +--ro peer?          uint16
|  +--ro negotiated?    uint16
|  +--ro remaining?     uint16
+--ro session-state?          enumeration
+--ro tcp-connection
|  +--ro local-address?      inet:ip-address
|  +--ro local-port?         inet:port-number
|  +--ro remote-address?     inet:ip-address
|  +--ro remote-port?        inet:port-number
+--ro up-time?              string
+--ro statistics
|  +--ro discontinuity-time    yang:date-and-time
|  +--ro received
|  |  +--ro total-octets?      yang:counter64
|  |  +--ro total-messages?   yang:counter64
|  |  +--ro address?          yang:counter64
|  |  +--ro address-withdraw? yang:counter64
|  |  +--ro initialization?    yang:counter64
|  |  +--ro keepalive?        yang:counter64
|  |  +--ro label-abort-request? yang:counter64
|  |  +--ro label-mapping?     yang:counter64
|  |  +--ro label-release?     yang:counter64
|  |  +--ro label-request?     yang:counter64
|  |  +--ro label-withdraw?    yang:counter64
|  |  +--ro notification?      yang:counter64
|  +--ro sent
|  |  +--ro total-octets?      yang:counter64
|  |  +--ro total-messages?   yang:counter64
|  |  +--ro address?          yang:counter64
|  |  +--ro address-withdraw? yang:counter64
|  |  +--ro initialization?    yang:counter64
|  |  +--ro keepalive?        yang:counter64
|  |  +--ro label-abort-request? yang:counter64
|  |  +--ro label-mapping?     yang:counter64
|  |  +--ro label-release?     yang:counter64
|  |  +--ro label-request?     yang:counter64
|  |  +--ro label-withdraw?    yang:counter64
|  |  +--ro notification?      yang:counter64
+--ro total-addresses?      uint32
+--ro total-labels?         uint32
+--ro total-fec-label-bindings? uint32

```

Figure 5

3.3.1. Derived States

Following are main areas for which LDP operational "derived" state is defined:

- Neighbor Adjacencies

- Peer

- Bindings (FEC-label and address)

- Capabilities

3.3.1.1. Adjacency state

Neighbor adjacencies are per address-family hello adjacencies that are formed with neighbors as result of LDP basic or extended discovery. In terms of organization, there is a source of discovery (e.g. interface or target address) along with its associated parameters and one or more discovered neighbors along with neighbor discovery related parameters. For the basic discovery, there could be more than one discovered neighbor for a given source (interface), whereas there is at most one discovered neighbor for an extended discovery source (local-address and target-address). This is also to be noted that the reason for a targeted neighbor adjacency could be either an active source (locally configured targeted) or passive source (to allow any incoming extended/targeted hellos). A neighbor/adjacency record also contains session-state that helps highlight whether a given adjacency has progressed to subsequent session level or to eventual peer level.

Following captures high level tree hierarchy for neighbor adjacency state.

```

+--rw mpls-ldp!
+--rw discovery
+--rw interfaces
|   +--rw interface* [interface]
|       +--rw address-family* [af]
|           +--ro state
|               +--ro ipv4 (or ipv6)
|                   +--ro hello-adjacencies* [adjacent-address]
|                       +--ro adjacent-address
|                           . . . .
|                           . . . .
+--rw targeted
+--rw address-family* [afi]
+--rw afi          address-family
+--ro state
+--ro ipv4 (or ipv6)
+--ro hello-adjacencies* [local-address adjacent-addresses]
    +--ro local-address
    +--ro adjacent-address
        . . . .
        . . . .
s]

```

Figure 6

3.3.1.2. Peer state

Peer related derived state is presented under peers tree. This is one of the core state that provides info on the session related parameters (mode, authentication, KA timeout etc.), TCP connection info, hello adjacencies for the peer, statistics related to messages and bindings, and capabilities exchange info.

Following captures high level tree hierarchy for peer state.

```

+--rw mpls-ldp!
  +--rw peers
    +--rw peer* [lsr-id]
      +--rw lsr-id
      +--ro state
        +--ro session-ka-holdtime?
        +-- . . . .
        +-- . . . .
        +--ro capability
        + +ro -- . . .
        +--ro address-family
          +--ro ipv4 (or ipv6)
            +--ro hello-adjacencies* [local-address adjacent-address]
            |
            | . . . .
            | . . . .
        +--ro received-peer-state
          +--ro . . . .
          +--ro capability
          |
          +--ro . . . .
        +--ro statistics
          +-- . . . .
          +-- . . . .

```

Figure 7

3.3.1.3. Bindings state

Binding state provides information on LDP FEC-label bindings as well as address binding for both inbound (received) as well as outbound (advertised) direction. FEC-label bindings are presented as a FEC-centric view, and address bindings are presented as an address-centric view:

```
FEC-Label bindings:
  FEC 200.1.1.1/32:
    advertised: local-label 16000
    peer 192.168.0.2:0
    peer 192.168.0.3:0
    peer 192.168.0.4:0
    received:
      peer 192.168.0.2:0, label 16002, used-in-forwarding=Yes
      peer 192.168.0.3:0, label 17002, used-in-forwarding=No
  FEC 200.1.1.2/32:
    . . . .
  FEC 201.1.0.0/16:
    . . . .

Address bindings:
  Addr 1.1.1.1:
    advertised
  Addr 1.1.1.2:
    advertised
  Addr 2.2.2.2:
    received, peer 192.168.0.2
  Addr 2.2.2.22:
    received, peer 192.168.0.2
  Addr 3.3.3.3:
    received, peer 192.168.0.3
  Addr 3.3.3.33:
    received, peer 192.168.0.3
```

Figure 8

Note that all local addresses are advertised to all peers and hence no need to provide per-peer information for local address advertisement. Furthermore, note that it is easy to derive a peer-centric view for the bindings from the information already provided in this model.

Following captures high level tree hierarchy for bindings state.

```

+--rw mpls-ldp!
  +--rw global
    +--rw address-family* [afi]
      +--rw afi          address-family
      +--ro state
        +--ro ipv4 (or ipv6)
          +--ro bindings
            +--ro address* [address]
              | +--ro address
              | +--ro direction?   advertised-received
              | +--ro peer?        leafref
            +--ro fec-label* [fec]
              +--ro fec          inet:ipv4-prefix
              +--ro peer* [peer advertisement-type]
                +--ro peer          leafref
                +--ro advertisement-type   advertised-received
                +--ro label?          mpls:mpls-label
                +--ro used-in-forwarding?  boolean

```

Figure 9

3.3.1.4. Capabilities state

LDP capabilities state comprise two types of information - global information (such as timer etc.), and per-peer information.

Following captures high level tree hierarchy for LDP capabilities state.

```

+--rw mpls-ldp!
  +--rw global
    | +--ro state
    |   +--ro capability
    |     +--ro . . . .
    |     +--ro . . . .
  +--rw peers
    +--rw peer* [lsr-id]
      +--rw lsr-id    yang:dotted-quad
      +--ro state
        +--ro received-peer-state
        +--ro capability
          +--ro . . . .
          +--ro . . . .

```

Figure 10

3.4. Notifications

This model defines a list of notifications to inform client of important events detected during the protocol operation. These events include events related to changes in the operational state of an LDP peer, hello adjacency, and FEC etc. It is to be noted that an LDP FEC is treated as operational (up) as long as it has at least 1 NHLFE with outgoing label.

Following is a simplified graphical representation of the data model for LDP notifications.

```

module: ietf-mpls-ldp
notifications:
  +---n mpls-ldp-peer-event
  |   +--ro event-type?    oper-status-event-type
  |   +--ro peer-ref?      leafref
  +---n mpls-ldp-hello-adjacency-event
  |   +--ro event-type?    oper-status-event-type
  |   +--ro (hello-adjacency-type)?
  |   |   +--:(targeted)
  |   |   |   +--ro targeted
  |   |   |   +--ro target-address?    inet:ip-address
  |   |   +--:(link)
  |   |   |   +--ro link
  |   |   |   +--ro next-hop-interface?    mpls-interface-ref
  |   |   |   +--ro next-hop-address?     inet:ip-address
  +---n mpls-ldp-fec-event
  |   +--ro event-type?    oper-status-event-type
  |   +--ro prefix?        inet:ip-prefix

```

Figure 11

3.5. Actions

This model defines a list of rpcs that allow performing an action or executing a command on the protocol. For example, it allows to clear (reset) LDP peers, hello-adjacencies, and statistics. The model makes an effort to provide different level of control so that a user is able to either clear all, or clear all for a given type, or clear a specific entity.

Following is a simplified graphical representation of the data model for LDP actions.

```

module: ietf-mpls-ldp
rpcs:
  +---x mpls-ldp-clear-peer
  |   +---w input
  |   +---w lsr-id?    union
  +---x mpls-ldp-clear-hello-adjacency
  |   +---w input
  |   +---w hello-adjacency
  |   +---w (hello-adjacency-type)?
  |   |   +---:(targeted)
  |   |   |   +---w targeted!
  |   |   |   +---w target-address?    inet:ip-address
  |   |   +---:(link)
  |   |   +---w link!
  |   |   +---w next-hop-interface?    mpls-interface-ref
  |   |   +---w next-hop-address?     inet:ip-address
  +---x mpls-ldp-clear-peer-statistics
  |   +---w input
  |   +---w lsr-id?    union

```

Figure 12

4. mLDP YANG Model

4.1. Overview

Due to tight dependency of mLDP on LDP, mLDP model builds on top of LDP model defined earlier in the document. Following are the main mLDP areas and documents that are within the scope of this model:

- o mLDP Base Specification [RFC6388]
- o mLDP Recursive FEC [RFC6512]
- o Targeted mLDP [RFC7060]
- o mLDP Fast-Reroute (FRR)
 - * Node Protection [RFC7715]
 - * Multicast-only
- o Hub-and-Spoke Multipoint LSPs [RFC7140]
- o mLDP In-band Signaling [RFC6826] (future revision)
- o mLDP In-band signaling in a VRF [RFC7246]

- o mLDP In-band Signaling with Wildcards [RFC7438] (future revision)
- o Configured Leaf LSPs (manually provisioned)

[Ed Note: Some of the topics in the above list are to be addressed/added in later revision of this document].

4.2. Configuration

4.2.1. Configuration Hierarchy

In terms of overall configuration layout, following figure highlights extensions to LDP configuration model to incorporate mLDP:

```

+-- mpls-ldp
+-- ...
+-- ...
+-- mldp
+-- ...
+-- ...
+-- address-family* [af]
+-- af
+-- ...
+-- ...
+-- global
+-- ...
+-- capability
+-- ...
+-- ...
+-- mldp
+-- ...
+-- ...
+-- discovery
+-- ...
+-- ...
+-- forwarding-nexthop
+-- interfaces
+-- interface* [interface]
+-- interface
+-- address-family* [af]
+-- af
+-- ...
+-- mldp-disable
+-- peers
+-- ...
+-- ...
+-- peer* [lsr-id]
+-- ...
+-- ...
+-- capability
+-- ...
+-- ...
+-- mldp
+-- ...
+-- ...

```

Figure 13

From above hierarchy, we can categorize mLDP configuration parameters into two types:

- o Parameters that leverage/extend LDP containers and parameters
- o Parameters that are mLDP specific

Following subsections first describe mLDP specific configuration parameters, followed by those leveraging LDP.

4.2.2. mldp container

mldp container resides directly under "mpls-ldp" and holds the configuration related to items that are mLDP specific. The main items under this container are:

- o mLDP enabling: To enable mLDP under a (VRF) routing instance, mldp container is enabled under LDP. Given that mLDP requires LDP signalling, it is not sensible to allow disabling LDP control plane under a (VRF) network-instance while requiring mLDP to be enabled for the same. However, if a user wishes only to allow signalling for multipoint FECs on an LDP/mLDP enabled VRF instance, he/she can use LDP label-policies to disable unicast FECs under the VRF.
- o mLDP per-AF features: mLDP manages its own list of IP address-families and the features enabled underneath. The per-AF mLDP configuration items include:
 - * Multicast-only FRR: This enables Multicast-only FRR functionality for a given AF under mLDP. The feature allows route-policy to be configured for finer control/applicability of the feature.
 - * Recursive FEC: The recursive-fec feature [RFC6512] can be enabled per AF with a route-policy.
 - * Configured Leaf LSPs: To provision multipoint leaf LSP manually, a container is provided per-AF under LDP. The configuration is flexible and allows a user to specify MP LSPs of type p2mp or mp2mp with IPv4 or IPv6 root address(es) by using either LSP-Id or (S,G).

Targeted mLDP feature specification [RFC7060] do not require any mLDP specific configuration. It, however, requires LDP upstream-label-assignment capability [RFC6389] to be enabled.

4.2.3. Leveraging LDP containers

mLDP configuration model leverages following configuration areas and containers that are already defined for LDP:

- o Capabilities: A new container "mldp" is defined under Capabilities container. This new container specifies any mLDP specific capabilities and their parameters. Moreover, a new "mldp" container is also added under per-peer capability container to override/control mLDP specific capabilities on a peer level. In the scope of this document, the most important capabilities related to mLDP are p2mp, mp2mp, make-before-break, hub-and-spoke, and node-protection.
- o Discovery and Peer: mLDP requires LDP discovery and peer procedures to form mLDP peering. A peer is treated as mLDP peer only when either P2MP or MP2MP capabilities have been successfully exchanged with the peer. If a user wish to selectively enable or disable mLDP with a LDP-enabled peer, he/she may use per-peer mLDP capabilities configuration. [Ed Note: The option to control mLDP enabling/disabling on a peer-list is being explored for future]. In most common deployments, it is desirable to disable mLDP (capabilities announcements) on a targeted-only LDP peering, where targeted-only peer is the one whose discovery sources are targeted only. In future revision, a configuration option for this support will also be provided.
- o Forwarding: By default, mLDP is allowed to select any of the LDP enabled interface as a downstream interface towards a nexthop (LDP/mLDP peer) for MP LSP programming. However, a configuration option is provided to allow mLDP to exclude a given interface from such a selection. Note that such a configuration option will be useful only when there are more than one interfaces available for the downstream selection.

This goes without saying that mLDP configuration tree follows the same approach as LDP, where the tree comprise leafs for intended configuration.

4.2.4. YANG tree

The following figure captures the YANG tree for mLDP configuration. To keep the focus, the figure has been simplified to display only mLDP items without any LDP items.

```
module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
```

```

+--rw global
|
|   +--rw config
|   |
|   |   +--rw capability
|   |   |
|   |   |   +--rw mldp {mldp}?
|   |   |   |
|   |   |   |   +--rw p2mp
|   |   |   |   |
|   |   |   |   |   +--rw enable?    boolean
|   |   |   |   |
|   |   |   |   +--rw mp2mp
|   |   |   |   |
|   |   |   |   |   +--rw enable?    boolean
|   |   |   |   |
|   |   |   |   +--rw make-before-break
|   |   |   |   |
|   |   |   |   |   +--rw enable?    boolean
|   |   |   |   |   +--rw switchover-delay? uint16
|   |   |   |   |   +--rw timeout?    uint16
|   |   |   +--rw hub-and-spoke {capability-mldp-hsmp}?
|   |   |   |
|   |   |   |   +--rw enable?    boolean
|   |   |   +--rw node-protection {capability-mldp-node-protection}?
|   |   |   |
|   |   |   |   +--rw plr?        boolean
|   |   |   |   +--rw merge-point
|   |   |   |   |
|   |   |   |   |   +--rw enable?    boolean
|   |   |   |   |   +--rw targeted-session-teardown-delay? uint16
|   |   +--rw mldp {mldp}?
|   |   |
|   |   |   +--rw config
|   |   |   |
|   |   |   |   +--rw enable?    boolean
|   |   |   +--rw address-family* [afi]
|   |   |   |
|   |   |   |   +--rw afi
|   |   |   |   |
|   |   |   |   |   ldp-address-family
|   |   |   |   +--rw config
|   |   |   |   |
|   |   |   |   |   +--rw multicast-only-frr {mldp-mofrr}?
|   |   |   |   |   |
|   |   |   |   |   |   +--rw prefix-list?    prefix-list-ref
|   |   |   |   |   +--rw recursive-fec
|   |   |   |   |   |
|   |   |   |   |   |   +--rw prefix-list?    prefix-list-ref
|   |   |   +--rw configured-leaf-lsps
|   |   |   |
|   |   |   |   +--rw p2mp
|   |   |   |   |
|   |   |   |   |   +--rw roots-ipv4
|   |   |   |   |   |
|   |   |   |   |   |   +--rw root* [root-address]
|   |   |   |   |   |   |
|   |   |   |   |   |   |   +--rw root-address    inet:ipv4-address
|   |   |   |   |   |   |   +--rw lsp* [lsp-id source-address group-address]
|   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   +--rw lsp-id        uint16
|   |   |   |   |   |   |   |   +--rw source-address  inet:ipv4-address
|   |   |   |   |   |   |   |   +--rw group-address   inet:ipv4-address-no-zone
|   |   |   |   +--rw roots-ipv6
|   |   |   |   |
|   |   |   |   |   +--rw root* [root-address]
|   |   |   |   |   |
|   |   |   |   |   |   +--rw root-address    inet:ipv6-address
|   |   |   |   |   |   +--rw lsp* [lsp-id source-address group-address]
|   |   |   |   |   |   |
|   |   |   |   |   |   |   +--rw lsp-id        uint16
|   |   |   |   |   |   |   +--rw source-address  inet:ipv6-address
|   |   |   |   |   |   |   +--rw group-address   inet:ipv6-address-no-zone
|   |   |   +--rw mp2mp
|   |   |   |
|   |   |   |   +--rw roots-ipv4
|   |   |   |   |
|   |   |   |   |   +--rw root* [root-address]

```

```

|--rw root-address          inet:ipv4-address
+--rw lsp* [lsp-id source-address group-address]
|   |--rw lsp-id            uint16
|   |--rw source-address     inet:ipv4-address
|   |--rw group-address      inet:ipv4-address-no-zone
+--rw roots-ipv6
|   |--rw root* [root-address]
|   |--rw root-address       inet:ipv6-address
|   |--rw lsp* [lsp-id source-address group-address]
|   |   |--rw lsp-id        uint16
|   |   |--rw source-address inet:ipv6-address
|   |   |--rw group-address  inet:ipv6-address-no-zone
+--rw forwarding-nexthop {forwarding-nexthop-config}?
|   +--rw interfaces
|   |   +--rw interface* [interface]
|   |   |   |--rw interface      if:interface-ref
|   |   |   |--rw address-family* [afi]
|   |   |   |   |--rw afi          address-family
|   |   |   |   |--rw config
|   |   |   |   |   |--rw mldp-disable?    boolean {mldp}?
+--rw peers
|   +--rw peer* [lsr-id]
|   |   +--rw capability
|   |   |   +--rw mldp {mldp}?
|   |   |   |   +--rw p2mp
|   |   |   |   |   +--rw enable?    boolean
|   |   |   |   +--rw mp2mp
|   |   |   |   |   +--rw enable?    boolean
|   |   |   +--rw make-before-break
|   |   |   |   +--rw enable?          boolean
|   |   |   |   +--rw switchover-delay? uint16
|   |   |   |   +--rw timeout?         uint16
|   |   +--rw hub-and-spoke {capability-mldp-hsmp}?
|   |   |   +--rw enable?    boolean
|   |   +--rw node-protection {capability-mldp-node-protection}?
|   |   |   +--rw plr?        boolean
|   |   +--rw merge-point
|   |   |   +--rw enable?          boolean
|   |   +--rw targeted-session-teardown-delay? uint16

```

Figure 14

4.3. Operational State

Operational state of mLDP can be queried and obtained from this read-only container "mldp" which resides under mpls-ldp container.

Please note this state tree refers both the configuration "applied" state as well as the "derived" state related to the mLDP protocol.

Following is a simplified graphical representation of the data model for mLDP operational state:

```

module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
  +--rw mpls-ldp!
    +--rw global
      +--ro state
        +--ro capability
          +--ro mldp {mldp}?
            +--ro p2mp
              | +--ro enable?    boolean
            +--ro mp2mp
              | +--ro enable?    boolean
            +--ro make-before-break
              | +--ro enable?          boolean
              | +--ro switchover-delay? uint16
              | +--ro timeout?         uint16
            +--ro hub-and-spoke {capability-mldp-hsmp}?
              | +--ro enable?    boolean
            +--ro node-protection {capability-mldp-node-protection}?
              +--ro plr?          boolean
              +--ro merge-point
                +--ro enable?          boolean
                +--ro targeted-session-teardown-delay? uint16
          +--rw mldp {mldp}?
            +--ro state
              +--ro enable?    boolean
            +--rw address-family* [afi]
              +--rw afi
                ldp-address-family
              +--ro state
                +--ro multicast-only-frr {mldp-mofrr}?
                  | +--ro prefix-list?    prefix-list-ref
                +--ro recursive-fec
                  | +--ro prefix-list?    prefix-list-ref
                +--ro ipv4
                  +--ro roots
                    +--ro root* [root-address]
                      +--ro root-address    inet:ipv4-address
                      +--ro is-self?        boolean
                      +--ro reachability* [address interface]
                        +--ro address        inet:ipv4-address
                        +--ro interface      mpls-interface-ref

```

```
| | | | | +--ro peer? leafref  
| | | | | +--ro bindings  
| | | | | +--ro opaque-type-lspid  
| | | | | | +--ro fec-label* [root-address lsp-id recur-root-addr  
ess recur-rd]  
| | | | |  
| | | | | +--ro root-address inet:ipv4-address  
| | | | | +--ro lsp-id uint32  
| | | | | +--ro recur-root-address inet:ip-address  
| | | | | +--ro recur-rd route-distinguisher  
| | | | | +--ro multipoint-type? multipoint-type  
| | | | | +--ro peer* [direction peer advertisement-type]  
| | | | | | +--ro direction downstream-upstream  
| | | | | | +--ro peer leafref  
| | | | | | +--ro advertisement-type advertised-received  
| | | | | | +--ro label? mpls:mpls-label  
| | | | | | +--ro mbb-role? enumeration  
| | | | | | +--ro mofrr-role? enumeration  
| | | | | +--ro opaque-type-src  
| | | | | | +--ro fec-label* [root-address source-address group-a  
ddress rd recur-root-address recur-rd]  
| | | | |  
| | | | | +--ro root-address inet:ipv4-address  
| | | | | +--ro source-address inet:ip-address  
| | | | | +--ro group-address inet:ip-address-no-zon  
e  
| | | | |  
| | | | | +--ro rd route-distinguisher  
| | | | | +--ro recur-root-address inet:ip-address  
| | | | | +--ro recur-rd route-distinguisher  
| | | | | +--ro multipoint-type? multipoint-type  
| | | | | +--ro peer* [direction peer advertisement-type]  
| | | | | | +--ro direction downstream-upstream  
| | | | | | +--ro peer leafref  
| | | | | | +--ro advertisement-type advertised-received  
| | | | | | +--ro label? mpls:mpls-label  
| | | | | | +--ro mbb-role? enumeration  
| | | | | | +--ro mofrr-role? enumeration  
+--ro opaque-type-bidir  
+--ro fec-label* [root-address rp group-address rd re  
cur-root-address recur-rd]  
| | | | |  
| | | | | +--ro root-address inet:ipv4-address  
| | | | | +--ro rp inet:ip-address  
| | | | | +--ro group-address inet:ip-address-no-zon  
e  
| | | | |  
| | | | | +--ro rd route-distinguisher  
| | | | | +--ro recur-root-address inet:ip-address  
| | | | | +--ro recur-rd route-distinguisher  
| | | | | +--ro multipoint-type? multipoint-type  
| | | | | +--ro peer* [direction peer advertisement-type]  
| | | | | | +--ro direction downstream-upstream  
| | | | | | +--ro peer leafref  
| | | | | | +--ro advertisement-type advertised-received  
| | | | | | +--ro label? mpls:mpls-label  
| | | | | | +--ro mbb-role? enumeration  
| | | | | | +--ro mofrr-role? enumeration
```



```

+--ro ipv6
+--ro roots
|   +--ro root* [root-address]
|       +--ro root-address      inet:ipv6-address
|       +--ro is-self?          boolean
|       +--ro reachability* [address interface]
|           +--ro address      inet:ipv6-address
|           +--ro interface    mpls-interface-ref
|           +--ro peer?        leafref
+--ro bindings
|   +--ro opaque-type-lspid
|       +--ro fec-label* [root-address lsp-id recur-root-addr
ess recur-rd]
|           +--ro root-address      inet:ipv6-address
|           +--ro lsp-id            uint32
|           +--ro recur-root-address inet:ip-address
|           +--ro recur-rd          route-distinguisher
|           +--ro multipoint-type?  multipoint-type
|           +--ro peer* [direction peer advertisement-type]
|               +--ro direction    downstream-upstream
|               +--ro peer          leafref
|               +--ro advertisement-type advertised-received
|               +--ro label?        mpls:mpls-label
|               +--ro mbb-role?      enumeration
|               +--ro mofrr-role?    enumeration
|   +--ro opaque-type-src
|       +--ro fec-label* [root-address source-address group-a
address rd recur-root-address recur-rd]
|           +--ro root-address      inet:ipv6-address
|           +--ro source-address    inet:ip-address
|           +--ro group-address     inet:ip-address-no-zon
e
|           +--ro rd                route-distinguisher
|           +--ro recur-root-address inet:ip-address
|           +--ro recur-rd          route-distinguisher
|           +--ro multipoint-type?  multipoint-type
|           +--ro peer* [direction peer advertisement-type]
|               +--ro direction    downstream-upstream
|               +--ro peer          leafref
|               +--ro advertisement-type advertised-received
|               +--ro label?        mpls:mpls-label
|               +--ro mbb-role?      enumeration
|               +--ro mofrr-role?    enumeration
|   +--ro opaque-type-bidir
|       +--ro fec-label* [root-address rp group-address rd re
cur-root-address recur-rd]
|           +--ro root-address      inet:ipv6-address
|           +--ro rp                inet:ip-address
|           +--ro group-address     inet:ip-address-no-zon
e
|           +--ro rd                route-distinguisher
|           +--ro recur-root-address inet:ip-address
|           +--ro recur-rd          route-distinguisher

```

```

+---ro multipoint-type? multipoint-type
+---ro peer* [direction peer advertisement-type]
+---ro direction downstream-upstream
+---ro peer leafref
+---ro advertisement-type advertised-received
+---ro label? mpls:mpls-label
+---ro mbb-role? enumeration
+---ro mofrr-role? enumeration
+---rw forwarding-nexthop {forwarding-nexthop-config}?
+---rw interfaces
+---rw interface* [interface]
+---rw address-family* [afi]
+---ro state
+---ro mldp-disable? boolean {mldp}?
+---rw peers
+---rw peer* [lsr-id]
+---ro state
+---ro capability
+---ro mldp {mldp}?
+---ro p2mp
+---ro enable? boolean
+---ro mp2mp
+---ro enable? boolean
+---ro make-before-break
+---ro enable? boolean
+---ro switchover-delay? uint16
+---ro timeout? uint16
+---ro hub-and-spoke {capability-mldp-hsmp}?
+---ro enable? boolean
+---ro node-protection {capability-mldp-node-protection}?
+---ro plr? boolean
+---ro merge-point
+---ro enable? boolean
+---ro targeted-session-teardown-delay? uint16
+---ro received-peer-state
+---ro capability
+---ro mldp {mldp}?
+---ro p2mp
+---ro enable? boolean
+---ro mp2mp
+---ro enable? boolean
+---ro make-before-break
+---ro enable? boolean
+---ro hub-and-spoke
+---ro enable? boolean
+---ro node-protection
+---ro plr? boolean
+---ro merge-point? boolean

```

Figure 15

4.3.1. Derived states

Following are main areas for which mLDP operational derived state is defined:

- o Root
- o Bindings (FEC-label)
- o Capabilities

4.3.1.1. Root state

Root address is a fundamental construct for MP FEC bindings and LSPs. The root state provides information on all the known roots in a given address-family, and their information on the root reachability (as learnt from RIB). In case of multi-path reachability to a root, the selection of upstream path is done on per-LSP basis at the time of LSP setup. Similarly, when protection mechanisms like MBB or MoFRR are in place, the path designation as active/standby or primary/backup is also done on per LSP basis. It is to be noted that a given root can be shared amongst multiple P2MP and/or MP2MP LSPs. Moreover, an LSP can be signaled to more than one root for RNR purposes.

The following diagram illustrates a root database on a branch/transit LSR:

```
root 1.1.1.1:
  path1:
    RIB: GigEthernet 1/0, 12.1.0.2;
    LDP: peer 192.168.0.1:0
  path2:
    RIB: GigEthernet 2/0, 12.2.0.2;
    LDP: peer 192.168.0.3:0

root 2.2.2.2:
  path1:
    RIB: 3.3.3.3;                (NOTE: This is a recursive path)
    LDP: peer 192.168.0.3:0      (NOTE: T-mLDP peer)

root 9.9.9.9:
  . . . .
```

Figure 16

A root entry on a root LSR itself will be presented as follows:

```
root 9.9.9.9:
  is-self
```

Figure 17

4.3.1.2. Bindings state

Binding state provides information on mLDP FEC-label bindings for both P2MP and MP2MP FEC types. Like LDP, the FEC-label binding derived state is presented in a FEC-centric view per address-family, and provides information on both inbound (received) and outbound (advertised) bindings. The FEC is presented as (root-address, opaque-type-data) and the direction (upstream or downstream) is picked with respect to root reachability. In case of MBB or/and MoFRR, the role of a given peer binding is also provided with respect to MBB (active or standby) or/and MoFRR (primary or backup).

This document covers following type of opaque values with their keys in the operational model of mLDP bindings:

Opaque Type	Key	RFC
Generic LSP Identifier	LSP Id	[RFC6388]
Transit IPv4 Source	Source, Group	[RFC6826]
Transit IPv6 Source	Source, Group	[RFC6826]
Transit IPv4 Bidir	RP, Group	[RFC6826]
Transit IPv6 Bidir	RP, Group	[RFC6826]
Transit VPNv4 Source	Source, Group, RD	[RFC7246]
Transit VPNv6 Source	Source, Group, RD	[RFC7246]
Transit VPNv4 Bidir	RP, Group, RD	[RFC7246]
Transit VPNv6 Bidir	RP, Group, RD	[RFC7246]
Recursive Opaque	Root	[RFC6512]
VPN-Recursive Opaque	Root, RD	[RFC6512]

Table 1: MP Opaque Types and keys

It is to be noted that there are three basic types (LSP Id, Source, and Bidir) and then there are variants (VPN, recursive, VPN-recursive) on top of these basic types.

Following captures high level tree hierarchy for mLDP bindings state:

```

+--rw mpls-ldp!
  +--rw mldp
    +--rw address-family* [afi]
      +--rw afi          address-family
      +--ro state
        +--ro ipv4 (or ipv6)
          +--ro bindings
            +--ro opaque-type-xxx [root-address, type-specific-key]
              +--ro root-address
              +--ro ...
              +--ro recur-root-address      inet:ipv4-address
              +--ro recur-rd                route-distinguisher
              +--ro multipoint-type?        multipoint-type
              +--ro peer* [direction peer advertisement-type]
                +--ro direction            downstream-upstream
                +--ro peer                  leafref
                +--ro advertisement-type    advertised-received
                +--ro label?                mpls:mpls-label
                +--ro mbb-role?             enumeration
                +--ro mofrr-role?           enumeration

```

Figure 18

In the above tree, the type-specific-key varies with the base type as listed in earlier Table 1. For example, if the opaque type is Generic LSP Identifier, then the type-specific-key will be a uint32 value corresponding to the LSP. Please see the complete model for all other types.

Moreover, the binding tree defines only three types of sub-trees (i.e. lspid, src, and bidir) which is able to map the respective variants (vpn, recursive, and vpn-recursive) accordingly. For example, the key for opaque-type-src is [R, S, G, rd, recur-R, recur-RD], where basic type will specify (R, S,G,-, -, -), VPN type will specify (R, S,G, rd, -, -), recursive type will specify [R, S,G, -, recur-R, -] and VPN-recursive type will specify [R, S,G, -, recur-R, recur-rd].

It is important to take note of the following:

- o The address-family ipv4/ipv4 applies to "root" address in the mLDP binding tree. The other addresses (source, group, RP etc) do not have to be of the same address family type as the root.
- o The "recur-root-address" field applies to Recursive opaque type, and (recur-root-address, recur-rd) fields applies to VPN-Recursive opaque types as defined in [RFC6512]
- o In case of a recursive FEC, the address-family of the recur-root-address could be different than the address-family of the root address of original encapsulated MP FEC

The following diagram illustrates the FEC-label binding information structure for a P2MP (Transit IPv4 Source type) LSP on a branch/transit LSR:

```

FEC (root 2.2.2.2, S=192.168.1.1, G=224.1.1.1):
  type: p2mp
  upstream:
    advertised:
      peer 192.168.0.1:0, label 16000 (local)
  downstream:
    received:
      peer 192.168.0.2:0, label 17000 (remote)
      peer 192.168.0.3:0, label 18000 (remote)

```

Figure 19

The following diagram illustrates the FEC-label binding information structure for a similar MP2MP LSP on a branch/transit LSR:

```

FEC (root 2.2.2.2, RP=192.168.9.9, G=224.1.1.1):
  type: mp2mp
  upstream:
    advertised:
      peer 192.168.0.1:0, label 16000 (local)
    received:
      peer 192.168.0.1:0, label 17000 (remote)
  downstream:
    advertised:
      peer 192.168.0.2:0, label 16001 (local), MBB role=active
      peer 192.168.0.3:0, label 16002 (local), MBB role=standby
    received:
      peer 192.168.0.2:0, label 17001 (remote)
      peer 192.168.0.3:0, label 18001 (remote)

```

Figure 20

4.3.1.3. Capabilities state

Like LDP, mLDP capabilities state comprise two types of information - global information and per-peer information.

4.4. Notifications

mLDP notification module consists of notification related to changes in the operational state of an mLDP FEC. Following is a simplified graphical representation of the data model for mLDP notifications:

```

notifications:
  +---n mpls-ml dp-fec-event
    +---ro event-type?      oper-status-event-type
    +---ro tree-type?       multipoint-type
    +---ro root?            inet:ip-address
    +---ro (lsp-key-type)?
      +---:(lsp-id-based)
        | +---ro lsp-id?      uint16
      +---:(source-group-based)
        +---ro source-address?  inet:ip-address
        +---ro group-address?   inet:ip-address

```

Figure 21

4.5. Actions

Currently, no RPCs/actions are defined for mLDP.

5. Open Items

Following is a list of open items that are to be discussed and addressed in future revisions of this document:

- o Close on augmentation off "mpls" list in "ietf-mpls" defined in [I-D.saad-mpls-base-yang]
- o Align operational state modeling with other routing procols and [I-D.openconfig-netmod-opstate]
- o Complete the section on Protocol-centric implementations and all-vrfs
- o Specify default values for configuration parameters
- o Revisit and cut down on the scope of the document and number of features it is trying to cover
- o Split the model into a base and extended items
- o Add statistics for mLDP root LSPs and bindings
- o Extend the "Configured Leaf LSPs" for various type of opaque-types
- o Extend mLDP notifications for other types of opaque values as well
- o Close on single vs separate document for mLDP Yang

6. YANG Specification

Following are actual YANG definition for LDP and mLDP constructs defined earlier in the document.

```
<CODE BEGINS> file "ietf-mpls-ldp@2016-07-08.yang" -->

module ietf-mpls-ldp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp";
  // replace with IANA namespace when assigned
  prefix ldp;

  import ietf-inet-types {
```



```
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-ip {
    prefix "ip";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-mpls {
    prefix "mpls";
  }

  organization
    "IETF MPLS Working Group";
  contact
    "WG Web:    <http://tools.ietf.org/wg/teas/>
    WG List:    <mailto:teas@ietf.org>

    WG Chair:   Loa Andersson
                <mailto:loa@pi.nu>

    WG Chair:   Ross Callon
                <mailto:rcallon@juniper.net>

    WG Chair:   George Swallow
                <mailto:swallow.ietf@gmail.com>

    Editor:     Kamran Raza
                <mailto:skraza@cisco.com>

    Editor:     Rajiv Asati
                <mailto:rajiva@cisco.com>

    Editor:     Xufeng Liu
                <mailto:xliu@kuatrotech.com>

    Editor:     Santosh Esale
```

```
<mailto:sesale@juniper.net>

Editor:   Xia Chen
          <mailto:jescia.chenxia@huawei.com>

Editor:   Himanshu Shah
          <mailto:hshah@ciena.com>;

description
  "This YANG module defines the essential components for the
  management of Multi-Protocol Label Switching (MPLS) Label
  Distribution Protocol (LDP) and Multipoint LDP (mLDP).";

revision 2016-07-08 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Data Model for MPLS LDP and mLDP.";
}

/*
 * Features
 */

feature admin-down-config {
  description
    "This feature indicates that the system allows to configure
    administrative down on a VRF instance and a peer.";
}

feature all-af-policy-config {
  description
    "This feature indicates that the system allows to configure
    policies that are applied to all address families.";
}

feature capability-end-of-lib {
  description
    "This feature indicates that the system allows to configure
    LDP end-of-lib capability.";
}

feature capability-mldp-hsmp {
  description
    "This feature indicates that the system allows to configure
    mLDP hub-and-spoke-multipoint capability.";
}
```

```
feature capability-mldp-node-protection {
  description
    "This feature indicates that the system allows to configure
    mLDP node-protection capability.";
}

feature capability-typed-wildcard-fec {
  description
    "This feature indicates that the system allows to configure
    LDP typed-wildcard-fec capability.";
}

feature capability-upstream-label-assignment {
  description
    "This feature indicates that the system allows to configure
    LDP upstream label assignment capability.";
}

feature forwarding-nexthop-config {
  description
    "This feature indicates that the system allows to configure
    forwarding nexthop on interfaces.";
}

feature global-session-authentication {
  description
    "This feature indicates that the system allows to configure
    authentication at global level.";
}

feature graceful-restart-helper-mode {
  description
    "This feature indicates that the system supports graceful
    restart helper mode.";
}

feature mldp {
  description
    "This feature indicates that the system supports Multicast
    LDP (mLDP).";
}

feature mldp-mofrr {
  description
    "This feature indicates that the system supports mLDP
    Multicast only FRR (MoFRR).";
}
```

```
feature per-interface-timer-config {
  description
    "This feature indicates that the system allows to configure
    interface hello timers at the per-interface level.";
}

feature per-peer-graceful-restart-config {
  description
    "This feature indicates that the system allows to configure
    graceful restart at the per-peer level.";
}

feature per-peer-session-attributes-config {
  description
    "This feature indicates that the system allows to configure
    session attributes at the per-peer level.";
}

feature policy-extended-discovery-config {
  description
    "This feature indicates that the system allows to configure
    policies to control the acceptance of extended neighbor
    discovery hello messages.";
}

feature policy-label-assignment-config {
  description
    "This feature indicates that the system allows to configure
    policies to assign labels according to certain prefixes.";
}

feature policy-ordered-label-config {
  description
    "This feature indicates that the system allows to configure
    ordered label policies.";
}

feature session-downstream-on-demand-config {
  description
    "This feature indicates that the system allows to configure
    session downstream-on-demand";
}

/*
 * Typedefs
 */
typedef ldp-address-family {
  type identityref {
```

```
    base rt:address-family;
  }
  description
    "LDP address family type.";
}

typedef duration32-inf {
  type union {
    type uint32;
    type enumeration {
      enum "infinite" {
        description "The duration is infinite.";
      }
    }
  }
  units seconds;
  description
    "Duration represented as 32 bit seconds with infinite.";
}

typedef advertised-received {
  type enumeration {
    enum advertised {
      description "Advertised information.";
    }
    enum received {
      description "Received information.";
    }
  }
  description
    "Received or advertised.";
}

typedef downstream-upstream {
  type enumeration {
    enum downstream {
      description "Downstream information.";
    }
    enum upstream {
      description "Upstream information.";
    }
  }
  description
    "Received or advertised.";
}

typedef label-adv-mode {
  type enumeration {
```

```
    enum downstream-unsolicited {
      description "Downstream Unsolicited.";
    }
    enum downstream-on-demand {
      description "Downstream on Demand.";
    }
  }
  description
    "Label Advertisement Mode.";
}

typedef mpls-interface-ref {
  type leafref {
    path "/rt:routing/mpls:mpls/mpls:interface/mpls:name";
  }
  description
    "This type is used by data models that need to reference
    mpls interfaces.";
}

typedef multipoint-type {
  type enumeration {
    enum p2mp {
      description "Point to multipoint.";
    }
    enum mp2mp {
      description "Multipoint to multipoint.";
    }
  }
  description
    "p2mp or mp2mp.";
}

typedef neighbor-list-ref {
  type string;
  description
    "A type for a reference to a neighbor list.";
}

typedef peer-list-ref {
  type string;
  description
    "A type for a reference to a peer list.";
}

typedef prefix-list-ref {
  type string;
  description
```

```
    "A type for a reference to a prefix list.";
}

typedef oper-status-event-type {
  type enumeration {
    enum up {
      value 1;
      description
        "Operational status changed to up.";
    }
    enum down {
      value 2;
      description
        "Operational status changed to down.";
    }
  }
  description "Operational status event type for notifications.";
}

typedef route-distinguisher {
  type string {
  }
  description
    "Type definition for route distinguisher.";
  reference
    "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs).";
}

/*
 * Identities
 */
identity adjacency-flag-base {
  description "Base type for adjacency flags.";
}

identity adjacency-flag-active {
  base "adjacency-flag-base";
  description
    "This adjacency is configured and actively created.";
}

identity adjacency-flag-passive {
  base "adjacency-flag-base";
  description
    "This adjacency is not configured and passively accepted.";
}

/*
```

```
* Groupings
*/

grouping adjacency-state-attributes {
  description
    "Adjacency state attributes.";

  leaf-list flag {
    type identityref {
      base "adjacency-flag-base";
    }
    description "Adjacency flags.";
  }
  container hello-holdtime {
    description "Hello holdtime state.";
    leaf adjacent {
      type uint16;
      units seconds;
      description "Peer holdtime.";
    }
    leaf negotiated {
      type uint16;
      units seconds;
      description "Negotiated holdtime.";
    }
    leaf remaining {
      type uint16;
      units seconds;
      description "Remaining holdtime.";
    }
  }

  leaf next-hello {
    type uint16;
    units seconds;
    description "Time to send the next hello message.";
  }

  container statistics {
    description
      "Statistics objects.";

    leaf discontinuity-time {
      type yang:date-and-time;
      mandatory true;
      description
        "The time on the most recent occasion at which any one or
        more of this interface's counters suffered a
```



```
        discontinuity.  If no such discontinuities have occurred
        since the last re-initialization of the local management
        subsystem, then this node contains the time the local
        management subsystem re-initialized itself.";
    }

    leaf hello-received {
        type yang:counter64;
        description
            "The number of hello messages received.";
    }
    leaf hello-dropped {
        type yang:counter64;
        description
            "The number of hello messages received.";
    }
} // statistics
} // adjacency-state-attributes

grouping basic-discovery-timers {
    description
        "Basic discovery timer attributes.";
    leaf hello-holdtime {
        type uint16 {
            range 15..3600;
        }
        units seconds;
        description
            "The time interval for which a LDP link Hello adjacency
            is maintained in the absence of link Hello messages from
            the LDP neighbor";
    }
    leaf hello-interval {
        type uint16 {
            range 5..1200;
        }
        units seconds;
        description
            "The interval between consecutive LDP link Hello messages
            used in basic LDP discovery";
    }
} // basic-discovery-timers

grouping binding-address-state-attributes {
    description
        "Address binding attributes";
    leaf advertisement-type {
        type advertised-received;
```

```
    description
      "Received or advertised.";
  }
  leaf peer {
    type leafref {
      path "../.../peers/peer/lsr-id";
    }
    must "../advertisement-type = 'received'" {
      description
        "Applicable for received address.";
    }
    description
      "LDP peer from which this address is received.";
  } // peer
} // binding-address-state-attributes

grouping binding-label-state-attributes {
  description
    "Label binding attributes";
  list peer {
    key "peer advertisement-type";
    description
      "List of advertised and received peers.";
    leaf peer {
      type leafref {
        path "../.../peers/peer/lsr-id";
      }
      description
        "LDP peer from which this binding is received,
        or to which this binding is advertised.";
    }
    leaf advertisement-type {
      type advertised-received;
      description
        "Received or advertised.";
    }
    leaf label {
      type mpls:mpls-label;
      description
        "Advertised (outbound) or received (inbound)
        label.";
    }
    leaf used-in-forwarding {
      type boolean;
      description
        "'true' if the lable is used in forwarding.";
    }
  }
} // peer
```

```
} // binding-label-state-attributes

grouping extended-discovery-policy-attributes {
  description
    "LDP policy to control the acceptance of extended neighbor
    discovery hello messages.";
  container hello-accept {
    if-feature policy-extended-discovery-config;
    description
      "Extended discovery acceptance policies.";

    leaf enable {
      type boolean;
      description
        "'true' to accept; 'false' to deny.";
    }
    leaf neighbor-list {
      type neighbor-list-ref;
      description
        "The name of a peer ACL.";
    }
  } // hello-accept
} // extended-discovery-policy-attributes

grouping extended-discovery-timers {
  description
    "Extended discovery timer attributes.";
  leaf hello-holdtime {
    type uint16 {
      range 15..3600;
    }
    units seconds;
    description
      "The time interval for which LDP targeted Hello adjacency
      is maintained in the absence of targeted Hello messages
      from an LDP neighbor.";
  }
  leaf hello-interval {
    type uint16 {
      range 5..3600;
    }
    units seconds;
    description
      "The interval between consecutive LDP targeted Hello
      messages used in extended LDP discovery.";
  }
}
```

```
} // extended-discovery-timers

grouping global-attributes {
  description "Configuration attributes at global level.";

  uses instance-attributes;
} // global-attributes

grouping graceful-restart-attributes {
  description
    "Graceful restart configuration attributes.";
  container graceful-restart {
    description
      "Attributes for graceful restart.";
    leaf enable {
      type boolean;
      description
        "Enable or disable graceful restart.";
    }
    leaf helper-enable {
      if-feature graceful-restart-helper-mode;
      type boolean;
      description
        "Enable or disable graceful restart helper mode.";
    }
    leaf reconnect-time {
      type uint16 {
        range 10..1800;
      }
      units seconds;
      description
        "Specifies the time interval that the remote LDP peer
        must wait for the local LDP peer to reconnect after the
        remote peer detects the LDP communication failure.";
    }
    leaf recovery-time {
      type uint16 {
        range 30..3600;
      }
      units seconds;
      description
        "Specifies the time interval, in seconds, that the remote
        LDP peer preserves its MPLS forwarding state after
        receiving the Initialization message from the restarted
        local LDP peer.";
    }
    leaf forwarding-holdtime {
      type uint16 {
```

```
        range 30..3600;
    }
    units seconds;
    description
        "Specifies the time interval, in seconds, before the
        termination of the recovery phase.";
    }
} // graceful-restart
} // graceful-restart-attributes

grouping graceful-restart-attributes-per-peer {
    description
        "Per peer graceful restart configuration attributes.";
    container graceful-restart {
        description
            "Attributes for graceful restart.";
        leaf enable {
            type boolean;
            description
                "Enable or disable graceful restart.";
        }
        leaf reconnect-time {
            type uint16 {
                range 10..1800;
            }
            units seconds;
            description
                "Specifies the time interval that the remote LDP peer
                must wait for the local LDP peer to reconnect after the
                remote peer detects the LDP communication failure.";
        }
        leaf recovery-time {
            type uint16 {
                range 30..3600;
            }
            units seconds;
            description
                "Specifies the time interval, in seconds, that the remote
                LDP peer preserves its MPLS forwarding state after
                receiving the Initialization message from the restarted
                local LDP peer.";
        }
    }
} // graceful-restart
} // graceful-restart-attributes-per-peer

grouping instance-attributes {
    description "Configuration attributes at instance level.";
```

```
container capability {
  description "Configure capability.";
  container end-of-lib {
    if-feature capability-end-of-lib;
    description
      "Configure end-of-lib capability.";
    leaf enable {
      type boolean;
      description
        "Enable end-of-lib capability.";
    }
  }
  container typed-wildcard-fec {
    if-feature capability-typed-wildcard-fec;
    description
      "Configure typed-wildcard-fec capability.";
    leaf enable {
      type boolean;
      description
        "Enable typed-wildcard-fec capability.";
    }
  }
  container upstream-label-assignment {
    if-feature capability-upstream-label-assignment;
    description
      "Configure upstream label assignment capability.";
    leaf enable {
      type boolean;
      description
        "Enable upstream label assignment.";
    }
  }
  container mldp {
    if-feature mldp;

    description
      "Multipoint capabilities.";
    uses mldp-capabilities;
  }
} // capability

uses graceful-restart-attributes;

leaf igp-synchronization-delay {
  type uint16 {
    range 3..60;
  }
  units seconds;
}
```

```
    description
      "Sets the interval that the LDP waits before notifying the
      Interior Gateway Protocol (IGP) that label exchange is
      completed so that IGP can start advertising the normal
      metric for the link.";
  }
  leaf lsr-id {
    type yang:dotted-quad;
    description "Router ID.";
  }
} // instance-attributes

grouping ldp-adjacency-ref {
  description
    "An absolute reference to an LDP adjacency.";
  choice hello-adjacency-type {
    description
      "Interface or targeted adjacency.";
    case targeted {
      container targeted {
        description "Targeted adjacency.";
        leaf target-address {
          type inet:ip-address;
          description
            "The target address.";
        }
      } // targeted
    }
    case link {
      container link {
        description "Link adjacency.";
        leaf next-hop-interface {
          type mpls-interface-ref;
          description
            "Interface connecting to next-hop.";
        }
        leaf next-hop-address {
          type inet:ip-address;
          must "../next-hop-interface" {
            description
              "Applicable when interface is specified.";
          }
        }
        description
          "IP address of next-hop.";
      }
    } // link
  }
}
```

```
    }  
  } // ldp-adjacency-ref  
  
  grouping ldp-fec-event {  
    description  
      "A LDP FEC event.";   
    leaf prefix {  
      type inet:ip-prefix;  
      description  
        "FEC.";   
    }  
  } // ldp-fec-event  
  
  grouping ldp-peer-ref {  
    description  
      "An absolute reference to an LDP peer.";   
    leaf peer-ref {  
      type leafref {  
        path "/rt:routing/rt:control-plane-protocols/mpls-ldp/"  
          + "peers/peer/lsr-id";  
      }  
      description  
        "Reference to an LDP peer.";   
    }  
  } // ldp-peer-ref  
  
  grouping mldp-capabilities {  
    description  
      "mLDP capabilities.";   
    container p2mp {  
      description  
        "Configure point-to-multipoint capability.";   
      leaf enable {  
        type boolean;  
        description  
          "Enable point-to-multipoint.";   
      }  
    }  
    container mp2mp {  
      description  
        "Configure multipoint-to-multipoint capability.";   
      leaf enable {  
        type boolean;  
        description  
          "Enable multipoint-to-multipoint.";   
      }  
    }  
    container make-before-break {
```



```
description
  "Configure make-before-break capability.";
leaf enable {
  type boolean;
  description
    "Enable make-before-break.";
}
leaf switchover-delay {
  type uint16;
  units seconds;
  description
    "Switchover delay in seconds.";
}
leaf timeout {
  type uint16;
  units seconds;
  description
    "Timeout in seconds.";
}
}
container hub-and-spoke {
  if-feature capability-mldp-hsmp;
  description
    "Configure hub-and-spoke-multipoint capability.";
  reference
    "RFC7140: LDP Extensions for Hub and Spoke Multipoint
    Label Switched Path";
  leaf enable {
    type boolean;
    description
      "Enable hub-and-spoke-multipoint.";
  }
}
container node-protection {
  if-feature capability-mldp-node-protection;
  description
    "Configure node-protection capability.";
  reference
    "RFC7715: mLDP Node Protection.";
  leaf plr {
    type boolean;
    description
      "Point of Local Repair capable for MP LSP node
      protection.";
  }
}
container merge-point {
  description
    "Merge Point capable for MP LSP node protection.";
```

```
    leaf enable {
      type boolean;
      description
        "Enable merge point capability.";
    }
    leaf targeted-session-teardown-delay {
      type uint16;
      units seconds;
      description
        "Targeted session teardown delay.";
    }
  } // merge-point
} // mldp-capabilities

grouping mldp-configured-lsp-roots {
  description
    "mLDP roots containers.";

  container roots-ipv4 {

    when "../../af = 'ipv4'" {
      description
        "Only for IPv4.";
    }
    description
      "Configured IPv4 multicast LSPs.";
    list root {
      key "root-address";
      description
        "List of roots for configured multicast LSPs.";

      leaf root-address {
        type inet:ipv4-address;
        description
          "Root address.";
      }
    }

    list lsp {
      must "(lsp-id = 0 and source-address != '0.0.0.0' and "
        + "group-address != '0.0.0.0') or "
        + "(lsp-id != 0 and source-address = '0.0.0.0' and "
        + "group-address = '0.0.0.0')";
      description
        "A LSP can be identified by either <lsp-id> or
        <source-address, group-address>.";
    }
    key "lsp-id source-address group-address";
  }
}
```

```
    description
      "List of LSPs.";
    leaf lsp-id {
      type uint16;
      description "ID to identify the LSP.";
    }
    leaf source-address {
      type inet:ipv4-address;
      description
        "Source address.";
    }
    leaf group-address {
      type inet:ipv4-address-no-zone;
      description
        "Group address.";
    }
  } // list lsp
} // list root
} // roots-ipv4

container roots-ipv6 {

  when "../../../af = 'ipv6'" {
    description
      "Only for IPv6.";
  }
  description
    "Configured IPv6 multicast LSPs.";

  list root {
    key "root-address";
    description
      "List of roots for configured multicast LSPs.";

    leaf root-address {
      type inet:ipv6-address;
      description
        "Root address.";
    }
  }

  list lsp {
    must "(lsp-id = 0 and source-address != ':::' and "
      + "group-address != ':::') or "
      + "(lsp-id != 0 and source-address = ':::' and "
      + "group-address = ':::')" {
      description
        "A LSP can be identified by either <lsp-id> or
        <source-address, group-address>.";
    }
  }
}
```

```
    }
    key "lsp-id source-address group-address";
    description
      "List of LSPs.";
    leaf lsp-id {
      type uint16;
      description "ID to identify the LSP.";
    }
    leaf source-address {
      type inet:ipv6-address;
      description
        "Source address.";
    }
    leaf group-address {
      type inet:ipv6-address-no-zone;
      description
        "Group address.";
    }
  } // list lsp
} // list root
} // roots-ipv6
} // mldp-configured-lsp-roots

grouping mldp-fec-event {
  description
    "A mLDP FEC event.";
  leaf tree-type {
    type multipoint-type;
    description
      "p2mp or mp2mp.";
  }
  leaf root {
    type inet:ip-address;
    description
      "Root address.";
  }
  choice lsp-key-type {
    description
      "LSP ID based or source-group based .";
    case lsp-id-based {
      leaf lsp-id {
        type uint16;
        description
          "ID to identify the LSP.";
      }
    }
    case source-group-based {
      leaf source-address {

```

```
        type inet:ip-address;
        description

            "LSP source address.";
    }
    leaf group-address {
        type inet:ip-address;
        description
            "Multicast group address.";
    }
} // case source-group-based
}
} // mldp-fec-event

grouping mldp-binding-label-state-attributes {
    description
        "mLDP label binding attributes.";

    leaf multipoint-type {
        type multipoint-type;
        description
            "The type of mutipoint, p2mp or mp2mp.";
    }
    list peer {
        key "direction peer advertisement-type";
        description
            "List of advertised and received peers.";
        leaf direction {
            type downstream-upstream;
            description
                "Downstream or upstream.";
        }
        leaf peer {
            type leafref {
                path
                    ".../.../.../.../.../.../.../.../.../peers/peer/lsr-id";
            }
            description
                "LDP peer from which this binding is received,
                or to which this binding is advertised.";
        }
        leaf advertisement-type {
            type advertised-received;
            description
                "Advertised or received.";
        }
        leaf label {
            type mpls:mpls-label;
        }
    }
}
```

```
        description
            "Advertised (outbound) or received (inbound) label.";
    }
    leaf mbb-role {
        when "../direction = 'upstream'" {
            description
                "For upstream.";
        }
        type enumeration {
            enum none {
                description "MBB is not enabled.";
            }
            enum active {
                description "This LSP is active.";
            }
            enum inactive {
                description "This LSP is inactive.";
            }
        }
        description
            "The MBB status of this LSP.";
    }
    leaf mofrr-role {
        when "../direction = 'upstream'" {
            description
                "For upstream.";
        }
        type enumeration {
            enum none {
                description "MOFRR is not enabled.";
            }
            enum primary {
                description "This LSP is primary.";
            }
            enum backup {
                description "This LSP is backup.";
            }
        }
        description
            "The MOFRR status of this LSP.";
    }
} // peer
} // mldp-binding-label-state-attributes

grouping peer-af-policy-container {
    description
        "LDP policy attribute container under peer address-family.";
    container label-policy {
```

```
description
  "Label policy attributes.";
container advertise {
  description
    "Label advertising policies.";
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Applies the prefix list to outgoing label
      advertisements.";
  }
}
container accept {
  description
    "Label advertisement acceptance policies.";
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Applies the prefix list to incoming label
      advertisements.";
  }
} // accept
} // label-policy
} // peer-af-policy-container

grouping peer-attributes {
  description "Peer configuration attributes.";

  leaf session-ka-holdtime {
    type uint16 {
      range 45..3600;
    }
    units seconds;
    description
      "The time interval after which an inactive LDP session
      terminates and the corresponding TCP session closes.
      Inactivity is defined as not receiving LDP packets from the
      peer.";
  }
  leaf session-ka-interval {
    type uint16 {
      range 15..1200;
    }
    units seconds;
    description
      "The interval between successive transmissions of keepalive
      packets. Keepalive packets are only sent in the absence of
      other LDP packets transmitted over the LDP session.";
  }
}
```

```
    }  
  } // peer-attributes  
  
  grouping peer-authentication {  
    description  
      "Peer authentication attributes.";  
    leaf session-authentication-md5-password {  
      type string {  
        length "1..80";  
      }  
      description  
        "Assigns an encrypted MD5 password to an LDP  
        peer";  
    } // md5-password  
  } // peer-authentication  
  
  grouping peer-state-derived {  
    description "Peer derived state attributes.";  
  
    container label-advertisement-mode {  
      description "Label advertisement mode state.";  
      leaf local {  
        type label-adv-mode;  
        description  
          "Local Label Advertisement Mode.";  
      }  
      leaf peer {  
        type label-adv-mode;  
        description  
          "Peer Label Advertisement Mode.";  
      }  
      leaf negotiated {  
        type label-adv-mode;  
        description  
          "Negotiated Label Advertisement Mode.";  
      }  
    }  
  }  
  leaf next-keep-alive {  
    type uint16;  
    units seconds;  
    description "Time to send the next KeepAlive message.";  
  }  
  
  leaf peer-ldp-id {  
    type yang:dotted-quad;  
    description "Peer LDP ID.";  
  }  
}
```



```
container received-peer-state {
  description "Peer features.";

  uses graceful-restart-attributes-per-peer;

  container capability {
    description "Configure capability.";
    container end-of-lib {
      description
        "Configure end-of-lib capability.";
      leaf enable {
        type boolean;
        description
          "Enable end-of-lib capability.";
      }
    }
  }
  container typed-wildcard-fec {
    description
      "Configure typed-wildcard-fec capability.";
    leaf enable {
      type boolean;
      description
        "Enable typed-wildcard-fec capability.";
    }
  }
  container upstream-label-assignment {
    description
      "Configure upstream label assignment capability.";
    leaf enable {
      type boolean;
      description
        "Enable upstream label assignment.";
    }
  }
  container mldp {
    if-feature mldp;
    description
      "Multipoint capabilities.";

    container p2mp {
      description
        "Configure point-to-multipoint capability.";
      leaf enable {
        type boolean;
        description
          "Enable point-to-multipoint.";
      }
    }
  }
}
```

```
container mp2mp {
  description
    "Configure multipoint-to-multipoint capability.";
  leaf enable {
    type boolean;
    description
      "Enable multipoint-to-multipoint.";
  }
}
container make-before-break {
  description
    "Configure make-before-break capability.";
  leaf enable {
    type boolean;
    description
      "Enable make-before-break.";
  }
}
container hub-and-spoke {
  description
    "Configure hub-and-spoke-multipoint capability.";
  reference
    "RFC7140: LDP Extensions for Hub and Spoke Multipoint
    Label Switched Path";
  leaf enable {
    type boolean;
    description
      "Enable hub-and-spoke-multipoint.";
  }
}
container node-protection {
  description
    "Configure node-protection capability.";
  reference
    "RFC7715: mLDP Node Protection.";
  leaf plr {
    type boolean;
    description
      "Point of Local Repair capable for MP LSP node
      protection.";
  }
  leaf merge-point {
    type boolean;
    description
      "Merge Point capable for MP LSP node protection.";
  } // merge-point
} // node-protection
} // mldp
```

```
    } // capability
  } // received-peer-state

  container session-holdtime {
    description "Session holdtime state.";
    leaf peer {
      type uint16;
      units seconds;
      description "Peer holdtime.";
    }
    leaf negotiated {
      type uint16;
      units seconds;
      description "Negotiated holdtime.";
    }
    leaf remaining {
      type uint16;
      units seconds;
      description "Remaining holdtime.";
    }
  } // session-holdtime

  leaf session-state {
    type enumeration {
      enum non-existent {
        description "NON EXISTENT state. Transport disconnected.";
      }
      enum initialized {
        description "INITIALIZED state.";
      }
      enum openrec {
        description "OPENREC state.";
      }
      enum opensent {
        description "OPENSENT state.";
      }
      enum operational {
        description "OPERATIONAL state.";
      }
    }
    description
      "Representing the operational status.";
  }

  container tcp-connection {
    description "TCP connection state.";
    leaf local-address {
      type inet:ip-address;
    }
  }
}
```

```
        description "Local address.";
    }
    leaf local-port {
        type inet:port-number;
        description "Local port.";
    }
    leaf remote-address {
        type inet:ip-address;
        description "Remote address.";
    }
    leaf remote-port {
        type inet:port-number;
        description "Remote port.";
    }
} // tcp-connection

leaf up-time {
    type string;
    description "Up time. The interval format in ISO 8601.";
}

container statistics {
    description
        "Statistics objects.";

    leaf discontinuity-time {
        type yang:date-and-time;
        mandatory true;
        description
            "The time on the most recent occasion at which any one or
            more of this interface's counters suffered a
            discontinuity.  If no such discontinuities have occurred
            since the last re-initialization of the local management
            subsystem, then this node contains the time the local
            management subsystem re-initialized itself.";
    }

    container received {
        description "Inbound statistics.";
        uses statistics-peer-received-sent;
    }
    container sent {
        description "Outbound statistics.";
        uses statistics-peer-received-sent;
    }
}

leaf total-addresses {
    type uint32;
```

```
        description
            "The number of learned addresses.";
    }
    leaf total-labels {
        type uint32;
        description
            "The number of learned labels.";
    }
    leaf total-fec-label-bindings {
        type uint32;
        description
            "The number of learned label-address bindings.";
    }
} // statistics
} // peer-state-derived

grouping policy-container {
    description
        "LDP policy attributes.";
    container label-policy {
        description
            "Label policy attributes.";
        container independent-mode {
            description
                "Independent label policy attributes.";
            container assign {

                if-feature policy-label-assignment-config;
                description
                    "Label assignment policies";
                choice prefix-option {
                    description
                        "Use either prefix-list or host-routes-only.";
                    case prefix-list {
                        leaf prefix-list {
                            type prefix-list-ref;
                            description
                                "Assign labels according to certain prefixes.";
                        }
                    }
                    case host-routes-only {
                        leaf host-routes-only {
                            type boolean;
                            description
                                "'true' to apply host routes only.";
                        }
                    }
                }
            } // prefix-option
        }
    }
}
```

```
}
container advertise {
  description
    "Label advertising policies.";
  container explicit-null {
    description
      "Enables an egress router to advertise an
       explicit null label (value 0) in place of an
       implicit null label (value 3) to the
       penultimate hop router.";
    leaf enable {
      type boolean;
      description
        "'true' to enable explicit null.";
    }
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Prefix list name. Applies the filters in the
         specified prefix list to label
         advertisements.
         If the prefix list is not specified, explicit
         null label advertisement is enabled for all
         directly connected prefixes.";
    }
  }
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Applies the prefix list to outgoing label
       advertisements.";
  }
}
container accept {
  description
    "Label advertisement acceptance policies.";
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Applies the prefix list to incoming label
       advertisements.";
  }
}
} // independent-mode
container ordered-mode {
  if-feature policy-ordered-label-config;
  description
```

```
    "Ordered label policy attributes.";
  container egress-lsr {
    description
      "Egress LSR label assignment policies";
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Assign labels according to certain prefixes.";
    }
  }
  container advertise {
    description
      "Label advertising policies.";
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Applies the prefix list to outgoing label
        advertisements.";
    }
  }
  container accept {
    description
      "Label advertisement acceptance policies.";
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Applies the prefix list to incoming label
        advertisements.";
    }
  }
} // ordered-mode
} // label-policy
} // policy-container

grouping statistics-peer-received-sent {
  description
    "Inbound and outbound statistic counters.";
  leaf total-octets {
    type yang:counter64;
    description
      "The total number of octets sent or received.";
  }
  leaf total-messages {
    type yang:counter64;
    description
      "The number of messages sent or received.";
  }
  leaf address {
```

```
    type yang:counter64;
    description
      "The number of address messages sent or received.";
  }
  leaf address-withdraw {
    type yang:counter64;
    description
      "The number of address-withdraw messages sent or received.";
  }
  leaf initialization {
    type yang:counter64;
    description
      "The number of initialization messages sent or received.";
  }
  leaf keepalive {
    type yang:counter64;
    description
      "The number of keepalive messages sent or received.";
  }
  leaf label-abort-request {
    type yang:counter64;
    description
      "The number of label-abort-request messages sent or
      received.";
  }
  leaf label-mapping {
    type yang:counter64;
    description
      "The number of label-mapping messages sent or received.";
  }
  leaf label-release {
    type yang:counter64;
    description
      "The number of label-release messages sent or received.";
  }
  leaf label-request {
    type yang:counter64;
    description
      "The number of label-request messages sent or received.";
  }
  leaf label-withdraw {
    type yang:counter64;
    description
      "The number of label-withdraw messages sent or received.";
  }
  leaf notification {
    type yang:counter64;
    description
```



```
        "The number of messages sent or received.";
    }
} // statistics-peer-received-sent

/*
 * Configuration data nodes
 */

augment "/rt:routing/rt:control-plane-protocols" {
    description "LDP augmentation.";

    container mpls-ldp {
        presence "Container for LDP protocol.";
        description
            "Container for LDP protocol.";

        container global {
            description
                "Global attributes for LDP.";
            container config {
                description
                    "Configuration data.";
                uses global-attributes;
            }
            container state {
                config false;
                description
                    "Operational state data.";
                uses global-attributes;
            }
        }

        container mldp {
            if-feature mldp;
            description
                "mLDP attributes at per instance level. Defining
                attributes here does not enable any MP capabilities.
                MP capabilities need to be explicitly enabled under
                container capability.";

            container config {
                description
                    "Configuration data.";
                leaf enable {
                    type boolean;
                    description
                        "Enable mLDP.";
                }
            }
        }
    }
}
```

```
container state {
  config false;
  description

    "Operational state data.";
  leaf enable {
    type boolean;
    description
      "Enable mLDP.";
  }
}

list address-family {
  key "afi";
  description
    "Per-af params.";
  leaf afi {
    type ldp-address-family;
    description
      "Address family type value.";
  }
}

container config {
  description
    "Configuration data.";
  container multicast-only-frr {
    if-feature mldp-mofrr;
    description
      "Multicast only FRR (MoFRR) policy.";
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Enables MoFRR for the specified access list.";
    }
  } // multicast-only-frr
  container recursive-fec {
    description
      "Recursive FEC policy.";
    leaf prefix-list {
      type prefix-list-ref;
      description
        "Enables recursive FEC for the specified access
        list.";
    }
  } // recursive-for
}
container state {
  config false;
```

```
description
  "Operational state data.";
container multicast-only-frr {
  if-feature mldp-mofrr;

  description
    "Multicast only FRR (MoFRR) policy.";
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Enables MoFRR for the specified access list.";
  }
} // multicast-only-frr
container recursive-fec {
  description
    "Recursive FEC policy.";
  leaf prefix-list {
    type prefix-list-ref;
    description
      "Enables recursive FEC for the specified access
      list.";
  }
} // recursive-fec

container ipv4 {
  when "../..afi = 'ipv4'" {
    description
      "Only for IPv4.";
  }
  description
    "IPv4 state information.";
  container roots {
    description
      "IPv4 multicast LSP roots.";
    list root {
      key "root-address";
      description
        "List of roots for configured multicast LSPs.";

      leaf root-address {
        type inet:ipv4-address;
        description
          "Root address.";
      }

      leaf is-self {
        type boolean;
        description
```

```
        "This is the root.";
    }

    list reachability {
        key "address interface";
        description
            "A next hop for reachability to root,
            as a RIB view.";
        leaf address {
            type inet:ipv4-address;
            description
                "The next hop address to reach root.";
        }
        leaf interface {
            type mpls-interface-ref;
            description
                "Interface connecting to next-hop.";
        }
        leaf peer {
            type leafref {
                path
                    "../.../peers/peer/"
                    + "lsr-id";
            }
            description
                "LDP peer from which this next hop can be
                reached.";
        }
    }
} // list root
} // roots
container bindings {
    description
        "mLDP FEC to label bindings.";
    container opaque-type-lspid {
        description
            "The type of opaque value element is
            the generic LSP identifier";
        reference
            "RFC6388: Label Distribution Protocol
            Extensions for Point-to-Multipoint and
            Multipoint-to-Multipoint Label Switched
            Paths.";
        list fec-label {
            key
                "root-address lsp-id "
                + "recur-root-address recur-rd";
            description
```

```
    "List of FEC to label bindings.";
  leaf root-address {
    type inet:ipv4-address;
    description
      "Root address.";
  }
  leaf lsp-id {
    type uint32;
    description "ID to identify the LSP.";
  }
  leaf recur-root-address {
    type inet:ip-address;
    description
      "Recursive root address.";
    reference
      "RFC6512: Using Multipoint LDP When the
      Backbone Has No Route to the Root";
  }
  leaf recur-rd {
    type route-distinguisher;
    description
      "Route Distinguisher in the VPN-Recursive
      Opaque Value.";
    reference
      "RFC6512: Using Multipoint LDP When the
      Backbone Has No Route to the Root";
  }
  uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-lspid

container opaque-type-src {
  description
    "The type of opaque value element is
    the transit source TLV";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched
    Paths.";
  list fec-label {
    key
      "root-address source-address group-address "
      + "rd recur-root-address recur-rd";
    description
      "List of FEC to label bindings.";
    leaf root-address {
      type inet:ipv4-address;
```

```
        description
            "Root address.";
    }
    leaf source-address {
        type inet:ip-address;
        description
            "Source address.";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address.";
    }
    leaf rd {
        type route-distinguisher;
        description
            "Route Distinguisher.";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    leaf recur-root-address {
        type inet:ip-address;
        description
            "Recursive root address.";
        reference
            "RFC6512: Using Multipoint LDP When the
            Backbone Has No Route to the Root";
    }
    leaf recur-rd {
        type route-distinguisher;
        description
            "Route Distinguisher in the VPN-Recursive
            Opaque Value.";
        reference
            "RFC6512: Using Multipoint LDP When the
            Backbone Has No Route to the Root";
    }
    uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-src

container opaque-type-bidir {
    description
        "The type of opaque value element is
        the generic LSP identifier";
```

```
reference
  "RFC6826: Multipoint LDP In-Band Signaling for
  Point-to-Multipoint and
  Multipoint-to-Multipoint Label Switched
  Paths.";
list fec-label {
  key
    "root-address rp group-address "
    + "rd recur-root-address recur-rd";
  description
    "List of FEC to label bindings.";
  leaf root-address {
    type inet:ipv4-address;
    description
      "Root address.";
  }
  leaf rp {
    type inet:ip-address;
    description
      "RP address.";
  }
  leaf group-address {
    type inet:ip-address-no-zone;
    description
      "Group address.";
  }
  leaf rd {
    type route-distinguisher;
    description
      "Route Distinguisher.";
    reference
      "RFC7246: Multipoint Label Distribution
      Protocol In-Band Signaling in a Virtual
      Routing and Forwarding (VRF) Table
      Context.";
  }
  leaf recur-root-address {
    type inet:ip-address;
    description
      "Recursive root address.";
    reference
      "RFC6512: Using Multipoint LDP When the
      Backbone Has No Route to the Root";
  }
  leaf recur-rd {
    type route-distinguisher;
    description
      "Route Distinguisher in the VPN-Recursive
```

```
        Opaque Value.";
        reference
        "RFC6512: Using Multipoint LDP When the
        Backbone Has No Route to the Root";
    }
    uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-bidir
} // bindings
} // ipv4

container ipv6 {
    when "../..afi = 'ipv6'" {
        description
        "Only for IPv6.";
    }
    description
    "IPv6 state information.";
    container roots {
        description
        "IPv6 multicast LSP roots.";
        list root {
            key "root-address";
            description
            "List of roots for configured multicast LSPs.";

            leaf root-address {
                type inet:ipv6-address;
                description
                "Root address.";
            }

            leaf is-self {
                type boolean;
                description
                "This is the root.";
            }
        }

        list reachability {
            key "address interface";
            description
            "A next hop for reachability to root,
            as a RIB view.";
            leaf address {
                type inet:ipv6-address;
                description
                "The next hop address to reach root.";
            }
        }
    }
}
```



```
    leaf interface {
      type mpls-interface-ref;
      description
        "Interface connecting to next-hop.";
    }
    leaf peer {
      type leafref {
        path
          "../.../.../.../.../.../.../.../peers/peer/"
          + "lsr-id";
      }
      description
        "LDP peer from which this next hop can be
        reached.";
    }
  }
} // list root
} // roots
container bindings {
  description
    "mLDP FEC to label bindings.";
  container opaque-type-lspid {
    description
      "The type of opaque value element is
      the generic LSP identifier";
    reference
      "RFC6388: Label Distribution Protocol
      Extensions for Point-to-Multipoint and
      Multipoint-to-Multipoint Label Switched
      Paths.";
    list fec-label {
      key
        "root-address lsp-id "
        + "recur-root-address recur-rd";
      description
        "List of FEC to label bindings.";
      leaf root-address {
        type inet:ipv6-address;
        description
          "Root address.";
      }
      leaf lsp-id {
        type uint32;
        description "ID to identify the LSP.";
      }
      leaf recur-root-address {
        type inet:ip-address;
        description

```

```
        "Recursive root address.";
        reference
        "RFC6512: Using Multipoint LDP When the
        Backbone Has No Route to the Root";
    }
    leaf recur-rd {
        type route-distinguisher;
        description
        "Route Distinguisher in the VPN-Recursive
        Opaque Value.";
        reference
        "RFC6512: Using Multipoint LDP When the
        Backbone Has No Route to the Root";
    }
    uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-lspid

container opaque-type-src {
    description
    "The type of opaque value element is
    the transit Source TLV";
    reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched
    Paths.";
    list fec-label {
        key
        "root-address source-address group-address "
        + "rd recur-root-address recur-rd";
        description
        "List of FEC to label bindings.";
        leaf root-address {
            type inet:ipv6-address;
            description
            "Root address.";
        }
        leaf source-address {
            type inet:ip-address;
            description
            "Source address.";
        }
        leaf group-address {
            type inet:ip-address-no-zone;
            description
            "Group address.";
        }
    }
}
```

```
leaf rd {
  type route-distinguisher;
  description
    "Route Distinguisher.";
  reference
    "RFC7246: Multipoint Label Distribution
    Protocol In-Band Signaling in a Virtual
    Routing and Forwarding (VRF) Table
    Context.";
}
leaf recur-root-address {
  type inet:ip-address;
  description
    "Recursive root address.";
  reference
    "RFC6512: Using Multipoint LDP When the
    Backbone Has No Route to the Root";
}
leaf recur-rd {
  type route-distinguisher;
  description
    "Route Distinguisher in the VPN-Recursive
    Opaque Value.";
  reference
    "RFC6512: Using Multipoint LDP When the
    Backbone Has No Route to the Root";
}
uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-src

container opaque-type-bidir {
  description
    "The type of opaque value element is
    the generic LSP identifier";
  reference
    "RFC6826: Multipoint LDP In-Band Signaling for
    Point-to-Multipoint and
    Multipoint-to-Multipoint Label Switched
    Paths.";
  list fec-label {
    key
      "root-address rp group-address "
      + "rd recur-root-address recur-rd";
    description
      "List of FEC to label bindings.";
    leaf root-address {
      type inet:ipv6-address;
```

```
        description
            "Root address.";
    }
    leaf rp {
        type inet:ip-address;
        description
            "RP address.";
    }
    leaf group-address {
        type inet:ip-address-no-zone;
        description
            "Group address.";
    }
    leaf rd {
        type route-distinguisher;
        description
            "Route Distinguisher.";
        reference
            "RFC7246: Multipoint Label Distribution
            Protocol In-Band Signaling in a Virtual
            Routing and Forwarding (VRF) Table
            Context.";
    }
    leaf recur-root-address {
        type inet:ip-address;
        description
            "Recursive root address.";
        reference
            "RFC6512: Using Multipoint LDP When the
            Backbone Has No Route to the Root";
    }
    leaf recur-rd {
        type route-distinguisher;
        description
            "Route Distinguisher in the VPN-Recursive
            Opaque Value.";
        reference
            "RFC6512: Using Multipoint LDP When the
            Backbone Has No Route to the Root";
    }
    uses mldp-binding-label-state-attributes;
} // fec-label
} // opaque-type-bidir
} // bindings
} // ipv6
} // state

container configured-leaf-lsps {
```

```
    description
      "Configured multicast LSPs.";

    container p2mp {
      description
        "Configured point-to-multipoint LSPs.";
      uses mldp-configured-lsp-roots;
    }
    container mp2mp {
      description
        "Configured multipoint-to-multipoint LSPs.";
      uses mldp-configured-lsp-roots;
    }
  } // configured-leaf-lsps
} // list address-family
} // mldp

list address-family {
  key "afi";
  description
    "Per-vrf per-af params.";
  leaf afi {
    type ldp-address-family;
    description
      "Address family type value.";
  }
}

container config {
  description
    "Configuration data.";
  leaf enable {
    type boolean;
    description
      "'true' to enable the address family.";
  }
  uses policy-container;

  container ipv4 {
    when "../..//afi = 'ipv4'" {
      description
        "Only for IPv4.";
    }
    description
      "IPv4 address family.";
    leaf transport-address {
      type inet:ipv4-address;
      description
        "The transport address advertised in LDP Hello
```

```
        messages.";
    }
} // ipv4
container ipv6 {
    when "../..afi = 'ipv6'" {
        description
            "Only for IPv6.";
    }
    description
        "IPv6 address family.";
    leaf transport-address {
        type inet:ipv6-address;
        description
            "The transport address advertised in LDP Hello
            messages.";
    }
} // ipv6
}
container state {
    config false;
    description
        "Operational state data.";
    leaf enable {
        type boolean;
        description
            "'true' to enable the address family.";
    }
}

uses policy-container;

container ipv4 {
    when "../..afi = 'ipv4'" {
        description
            "Only for IPv4.";
    }
    description
        "IPv4 address family.";
    leaf transport-address {
        type inet:ipv4-address;
        description
            "The transport address advertised in LDP Hello
            messages.";
    }
}

container bindings {
    description
        "LDP address and label binding information.";
    list address {
```

```
    key "address";
    description
        "List of address bindings.";
    leaf address {
        type inet:ipv4-address;
        description
            "Binding address.";
    }
    uses binding-address-state-attributes;
} // binding-address

list fec-label {
    key "fec";
    description
        "List of label bindings.";
    leaf fec {
        type inet:ipv4-prefix;
        description
            "Prefix FEC.";
    }
    uses binding-label-state-attributes;
} // fec-label
} // binding
} // ipv4
container ipv6 {
    when "../..afi = 'ipv6'" {
        description
            "Only for IPv6.";
    }
    description
        "IPv6 address family.";
    leaf transport-address {
        type inet:ipv6-address;
        description
            "The transport address advertised in LDP Hello
            messages.";
    }
}

container binding {
    description
        "LDP address and label binding information.";
    list address {
        key "address";
        description
            "List of address bindings.";
        leaf address {
            type inet:ipv6-address;
            description
```

```
        "Binding address.";
    }
    uses binding-address-state-attributes;
} // binding-address

list fec-label {
    key "fec";
    description
        "List of label bindings.";
    leaf fec {
        type inet:ipv6-prefix;
        description
            "Prefix FEC.";
    }
    uses binding-label-state-attributes;
} // fec-label
} // binding
} // ipv6
} // state
} // address-family

container discovery {
    description
        "Neighbor discovery configuration.";

    container interfaces {
        description
            "A list of interfaces for basic discovery.";
        container config {
            description
                "Configuration data.";
            uses basic-discovery-timers;
        }
        container state {
            config false;
            description

                "Operational state data.";
            uses basic-discovery-timers;
        }
    }

    list interface {
        key "interface";
        description
            "List of LDP interfaces.";
        leaf interface {
            type mpls-interface-ref;
            description
```



```
        "Interface.";
    }
    container config {
        description
            "Configuration data.";
        uses basic-discovery-timers {
            if-feature per-interface-timer-config;
        }
        leaf igp-synchronization-delay {
            if-feature per-interface-timer-config;
            type uint16 {
                range 3..60;
            }
            units seconds;
            description
                "Sets the interval that the LDP waits before
                notifying the Interior Gateway Protocol (IGP)
                that label exchange is completed so that IGP
                can start advertising the normal metric for
                the link.";
        }
    }
    container state {
        config false;
        description
            "Operational state data.";
        uses basic-discovery-timers {
            if-feature per-interface-timer-config;
        }
        leaf igp-synchronization-delay {
            if-feature per-interface-timer-config;
            type uint16 {
                range 3..60;
            }
            units seconds;
            description
                "Sets the interval that the LDP waits before
                notifying the Interior Gateway Protocol (IGP)
                that label exchange is completed so that IGP
                can start advertising the normal metric for
                the link.";
        }
        leaf next-hello {
            type uint16;
            units seconds;
            description "Time to send the next hello message.";
        }
    }
} // state
```

```
list address-family {
  key "afi";
  description
    "Per-vrf per-af params.";
  leaf afi {
    type ldp-address-family;
    description
      "Address family type value.";
  }
  container config {
    description
      "Configuration data.";
    leaf enable {
      type boolean;
      description
        "Enable the address family on the interface.";
    }
  }

  container ipv4 {
    must "/if:interfaces/if:interface"
      + "[name = current()/../../../../interface]/"
      + "ip:ipv4" {
      description
        "Only if IPv4 is enabled on the interface.";
    }
    description
      "IPv4 address family.";
    leaf transport-address {
      type union {
        type enumeration {
          enum "use-interface-address" {
            description
              "Use interface address as the transport
              address.";
          }
        }
        type inet:ipv4-address;
      }
    }
    description
      "IP address to be advertised as the LDP
      transport address.";
  }
}

container ipv6 {
  must "/if:interfaces/if:interface"
    + "[name = current()/../../../../interface]/"
    + "ip:ipv6" {
  }
```

```
        description
            "Only if IPv6 is enabled on the interface.";
    }
    description
        "IPv6 address family.";
    leaf transport-address {
        type union {
            type enumeration {
                enum "use-interface-address" {
                    description
                        "Use interface address as the transport
                        address.";
                }
            }
            type inet:ipv4-address;
        }
        description
            "IP address to be advertised as the LDP
            transport address.";
    }
} // ipv6
}
container state {
    config false;
    description
        "Operational state data.";
    leaf enable {
        type boolean;
        description
            "Enable the address family on the interface.";
    }
}

container ipv4 {
    must "/if:interfaces/if:interface"
        + "[name = current()/../../../interface]/"
        + "ip:ipv4" {
        description
            "Only if IPv4 is enabled on the interface.";
    }
    description
        "IPv4 address family.";
    leaf transport-address {
        type union {
            type enumeration {

                enum "use-interface-address" {
                    description
                        "Use interface address as the transport
```

```
        address.";
    }
}
type inet:ipv4-address;
}
description
    "IP address to be advertised as the LDP
    transport address.";
}

list hello-adjacencies {
    key "adjacent-address";
    description "List of hello adjacencies.";

    leaf adjacent-address {
        type inet:ipv4-address;
        description
            "Neighbor address of the hello adjacency.";
    }

    uses adjacency-state-attributes;

    leaf peer {
        type leafref {
            path "../.../.../.../.../.../peers/peer/"
                + "lsr-id";
        }
        description
            "LDP peer from this adjacency.";
    }
} // hello-adjacencies
}

container ipv6 {
    must "/if:interfaces/if:interface"
        + "[name = current()/../.../.../interface]/"
        + "ip:ipv6" {
        description
            "Only if IPv6 is enabled on the interface.";
    }
}
description
    "IPv6 address family.";
leaf transport-address {
    type union {
        type enumeration {
            enum "use-interface-address" {
                description
                    "Use interface address as the transport
                    address.";
            }
        }
    }
}
```

```

    }
  }
  type inet:ipv4-address;
}
description
  "IP address to be advertised as the LDP
  transport address.";
}

list hello-adjacencies {
  key "adjacent-address";
  description "List of hello adjacencies.";

  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Neighbor address of the hello adjacency.";
  }

  uses adjacency-state-attributes;

  leaf peer {
    type leafref {
      path "../../../../../peers/peer/"
        + "lsr-id";
    }
    description
      "LDP peer from this adjacency.";
  }
} // hello-adjacencies
} // ipv6
}
} // address-family
} // list interface
} // interfaces

container targeted
{
  description
    "A list of targeted neighbors for extended discovery.";
  container config {

    description
      "Configuration data.";
    uses extended-discovery-timers;
    uses extended-discovery-policy-attributes;
  }
  container state {

```

```
    config false;
    description
      "Operational state data.";
    uses extended-discovery-timers;
    uses extended-discovery-policy-attributes;
  }

  list address-family {
    key "afi";
    description
      "Per-af params.";
    leaf afi {
      type ldp-address-family;
      description
        "Address family type value.";
    }
  }

  container state {
    config false;
    description
      "Operational state data.";

    container ipv4 {
      when "../../afi = 'ipv4'" {
        description
          "For IPv4.";
      }
      description
        "IPv4 address family.";
      list hello-adjacencies {
        key "local-address adjacent-address";
        description "List of hello adjacencies.";

        leaf local-address {
          type inet:ipv4-address;
          description
            "Local address of the hello adjacency.";
        }
        leaf adjacent-address {
          type inet:ipv4-address;
          description
            "Neighbor address of the hello adjacency.";
        }
      }

      uses adjacency-state-attributes;

      leaf peer {
        type leafref {
```

```

        path "../../../../../../../peers/peer/"
          + "lsr-id";
      }
      description
        "LDP peer from this adjacency.";
    }
  } // hello-adjacencies
} // ipv4

container ipv6 {
  when "../../../afi = 'ipv6'" {
    description
      "For IPv6.";
  }
  description
    "IPv6 address family.";
  list hello-adjacencies {
    key "local-address adjacent-address";
    description "List of hello adjacencies.";

    leaf local-address {
      type inet:ipv6-address;
      description
        "Local address of the hello adjacency.";
    }
    leaf adjacent-address {
      type inet:ipv6-address;
      description
        "Neighbor address of the hello adjacency.";
    }
  }

  uses adjacency-state-attributes;

  leaf peer {
    type leafref {
      path "../../../../../../../peers/peer/"
        + "lsr-id";
    }
    description
      "LDP peer from this adjacency.";
  }
} // hello-adjacencies
} // ipv6
} // state

container ipv4 {
  when "../../../afi = 'ipv4'" {
    description

```

```
        "For IPv4.";
    }
    description
        "IPv4 address family.";
    list target {
        key "adjacent-address";
        description
            "Targeted discovery params.";

        leaf adjacent-address {
            type inet:ipv4-address;
            description
                "Configures a remote LDP neighbor and enables
                 extended LDP discovery of the specified
                 neighbor.";
        }
    }
    container config {
        description
            "Configuration data.";
        leaf enable {
            type boolean;
            description
                "Enable the target.";
        }
        leaf local-address {
            type inet:ipv4-address;
            description
                "The local address.";
        }
    }
    container state {
        config false;
        description
            "Operational state data.";
        leaf enable {
            type boolean;
            description
                "Enable the target.";
        }
        leaf local-address {
            type inet:ipv4-address;
            description
                "The local address.";
        }
    }
    } // state
} // ipv4
container ipv6 {
```



```
when "../afi = 'ipv6'" {
  description
    "For IPv6.";
}
description
  "IPv6 address family.";
list target {
  key "adjacent-address";
  description
    "Targeted discovery params.";

  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Configures a remote LDP neighbor and enables
       extended LDP discovery of the specified
       neighbor.";
  }
  container config {
    description
      "Configuration data.";
    leaf enable {
      type boolean;
      description
        "Enable the target.";
    }
    leaf local-address {
      type inet:ipv6-address;
      description
        "The local address.";
    }
  }
}
container state {
  config false;
  description
    "Operational state data.";
  leaf enable {
    type boolean;
    description
      "Enable the target.";
  }
  leaf local-address {
    type inet:ipv6-address;
    description
      "The local address.";
  }
} // state
}
```

```
        } // ipv6
      } // address-family
    } // targeted
  } // discovery

  container forwarding-nexthop {
    if-feature forwarding-nexthop-config;
    description
      "Configuration for forwarding nexthop.";

    container interfaces {
      description
        "A list of interfaces on which forwarding is
        disabled.";

      list interface {
        key "interface";
        description
          "List of LDP interfaces.";
        leaf interface {
          type mpls-interface-ref;
          description
            "Interface.";
        }
      }
      list address-family {
        key "afi";
        description
          "Per-vrf per-af params.";
        leaf afi {
          type ldp-address-family;
          description
            "Address family type value.";
        }
      }
      container config {
        description
          "Configuration data.";
        leaf ldp-disable {
          type boolean;
          description
            "Disable LDP forwarding on the interface.";
        }
        leaf mldp-disable {
          if-feature mldp;
          type boolean;
          description
            "Disable mLDP forwarding on the interface.";
        }
      }
    }
  }
```

```
        container state {
            config false;
            description
                "Operational state data.";
            leaf ldp-disable {
                type boolean;
                description
                    "Disable LDP forwarding on the interface.";
            }
            leaf mldp-disable {
                if-feature mldp;

                type boolean;
                description
                    "Disable mLDP forwarding on the interface.";
            }
        } // address-family
    } // list interface
} // interfaces
} // forwarding-nexthop
uses policy-container {
    if-feature all-af-policy-config;
}
} // global

container peers {
    description
        "Peers configuration attributes.";

    container config {
        description
            "Configuration data.";
        uses peer-authentication {
            if-feature global-session-authentication;
        }
        uses peer-attributes;

        container session-downstream-on-demand {
            if-feature session-downstream-on-demand-config;
            description
                "Session downstream-on-demand attributes.";
            leaf enable {
                type boolean;
                description
                    "'true' if session downstream-on-demand is enabled.";
            }
            leaf peer-list {
```

```
        type peer-list-ref;
        description
            "The name of a peer ACL.";
    }
}
}
container state {
    config false;
    description
        "Operational state data.";
    uses peer-authentication {
        if-feature global-session-authentication;
    }
    uses peer-attributes;

    container session-downstream-on-demand {
        if-feature session-downstream-on-demand-config;
        description
            "Session downstream-on-demand attributes.";
        leaf enable {
            type boolean;
            description
                "'true' if session downstream-on-demand is enabled.";
        }
        leaf peer-list {
            type peer-list-ref;
            description
                "The name of a peer ACL.";
        }
    }
}

list peer {
    key "lsr-id";
    description
        "List of peers.";

    leaf lsr-id {
        type yang:dotted-quad;
        description "LSR ID.";
    }

    container config {
        description
            "Configuration data.";
        leaf admin-down {
            type boolean;
            default false;
        }
    }
}
```

```
        description
          "'true' to disable the peer.";
      }

      container capability {
        description
          "Per peer capability";
        container mldp {
          if-feature mldp;
          description
            "mLDP capabilities.";
          uses mldp-capabilities;
        }
      }

      uses peer-af-policy-container {
        if-feature all-af-policy-config;
      }

      uses peer-authentication;

      uses graceful-restart-attributes-per-peer {
        if-feature per-peer-graceful-restart-config;
      }

      uses peer-attributes {
        if-feature per-peer-session-attributes-config;
      }

      container address-family {
        description
          "Per-vrf per-af params.";
        container ipv4 {
          description
            "IPv4 address family.";
          uses peer-af-policy-container;
        }
        container ipv6 {
          description
            "IPv6 address family.";
          uses peer-af-policy-container;
        } // ipv6
      } // address-family
    }
  }
  container state {
    config false;
    description
      "Operational state data.";
```

```
leaf admin-down {
    type boolean;
    default false;
    description
        "'true' to disable the peer.";
}

container capability {
    description
        "Per peer capability";
    container mldp {
        if-feature mldp;
        description
            "mLDP capabilities.";
        uses mldp-capabilities;
    }
}

uses peer-af-policy-container {
    if-feature all-af-policy-config;
}

uses peer-authentication;

uses graceful-restart-attributes-per-peer {
    if-feature per-peer-graceful-restart-config;
}

uses peer-attributes {
    if-feature per-peer-session-attributes-config;
}

container address-family {
    description
        "Per-vrf per-af params.";
    container ipv4 {
        description
            "IPv4 address family.";
        uses peer-af-policy-container;

        list hello-adjacencies {
            key "local-address adjacent-address";
            description "List of hello adjacencies.";

            leaf local-address {
                type inet:ipv4-address;
                description
                    "Local address of the hello adjacency.";
            }
        }
    }
}
```

```
    }
    leaf adjacent-address {
        type inet:ipv4-address;
        description
            "Neighbor address of the hello adjacency.";
    }

    uses adjacency-state-attributes;

    leaf interface {
        type mpls-interface-ref;
        description "Interface for this adjacency.";
    }
} // hello-adjacencies
} // ipv4
container ipv6 {
    description
        "IPv6 address family.";
    uses peer-af-policy-container;

    list hello-adjacencies {
        key "local-address adjacent-address";
        description "List of hello adjacencies.";

        leaf local-address {
            type inet:ipv6-address;
            description
                "Local address of the hello adjacency.";
        }
        leaf adjacent-address {
            type inet:ipv6-address;
            description
                "Neighbor address of the hello adjacency.";
        }
    }

    uses adjacency-state-attributes;

    leaf interface {
        type mpls-interface-ref;
        description "Interface for this adjacency.";
    }
} // hello-adjacencies
} // ipv6
} // address-family

    uses peer-state-derived;
} // state
} // list peer
```

```
    } // peers
  } // container mpls-ldp
}

/*
 * RPCs
 */
rpc mpls-ldp-clear-peer {
  description
    "Clears the session to the peer.";
  input {
    leaf lsr-id {
      type union {
        type yang:dotted-quad;
        type uint32;
      }
      description
        "LSR ID of peer to be cleared. If this is not provided
        then all peers are cleared";
    }
  }
}

rpc mpls-ldp-clear-hello-adjacency {
  description
    "Clears the hello adjacency";
  input {
    container hello-adjacency {
      description
        "Link adjacency or targettted adjacency. If this is not
        provided then all hello adjacencies are cleared";
      choice hello-adjacency-type {
        description "Adjacency type.";
        case targeted {
          container targeted {
            presence "Present to clear targeted adjacencies.";
            description
              "Clear targeted adjacencies.";
            leaf target-address {
              type inet:ip-address;
              description
                "The target address. If this is not provided then
                all targeted adjacencies are cleared";
            }
          } // targeted
        }
        case link {
          container link {
```



```

        presence "Present to clear link adjacencies.";
        description
            "Clear link adjacencies.";
        leaf next-hop-interface {
            type mpls-interface-ref;
            description
                "Interface connecting to next-hop. If this is not
                provided then all link adjacencies are cleared.";
        }
        leaf next-hop-address {
            type inet:ip-address;
            must "../next-hop-interface" {
                description
                    "Applicable when interface is specified.";
            }
            description
                "IP address of next-hop. If this is not provided
                then adjacencies to all next-hops on the given
                interface are cleared.";
        } // next-hop-address
    } // link
}
}
}
}
}

rpc mpls-ldp-clear-peer-statistics {
    description
        "Clears protocol statistics (e.g. sent and received
        counters).";
    input {
        leaf lsr-id {
            type union {
                type yang:dotted-quad;
                type uint32;
            }
            description
                "LSR ID of peer whose statistic are to be cleared.
                If this is not provided then all peers statistics are
                cleared";
        }
    }
}

/*
 * Notifications

```

```
*/
notification mpls-ldp-peer-event {
    description
        "Notification event for a change of LDP peer operational
        status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    uses ldp-peer-ref;
}

notification mpls-ldp-hello-adjacency-event {
    description
        "Notification event for a change of LDP adjacency operational
        status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    uses ldp-adjacency-ref;
}

notification mpls-ldp-fec-event {
    description
        "Notification event for a change of FEC status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    uses ldp-fec-event;
}

notification mpls-mldp-fec-event {
    description
        "Notification event for a change of FEC status.";
    leaf event-type {
        type oper-status-event-type;
        description "Event type.";
    }
    uses mldp-fec-event;
}
}

<CODE ENDS>
```

Figure 22

7. Security Considerations

The configuration, state, action and notification data defined using YANG data models in this document are likely to be accessed via the protocols such as NETCONF [RFC6241] etc.

Hence, YANG implementations MUST comply with the security requirements specified in section 15 of [RFC6020]. Additionally, NETCONF implementations MUST comply with the security requirements specified in sections 2.2, 2.3 and 9 of [RFC6241] as well as section 3.7 of [RFC6536].

8. IANA Considerations

This document does not extend LDP or mLDP base protocol specification and hence there are no IANA considerations.

Note to the RFC Editor: Please remove IANA section before the publication.

9. Acknowledgments

The authors would like to acknowledge Eddie Chami, Nagendra Kumar, Mannan Venkatesan, Pavan Beeram for their contribution to this document. We also acknowledge Ladislav Lhotka for his useful comments as the YANG Doctor.

10. References

10.1. Normative References

- [I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.
- [I-D.rtgyangdt-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., and D. Bogdanovic, "Network Instance Model", draft-rtgyangdt-rtgwg-ni-model-00 (work in progress), May 2016.
- [I-D.saad-mpls-base-yang]
Raza, K., Gandhi, R., Liu, X., Beeram, V., Saad, T., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for MPLS Base", draft-saad-mpls-base-yang-00 (work in progress), May 2016.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, DOI 10.17487/RFC3478, February 2003, <<http://www.rfc-editor.org/info/rfc3478>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<http://www.rfc-editor.org/info/rfc5036>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<http://www.rfc-editor.org/info/rfc5331>>.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, DOI 10.17487/RFC5561, July 2009, <<http://www.rfc-editor.org/info/rfc5561>>.
- [RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, DOI 10.17487/RFC5918, August 2010, <<http://www.rfc-editor.org/info/rfc5918>>.
- [RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas, "Signaling LDP Label Advertisement Completion", RFC 5919, DOI 10.17487/RFC5919, August 2010, <<http://www.rfc-editor.org/info/rfc5919>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.

- [RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <<http://www.rfc-editor.org/info/rfc6388>>.
- [RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, DOI 10.17487/RFC6389, November 2011, <<http://www.rfc-editor.org/info/rfc6389>>.
- [RFC6512] Wijnands, IJ., Rosen, E., Napierala, M., and N. Leymann, "Using Multipoint LDP When the Backbone Has No Route to the Root", RFC 6512, DOI 10.17487/RFC6512, February 2012, <<http://www.rfc-editor.org/info/rfc6512>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC6826] Wijnands, IJ., Ed., Eckert, T., Leymann, N., and M. Napierala, "Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6826, DOI 10.17487/RFC6826, January 2013, <<http://www.rfc-editor.org/info/rfc6826>>.
- [RFC7060] Napierala, M., Rosen, E., and IJ. Wijnands, "Using LDP Multipoint Extensions on Targeted LDP Sessions", RFC 7060, DOI 10.17487/RFC7060, November 2013, <<http://www.rfc-editor.org/info/rfc7060>>.
- [RFC7140] Jin, L., Jounay, F., Wijnands, IJ., and N. Leymann, "LDP Extensions for Hub and Spoke Multipoint Label Switched Path", RFC 7140, DOI 10.17487/RFC7140, March 2014, <<http://www.rfc-editor.org/info/rfc7140>>.
- [RFC7246] Wijnands, IJ., Ed., Hitchen, P., Leymann, N., Henderickx, W., Gulko, A., and J. Tantsura, "Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context", RFC 7246, DOI 10.17487/RFC7246, June 2014, <<http://www.rfc-editor.org/info/rfc7246>>.
- [RFC7438] Wijnands, IJ., Ed., Rosen, E., Gulko, A., Joerde, U., and J. Tantsura, "Multipoint LDP (mLDP) In-Band Signaling with Wildcards", RFC 7438, DOI 10.17487/RFC7438, January 2015, <<http://www.rfc-editor.org/info/rfc7438>>.

- [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <<http://www.rfc-editor.org/info/rfc7552>>.
- [RFC7715] Wijnands, IJ., Ed., Raza, K., Atlas, A., Tantsura, J., and Q. Zhao, "Multipoint LDP (mLDP) Node Protection", RFC 7715, DOI 10.17487/RFC7715, January 2016, <<http://www.rfc-editor.org/info/rfc7715>>.

10.2. Informative References

- [I-D.ietf-rtgwg-policy-model]
Shaikh, A., Shakir, R., D'Souza, K., and C. Chase,
"Routing Policy Configuration Model for Service Provider
Networks", draft-ietf-rtgwg-policy-model-01 (work in
progress), April 2016.
- [I-D.iwijnand-mpls-mlbp-multi-topology]
Wijnands, I. and K. Raza, "mLDP Extensions for Multi
Topology Routing", draft-iwijnand-mpls-mlbp-multi-
topology-03 (work in progress), June 2013.
- [I-D.openconfig-netmod-opstate]
Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling
of Operational State Data in YANG", draft-openconfig-
netmod-opstate-01 (work in progress), July 2015.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private
Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February
2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC7307] Zhao, Q., Raza, K., Zhou, C., Fang, L., Li, L., and D.
King, "LDP Extensions for Multi-Topology", RFC 7307,
DOI 10.17487/RFC7307, July 2014,
<<http://www.rfc-editor.org/info/rfc7307>>.

Appendix A. Additional Contributors

Stephane Litkowski
Orange.
Email: stephane.litkowski@orange.com

Reshad Rahman
Cisco Systems Inc.
Email: rrahman@cisco.com

Danial Johari
Cisco Systems Inc.
Email: dajohari@cisco.com

Authors' Addresses

Kamran Raza
Cisco Systems, Inc.
Email: skraza@cisco.com

Rajiv Asati
Cisco Systems, Inc.
Email: rajiva@cisco.com

Sowmya Krishnaswamy
Cisco Systems, Inc.
Email: sowkrish@cisco.com

Xufeng Liu
Ericsson
Email: xliu@kuatrotech.com

Jeff Tantsura
Ericsson
Email: jeff.tantsura@ericsson.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Xia Chen
Huawei Technologies
Email: jescia.chenxia@huawei.com

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

Matthew Bocci
Alcatel-Lucent
Email: matthew.bocci@alcatel-lucent.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 21, 2020

K. Raza, Ed.
R. Asati
Cisco Systems

X. Liu
Volta Networks

S. Esale
Juniper Networks

X. Chen
Huawei Technologies

H. Shah
Ciena Corporation

March 20, 2020

YANG Data Model for MPLS LDP
draft-ietf-mpls-ldp-yang-09

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP). The model also serves as the base model to define Multipoint LDP (mLDP) model.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 21, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Base and Extended	3
2. Specification of Requirements	4
3. Overview	4
4. The Complete Tree	7
5. Configuration	16
5.1. Configuration Hierarchy	19
5.1.1. Global parameters	20
5.1.2. Capabilities parameters	20
5.1.3. Per-Address-Family parameters	20
5.1.4. Hello Discovery parameters	20
5.1.5. Peer parameters	21
5.1.6. Forwarding parameters	21
6. Operational State	22
6.1. Adjacency state	22
6.2. Peer state	23
6.3. Bindings state	24
6.4. Capabilities state	26
7. Notifications	27
8. Action	27
9. YANG Specification	27
9.1. Base	27
9.2. Extended	59
10. Security Considerations	80
10.1. YANG model	80
10.1.1. Writable nodes	81
10.1.2. Readable nodes	81
10.1.3. RPC operations	82
10.1.4. Notifications	83
11. IANA Considerations	83
12. Acknowledgments	83
13. Contributors	84
14. Normative References	84
15. Informative References	87
Appendix A. Data Tree Example	88
Authors' Addresses	92

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is one of the network management protocols that defines mechanisms to manage network devices. YANG [RFC6020] [RFC7950] is a modular language that represents data structures in an XML tree format, and is used as a data modelling language for the NETCONF.

This document introduces a YANG data model for MPLS Label Distribution Protocol (LDP) [RFC5036]. This model also covers LDP IPv6 [RFC7552] and LDP capabilities [RFC5561] specifications.

The data model is defined for the following constructs that are used for managing the protocol:

- * Configuration
- * Operational State
- * Executables (Actions)
- * Notifications

This document is organized to define the data model for each of the above constructs in the sequence as listed above.

1.1. Base and Extended

The configuration and state items are divided into the following two broad categories:

- * Base
- * Extended

The "base" category contains the basic and fundamental features that are covered in LDP base specification [RFC5036] and constitute the minimum requirements for a typical base LDP deployment. Whereas, the "extended" category contains other non-base features. All the items in a base category are mandatory and hence no "if-feature" is allowed under the "base" category. The base and extended categories are defined in their own modules as described later.

The example of base feature includes the configuration of LDP lsr-id, enabling LDP interfaces, setting password for LDP session etc., whereas the examples of extended feature include inbound/outbound label policies, igp sync [RFC5443], downstream-on-demand etc. It is

worth highlighting that LDP IPv6 [RFC7552] is also categorized as an extended feature.

While "base" model support will suffice for small deployments, it is expected that large deployments will require both the "base" and "extended" models support from the vendors.

2. Specification of Requirements

In this document, the word "IP" is used to refer to both IPv4 and IPv6, unless otherwise explicitly stated. For example, "IP address family" should be read as "IPv4 and/or IPv6 address family".

3. Overview

This document defines two new modules for LDP YANG support:

- * "ietf-mpls-ldp" module that specifies the base LDP features and augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol defined in [RFC8349]. We define new identity 'mpls-ldp' for LDP and the model allows only a single instance of 'mpls-ldp'.
- * "ietf-mpls-ldp-extended" module that specifies the extended LDP features and augments the base LDP module.

It is to be noted that mLDP YANG model [I-D.ietf-mpls-mldp-yang] augments LDP base and extended modules to specify the mLDP specific base and extended features.

There are four types of containers in our module(s):

- * Read-Write parameters for configuration (Section 5)
- * Read-only parameters for operational state (Section 6)
- * Notifications for events (Section 7)
- * RPCs for executing commands to perform some action (Section 8)

The modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407]. When protocol states are retrieved from the NMDA operational state datastore, the returned states cover all "config true" (rw) and "config false" (ro) nodes defined in the schema.

Following diagram depicts high level LDP YANG tree organization and hierarchy:

```

+-- rw routing
  +-- rw control-plane-protocols
    +-- rw control-plane-protocol
      +-- rw mpls-ldp
        +-- rw ...
          +-- rw ... // base
          |   +-- rw ...
          |   +-- ro ...
          |   +--
          +-- ro ...
          |   +-- ro ...
          |   +-- ro ...
          |   +--
          +-- rw ldp-ext: .... // extended
          |   +-- rw ...
          |   +-- ro ...
          |   +--
          +-- ro ...
          |   +-- ro ...
          |   +-- ro ...

```

rpcs:

```

+-- x mpls-ldp-some_action
+-- x . . . . .

```

notifications:

```

+--- n mpls-ldp-some_event
+--- n ...

```

Figure 1: LDP YANG tree organization

Before going into data model details, it is important to take note of the following points:

- * This model aims to address only the core LDP parameters as per RFC specification, as well as well-known and widely deployed manageability controls (such as label filtering policies to apply filtering rules on the assignment, advertisement, and acceptance for label bindings). Any vendor specific feature should be defined in a vendor-specific augmentation of this model.
- * Multi-topology LDP [RFC7307] is beyond the scope of this document.

- * This model does not cover any applications running on top of LDP, nor does it cover any OAM procedures for LDP.
- * This model is a VPN Routing and Forwarding (VRF)-centric model. It is important to note that [RFC4364] defines VRF tables and default forwarding tables as different, however from a YANG modelling perspective this introduces unnecessary complications, hence we are treating the default forwarding table as just another VRF.
- * A "network-instance", as defined in [RFC8529], refers to a VRF instance (both default and non-default) within the scope of this model.
- * This model supports two address-families, namely "ipv4" and "ipv6".
- * This model assumes platform-wide label space (i.e. label space Id of zero). However, when Upstream Label assignment [RFC6389] is in use, an upstream assigned label is looked up in a Context-Specific label space as defined in [RFC5331].
- * The label and peer policies (including filters) are defined using prefix-set and neighbor-set respectively as defined in routing-policy model [I-D.ietf-rtgwg-policy-model].
- * This model uses the terms LDP "neighbor"/"adjacency", "session", and "peer" with the following semantics:
 - Neighbor/Adjacency: An LDP enabled LSR that is discovered through LDP discovery mechanisms.
 - Session: An LDP neighbor with whom a TCP connection has been established.
 - Peer: An LDP session which has successfully progressed beyond its initialization phase and is either already exchanging the bindings or is ready to do so.

It is to be noted that LDP Graceful Restart (GR) mechanisms defined in [RFC3478] allow keeping the exchanged bindings for some time after a session goes down with a peer. We call such a state belonging to a "stale" peer -- i.e. keeping peer bindings from a peer with whom currently there is either no connection established or connection is established but GR session is in recovery state. When used in this document, the above terms will refer strictly to the semantics and definitions defined for them.

A simplified graphical tree representation of base and extended LDP YANG data model is presented in Figure 2. The meaning of the symbols in these tree diagrams is defined in [RFC8340].

The actual YANG specification for base and extended modules is captured in Section 9.

While presenting the YANG tree view and actual specification, this document assumes readers' familiarity with the concepts of YANG modeling, its presentation and its compilation.

4. The Complete Tree

Following is a complete tree representation of configuration, state, notification, and RPC items under LDP base and extended modules.

```

module: ietf-mpls-ldp
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol:
      +--rw mpls-ldp
        +--rw global
          +--rw capability
            +--rw ldp-ext:end-of-lib {capability-end-of-lib}?
            |   +--rw ldp-ext:enabled?    boolean
            +--rw ldp-ext:typed-wildcard-fec
            |   {capability-typed-wildcard-fec}?
            |   +--rw ldp-ext:enabled?    boolean
            +--rw ldp-ext:upstream-label-assignment
            |   {capability-upstream-label-assignment}?
            |   +--rw ldp-ext:enabled?    boolean
          +--rw graceful-restart
            +--rw enabled?                boolean
            +--rw reconnect-time?         uint16
            +--rw recovery-time?          uint16
            +--rw forwarding-holdtime?     uint16
            +--rw ldp-ext:helper-enabled?  boolean
            |   {graceful-restart-helper-mode}?
          +--rw lsr-id?
            |   rt-types:router-id
          +--rw address-families
            +--rw ipv4!
            |   +--rw enabled?                boolean
            |   +--ro label-distribution-control-mode?  enumeration
            |   +--ro bindings
            |   |   +--ro address* [address]
            |   |   |   +--ro address          inet:ipv4-address
            |   |   |   +--ro advertisement-type?  advertised-received
            |   |   |   +--ro peer
  
```

```

|         +---ro lsr-id?                leafref
|         +---ro label-space-id?       leafref
+---ro fec-label* [fec]
|         +---ro fec        inet:ipv4-prefix
|         +---ro peer*
|             [lsr-id label-space-id advertisement-type]
|         +---ro lsr-id                leafref
|         +---ro label-space-id        leafref
|         +---ro advertisement-type
|             | advertised-received
|         +---ro label?
|             | rt-types:mpls-label
|         +---ro used-in-forwarding?   boolean
+---rw ldp-ext:label-policy
|   +---rw ldp-ext:advertise
|       | +---rw ldp-ext:egress-explicit-null
|       | | +---rw ldp-ext:enabled?   boolean
|       | +---rw ldp-ext:prefix-list?
|       |     prefix-list-ref
|   +---rw ldp-ext:accept
|       | +---rw ldp-ext:prefix-list?  prefix-list-ref
|   +---rw ldp-ext:assign
|       | {policy-label-assignment-config}?
|       | +---rw ldp-ext:independent-mode
|       | | +---rw ldp-ext:prefix-list?  prefix-list-ref
|       | +---rw ldp-ext:ordered-mode
|       |     {policy-ordered-label-config}?
|       | +---rw ldp-ext:egress-prefix-list?
|       |     prefix-list-ref
+---rw ldp-ext:transport-address?
|   inet:ipv4-address
+---rw ldp-ext:ipv6!
|   +---rw ldp-ext:enabled?
|       | boolean
+---rw ldp-ext:label-policy
|   +---rw ldp-ext:advertise
|       | +---rw ldp-ext:egress-explicit-null
|       | | +---rw ldp-ext:enabled?   boolean
|       | +---rw ldp-ext:prefix-list?
|       |     prefix-list-ref
|   +---rw ldp-ext:accept
|       | +---rw ldp-ext:prefix-list?  prefix-list-ref
|   +---rw ldp-ext:assign
|       | {policy-label-assignment-config}?
|       | +---rw ldp-ext:independent-mode
|       | | +---rw ldp-ext:prefix-list?  prefix-list-ref
|       | +---rw ldp-ext:ordered-mode
|       |     {policy-ordered-label-config}?

```



```

|         +---rw ldp-ext:egress-prefix-list?
|             prefix-list-ref
+---rw ldp-ext:transport-address
|     inet:ipv6-address
+---ro ldp-ext:label-distribution-control-mode?
|     enumeration
+---ro ldp-ext:bindings
|   +---ro ldp-ext:address* [address]
|       +---ro ldp-ext:address
|           |
|           inet:ipv6-address
|       +---ro ldp-ext:advertisement-type?
|           |
|           advertised-received
|       +---ro ldp-ext:peer
|           +---ro ldp-ext:lsr-id?          leafref
|           +---ro ldp-ext:label-space-id? leafref
+---ro ldp-ext:fec-label* [fec]
|   +---ro ldp-ext:fec          inet:ipv6-prefix
|   +---ro ldp-ext:peer*
|       [lsr-id label-space-id advertisement-type]
|       +---ro ldp-ext:lsr-id          leafref
|       +---ro ldp-ext:label-space-id  leafref
|       +---ro ldp-ext:advertisement-type
|           |
|           advertised-received
|       +---ro ldp-ext:label?
|           |
|           rt-types:mpls-label
|       +---ro ldp-ext:used-in-forwarding? boolean
+---rw ldp-ext:forwarding-nexthop
|   {forwarding-nexthop-config}?
+---rw ldp-ext:interfaces
|   +---rw ldp-ext:interface* [name]
|       +---rw ldp-ext:name          if:interface-ref
|       +---rw ldp-ext:address-family* [afi]
|           +---rw ldp-ext:afi          identityref
|           +---rw ldp-ext:ldp-disable? boolean
+---rw ldp-ext:igp-synchronization-delay? uint16
+---rw discovery
|   +---rw interfaces
|       +---rw hello-holdtime?    uint16
|       +---rw hello-interval?    uint16
|       +---rw interface* [name]
|           +---rw name
|               |
|               if:interface-ref
|       +---ro next-hello?          uint16
|       +---rw address-families
|           |
|           +---rw ipv4!
|               |
|               +---rw enabled?      boolean
|               +---ro hello-adjacencies
|                   |
|                   +---ro hello-adjacency* [adjacent-address]

```

```

+--ro adjacent-address
|   inet:ipv4-address
+--ro flag*                               identityref
+--ro hello-holdtime
|   +--ro adjacent?      uint16
|   +--ro negotiated?    uint16
|   +--ro remaining?     uint16
+--ro next-hello?      uint16
+--ro statistics
|   +--ro discontinuity-time
|   |   yang:date-and-time
|   +--ro hello-received?
|   |   yang:counter64
|   +--ro hello-dropped?
|   |   yang:counter64
+--ro peer
|   +--ro lsr-id?        leafref
|   +--ro label-space-id? leafref
+--rw ldp-ext:transport-address? union
+--rw ldp-ext:ipv6!
+--rw ldp-ext:enabled?          boolean
+--ro ldp-ext:hello-adjacencies
|   +--ro ldp-ext:hello-adjacency*
|   |   [adjacent-address]
|   +--ro ldp-ext:adjacent-address
|   |   inet:ipv6-address
|   +--ro ldp-ext:flag*
|   |   identityref
|   +--ro ldp-ext:hello-holdtime
|   |   +--ro ldp-ext:adjacent?      uint16
|   |   +--ro ldp-ext:negotiated?    uint16
|   |   +--ro ldp-ext:remaining?     uint16
|   +--ro ldp-ext:next-hello?      uint16
|   +--ro ldp-ext:statistics
|   |   +--ro ldp-ext:discontinuity-time
|   |   |   yang:date-and-time
|   |   +--ro ldp-ext:hello-received?
|   |   |   yang:counter64
|   |   +--ro ldp-ext:hello-dropped?
|   |   |   yang:counter64
|   +--ro ldp-ext:peer
|   |   +--ro ldp-ext:lsr-id?        leafref
|   |   +--ro ldp-ext:label-space-id? leafref
+--rw ldp-ext:transport-address? union
+--rw ldp-ext:hello-holdtime?      uint16
|   {per-interface-timer-config}?
+--rw ldp-ext:hello-interval?      uint16
|   {per-interface-timer-config}?

```

```

    |         +---rw ldp-ext:igp-synchronization-delay?   uint16
    |         |         {per-interface-timer-config}?
+---rw targeted
    |   +---rw hello-holdtime?      uint16
    |   +---rw hello-interval?     uint16
    |   +---rw hello-accept
    |   |   +---rw enabled?          boolean
    |   |   +---rw ldp-ext:neighbor-list? neighbor-list-ref
    |   |   |         {policy-targeted-discovery-config}?
+---rw address-families
    |   +---rw ipv4!
    |   |   +---ro hello-adjacencies
    |   |   |   +---ro hello-adjacency*
    |   |   |   |   [local-address adjacent-address]
    |   |   |   |   +---ro local-address      inet:ipv4-address
    |   |   |   |   +---ro adjacent-address   inet:ipv4-address
    |   |   |   |   +---ro flag*             identityref
    |   |   |   |   +---ro hello-holdtime
    |   |   |   |   |   +---ro adjacent?      uint16
    |   |   |   |   |   +---ro negotiated?    uint16
    |   |   |   |   |   +---ro remaining?     uint16
    |   |   |   |   +---ro next-hello?        uint16
    |   |   |   |   +---ro statistics
    |   |   |   |   |   +---ro discontinuity-time
    |   |   |   |   |   |   yang:date-and-time
    |   |   |   |   |   +---ro hello-received?
    |   |   |   |   |   |   yang:counter64
    |   |   |   |   |   +---ro hello-dropped?
    |   |   |   |   |   |   yang:counter64
    |   |   |   |   +---ro peer
    |   |   |   |   |   +---ro lsr-id?         leafref
    |   |   |   |   |   +---ro label-space-id? leafref
    |   |   +---rw target* [adjacent-address]
    |   |   |   +---rw adjacent-address   inet:ipv4-address
    |   |   |   +---rw enabled?          boolean
    |   |   |   +---rw local-address?    inet:ipv4-address
+---rw ldp-ext:ipv6!
    |   +---ro ldp-ext:hello-adjacencies
    |   |   +---ro ldp-ext:hello-adjacency*
    |   |   |   [local-address adjacent-address]
    |   |   |   +---ro ldp-ext:local-address
    |   |   |   |   inet:ipv6-address
    |   |   |   +---ro ldp-ext:adjacent-address
    |   |   |   |   inet:ipv6-address
    |   |   |   +---ro ldp-ext:flag*
    |   |   |   |   identityref
    |   |   |   +---ro ldp-ext:hello-holdtime
    |   |   |   |   +---ro ldp-ext:adjacent?    uint16

```

```

|
|
|
|      +---ro ldp-ext:negotiated?      uint16
|      +---ro ldp-ext:remaining?      uint16
|      +---ro ldp-ext:next-hello?      uint16
|      +---ro ldp-ext:statistics
|      |      +---ro ldp-ext:discontinuity-time
|      |      |      yang:date-and-time
|      |      +---ro ldp-ext:hello-received?
|      |      |      yang:counter64
|      |      +---ro ldp-ext:hello-dropped?
|      |      |      yang:counter64
|      +---ro ldp-ext:peer
|      |      +---ro ldp-ext:lsr-id?      leafref
|      |      +---ro ldp-ext:label-space-id?      leafref
+---rw ldp-ext:target* [adjacent-address]
|      +---rw ldp-ext:adjacent-address
|      |      inet:ipv6-address
+---rw ldp-ext:enabled?      boolean
+---rw ldp-ext:local-address?
|      inet:ipv6-address
+---rw peers
+---rw authentication
|      +---rw (authentication-type)?
|      |      +---: (password)
|      |      |      +---rw key?      string
|      |      |      +---rw crypto-algorithm?      identityref
|      |      +---: (ldp-ext:key-chain) {key-chain}?
|      |      |      +---rw ldp-ext:key-chain?      key-chain:key-chain-ref
+---rw session-ka-holdtime?      uint16
+---rw session-ka-interval?      uint16
+---rw peer* [lsr-id label-space-id]
|      +---rw lsr-id      rt-types:router-id
|      +---rw label-space-id      uint16
|      +---rw authentication
|      |      +---rw (authentication-type)?
|      |      |      +---: (password)
|      |      |      |      +---rw key?      string
|      |      |      |      +---rw crypto-algorithm?      identityref
|      |      |      +---: (ldp-ext:key-chain) {key-chain}?
|      |      |      |      +---rw ldp-ext:key-chain?
|      |      |      |      |      key-chain:key-chain-ref
+---rw address-families
|      +---rw ipv4!
|      |      +---ro hello-adjacencies
|      |      |      +---ro hello-adjacency*
|      |      |      |      [local-address adjacent-address]
|      |      |      |      +---ro local-address      inet:ipv4-address
|      |      |      |      +---ro adjacent-address      inet:ipv4-address
|      |      |      +---ro flag*      identityref

```

```

+--ro hello-holdtime
|   +--ro adjacent?      uint16
|   +--ro negotiated?    uint16
|   +--ro remaining?     uint16
+--ro next-hello?        uint16
+--ro statistics
|   +--ro discontinuity-time
|       |   yang:date-and-time
+--ro hello-received?
|       |   yang:counter64
+--ro hello-dropped?
|       |   yang:counter64
+--ro interface?        if:interface-ref
+--rw ldp-ext:label-policy
+--rw ldp-ext:advertise
|   +--rw ldp-ext:prefix-list?    prefix-list-ref
+--rw ldp-ext:accept
|   +--rw ldp-ext:prefix-list?    prefix-list-ref
+--rw ldp-ext:ipv6!
+--ro ldp-ext:hello-adjacencies
|   +--ro ldp-ext:hello-adjacency*
|       |   [local-address adjacent-address]
+--ro ldp-ext:local-address
|       |   inet:ipv6-address
+--ro ldp-ext:adjacent-address
|       |   inet:ipv6-address
+--ro ldp-ext:flag*
|       |   identityref
+--ro ldp-ext:hello-holdtime
|   +--ro ldp-ext:adjacent?      uint16
|   +--ro ldp-ext:negotiated?    uint16
|   +--ro ldp-ext:remaining?     uint16
+--ro ldp-ext:next-hello?        uint16
+--ro ldp-ext:statistics
|   +--ro ldp-ext:discontinuity-time
|       |   yang:date-and-time
+--ro ldp-ext:hello-received?
|       |   yang:counter64
+--ro ldp-ext:hello-dropped?
|       |   yang:counter64
+--ro ldp-ext:interface?
|       |   if:interface-ref
+--rw ldp-ext:label-policy
+--rw ldp-ext:advertise
|   +--rw ldp-ext:prefix-list?    prefix-list-ref
+--rw ldp-ext:accept
|   +--rw ldp-ext:prefix-list?    prefix-list-ref
+--ro label-advertisement-mode

```

```

|   +--ro local?          label-adv-mode
|   +--ro peer?           label-adv-mode
|   +--ro negotiated?     label-adv-mode
+--ro next-keep-alive?    uint16
+--ro received-peer-state
|   +--ro graceful-restart
|   |   +--ro enabled?    boolean
|   |   +--ro reconnect-time?  uint16
|   |   +--ro recovery-time?  uint16
|   +--ro capability
|   |   +--ro end-of-lib
|   |   |   +--ro enabled?  boolean
|   |   +--ro typed-wildcard-fec
|   |   |   +--ro enabled?  boolean
|   |   +--ro upstream-label-assignment
|   |   |   +--ro enabled?  boolean
+--ro session-holdtime
|   +--ro peer?          uint16
|   +--ro negotiated?    uint16
|   +--ro remaining?     uint16
+--ro session-state?     enumeration
+--ro tcp-connection
|   +--ro local-address?  inet:ip-address
|   +--ro local-port?     inet:port-number
|   +--ro remote-address? inet:ip-address
|   +--ro remote-port?    inet:port-number
+--ro up-time?
|   rt-types:timeticks64
+--ro statistics
|   +--ro discontinuity-time  yang:date-and-time
|   +--ro received
|   |   +--ro total-octets?  yang:counter64
|   |   +--ro total-messages? yang:counter64
|   |   +--ro address?       yang:counter64
|   |   +--ro address-withdraw? yang:counter64
|   |   +--ro initialization? yang:counter64
|   |   +--ro keepalive?     yang:counter64
|   |   +--ro label-abort-request? yang:counter64
|   |   +--ro label-mapping? yang:counter64
|   |   +--ro label-release? yang:counter64
|   |   +--ro label-request? yang:counter64
|   |   +--ro label-withdraw? yang:counter64
|   |   +--ro notification?  yang:counter64
|   +--ro sent
|   |   +--ro total-octets?  yang:counter64
|   |   +--ro total-messages? yang:counter64
|   |   +--ro address?       yang:counter64
|   |   +--ro address-withdraw? yang:counter64

```

```

| | | +---ro initialization?          yang:counter64
| | | +---ro keepalive?             yang:counter64
| | | +---ro label-abort-request?    yang:counter64
| | | +---ro label-mapping?         yang:counter64
| | | +---ro label-release?         yang:counter64
| | | +---ro label-request?         yang:counter64
| | | +---ro label-withdraw?        yang:counter64
| | | +---ro notification?          yang:counter64
| | +---ro total-addresses?         uint32
| | +---ro total-labels?            uint32
| | +---ro total-fec-label-bindings? uint32
+---rw ldp-ext:admin-down?          boolean
|   {per-peer-admin-down}?
+---rw ldp-ext:graceful-restart
|   {per-peer-graceful-restart-config}?
+---rw ldp-ext:enabled?             boolean
+---rw ldp-ext:reconnect-time?      uint16
+---rw ldp-ext:recovery-time?       uint16
+---rw ldp-ext:session-ka-holdtime? uint16
|   {per-peer-session-attributes-config}?
+---rw ldp-ext:session-ka-interval? uint16
|   {per-peer-session-attributes-config}?
+---rw ldp-ext:session-downstream-on-demand
|   {session-downstream-on-demand-config}?
+---rw ldp-ext:enabled?             boolean
+---rw ldp-ext:peer-list?           peer-list-ref
+---rw ldp-ext:dual-stack-transport-preference
|   {peers-dual-stack-transport-preference}?
+---rw ldp-ext:max-wait?            uint16
+---rw ldp-ext:prefer-ipv4!
|   +---rw ldp-ext:peer-list?       peer-list-ref

rpccs:
+---x mpls-ldp-clear-peer
|   +---w input
|   |   +---w protocol-name?        leafref
|   |   +---w lsr-id?               leafref
|   |   +---w label-space-id?       leafref
+---x mpls-ldp-clear-hello-adjacency
|   +---w input
|   |   +---w hello-adjacency
|   |   |   +---w protocol-name?    leafref
|   |   |   +---w (hello-adjacency-type)?
|   |   |   |   +---:(targeted)
|   |   |   |   |   +---w targeted!
|   |   |   |   |   +---w target-address?  inet:ip-address
|   |   |   +---:(link)
|   |   +---w link!

```

```

|               +---w next-hop-interface?  leafref
|               +---w next-hop-address?    inet:ip-address
+---x mpls-ldp-clear-peer-statistics
  +---w input
    +---w protocol-name?  leafref
    +---w lsr-id?         leafref
    +---w label-space-id? leafref

notifications:
+---n mpls-ldp-peer-event
|   +--ro event-type?  oper-status-event-type
|   +--ro peer
|       +--ro protocol-name?  leafref
|       +--ro lsr-id?         leafref
|       +--ro label-space-id? leafref
+---n mpls-ldp-hello-adjacency-event
|   +--ro event-type?  oper-status-event-type
|   +--ro protocol-name?  leafref
|   +--ro (hello-adjacency-type)?
|       +--:(targeted)
|           +--ro targeted
|               +--ro target-address?  inet:ip-address
|       +--:(link)
|           +--ro link
|               +--ro next-hop-interface?  if:interface-ref
|               +--ro next-hop-address?    inet:ip-address
+---n mpls-ldp-fec-event
|   +--ro event-type?  oper-status-event-type
|   +--ro protocol-name?  leafref
|   +--ro fec?           inet:ip-prefix

```

Figure 2: Complete Tree

5. Configuration

This specification defines the configuration parameters for base LDP as specified in [RFC5036] and LDP IPv6 [RFC7552]. Moreover, it incorporates provisions to enable LDP Capabilities [RFC5561], and defines some of the most significant and commonly used capabilities such as Typed Wildcard FEC [RFC5918], End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

This model augments /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol that is defined in [RFC8349] and follows NMDA as mentioned earlier.

Following is the high-level configuration organization for base LDP module:

```

augment /rt:routing/rt:control-plane-protocols:
  /rt:control-plane-protocol:
    +-- mpls-ldp
      +-- global
        +-- ...
        +-- ...
        +-- address-families
          +-- ipv4
            +-- . . .
            +-- . . .
        +-- capability
          +-- ...
          +-- ...
      +-- discovery
        +-- interfaces
          +-- ...
          +-- ...
          +-- interface* [interface]
            +-- ...
            +-- address-families
              +-- ipv4
                +-- ...
                +-- ...
        +-- targeted
          +-- ...
          +-- address-families
            +-- ipv4
              +- target* [adjacent-address]
                +- ...
                +- ...
      +-- peers
        +-- ...
        +-- ...
        +-- peer* [lsr-id label-space-id]
          +-- ...
          +-- ...

```

Figure 3: Base Configuration organization

Following is the high-level configuration organization for extended LDP:

```

augment /rt:routing/rt:control-plane-protocols/rt:control-plane-protoc
ol
    +-- mpls-ldp
    |   +-- global
    |   |   +-- ...
    |   |   +-- ...
    |   |   +-- address-families
    |   |   |   +-- ipv4
    |   |   |   |   +-- . . .
    |   |   |   |   +-- . . .
    |   |   |   |   +-- label-policy
    |   |   |   |   |   +-- ...
    |   |   |   |   |   +-- ...
    |   |   |   +-- ipv6
    |   |   |   |   +-- . . .
    |   |   |   |   +-- . . .
    |   |   |   |   +-- label-policy
    |   |   |   |   |   +-- ...
    |   |   |   |   |   +-- ...
    |   |   |   +-- capability
    |   |   |   |   +-- ...
    |   |   |   |   +-- ...
    |   |   +-- discovery
    |   |   |   +-- interfaces
    |   |   |   |   +-- ...
    |   |   |   |   +-- ...
    |   |   |   |   +-- interface* [interface]
    |   |   |   |   |   +-- ...
    |   |   |   |   |   +-- address-families
    |   |   |   |   |   |   +-- ipv4
    |   |   |   |   |   |   |   +-- ...
    |   |   |   |   |   |   |   +-- ...
    |   |   |   |   |   +-- ipv6
    |   |   |   |   |   |   +-- ...
    |   |   |   |   |   |   +-- ...
    |   |   |   +-- targeted
    |   |   |   |   +-- ...
    |   |   |   |   +-- address-families
    |   |   |   |   |   +-- ipv6
    |   |   |   |   |   |   +-- target* [adjacent-address]
    |   |   |   |   |   |   |   +-- ...
    |   |   |   |   |   |   |   +-- ...
    |   +-- forwarding-nextthop
    |   |   +-- ...
    |   |   +-- ...
    +-- peers
    |   +-- ...
    |   +-- ...
    +-- peer*

```

```

+-- ...
+-- ...
+-- label-policy
|   +-- ..
+-- address-families
    +-- ipv4
    |   +-- ...
    +-- ipv6
        +-- ...

```

Figure 4: Extended Configuration organization

Given the configuration hierarchy, the model allows inheritance such that an item in a child tree is able to derive value from a similar or related item in one of the parents. For instance, hello holdtime can be configured per-VRF or per-VRF-interface, thus allowing inheritance as well flexibility to override with a different value at any child level.

5.1. Configuration Hierarchy

LDP module resides under a network-instance and the scope of any LDP configuration defined under this tree is per network-instance (per-VRF). This configuration is further divided into sub categories as follows.

- * Global parameters
- * Per-address-family parameters
- * LDP Capabilities parameters
- * Hello Discovery parameters
 - interfaces
 - o Global
 - o Per-interface: Global
 - o Per-interface: Per-address-family
 - targeted
 - o Global

- o Per-address-family: Per-target
- * Peer parameters
 - Global
 - Per-peer: Global
 - Per-peer: Per-address-family

- * Forwarding parameters

Following subsections briefly explain these configuration areas.

5.1.1. Global parameters

There are configuration items that are available directly under a VRF instance and do not fall under any other sub tree. Example of such a parameter is LDP LSR Id that is typically configured per VRF. To keep legacy LDP features and applications working in an LDP IPv4 networks with this model, this document recommends an operator to pick a routable IPv4 unicast address (within a routing domain) as an LSR Id.

5.1.2. Capabilities parameters

This container falls under the global tree and holds the LDP capabilities that are to be enabled for certain features. By default, an LDP capability is disabled unless explicitly enabled. These capabilities are typically used to negotiate with LDP peer(s) the support/non-support related to a feature and its parameters. The scope of a capability enabled under this container applies to all LDP peers in the given VRF instance. There is also a peer level capability container that is provided to override a capability that is enabled/specified at VRF level.

5.1.3. Per-Address-Family parameters

Any LDP configuration parameter related to IP address family (AF) whose scope is VRF wide is configured under this tree. The examples of per-AF parameters include enabling LDP for an address family, prefix-list based label policies, and LDP transport address.

5.1.4. Hello Discovery parameters

This container is used to hold LDP configuration related to Hello and discovery process for both basic (link) and extended (targeted) discovery.

The "interfaces" is a container to configure parameters related to VRF interfaces. There are parameters that apply to all interfaces (such as hello timers), as well as parameters that can be configured per-interface. Hence, an interface list is defined under "interfaces" container. The model defines parameters to configure per-interface non AF related items, as well as per-interface per-AF items. The example of the former is interface hello timers, and example of the latter is enabling hellos for a given AF under an interface.

The "targeted" container under a VRF instance allows to configure LDP targeted discovery related parameters. Within this container, the "target" list provides a means to configure multiple target addresses to perform extended discovery to a specific destination target, as well as to fine-tune the per-target parameters.

5.1.5. Peer parameters

This container is used to hold LDP configuration related to LDP sessions and peers under a VRF instance. This container allows to configure parameters that either apply on VRF's all peers or a subset (peer-list) of VRF peers. The example of such parameters include authentication password, session KA timers etc. Moreover, the model also allows per-peer parameter tuning by specifying a "peer" list under the "peers" container. A peer is uniquely identified by its LSR Id.

Like per-interface parameters, some per-peer parameters are AF-agnostic (i.e. either non AF related or apply to both IP address families), and some that belong to an AF. The example of the former is per-peer session password configuration, whereas the example of the latter is prefix-list based label policies (inbound and outbound) that apply to a given peer.

5.1.6. Forwarding parameters

This container is used to hold configuration used to control LDP forwarding behavior under a VRF instance. One example of a configuration under this container is when a user wishes to enable neighbor discovery on an interface but wishes to disable use of the same interface as forwarding nexthop. This example configuration makes sense only when there are more than one LDP enabled interfaces towards the neighbor.

6. Operational State

Operational state of LDP can be queried and obtained from read-only state containers that fall under the same tree (/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol) as the configuration.

Following are main areas for which LDP operational state is defined:

- * Neighbor Adjacencies
- * Peer
- * Bindings (FEC-label and address)
- * Capabilities

6.1. Adjacency state

Neighbor adjacencies are per address-family hello adjacencies that are formed with neighbors as result of LDP basic or extended discovery. In terms of organization, there is a source of discovery (e.g. interface or target address) along with its associated parameters and one or more discovered neighbors along with neighbor discovery related parameters. For the basic discovery, there could be more than one discovered neighbor for a given source (interface), whereas there is at most one discovered neighbor for an extended discovery source (local-address and target-address). It is also to be noted that the reason for a targeted neighbor adjacency could be either an active source (locally configured targeted) or passive source (to allow any incoming extended/targeted hellos). A neighbor/adjacency record also contains session-state that helps highlight whether a given adjacency has progressed to subsequent session level or to eventual peer level.

Following captures high level tree hierarchy for neighbor adjacency state. The tree is shown for ipv4 address-family only; a similar tree exists for ipv6 address-family as well.

```

+--rw mpls-ldp!
  +--rw discovery
    +--rw interfaces
      +--rw interface* [interface]
        +--rw address-families
          +--rw ipv4
            +--ro hello-adjacencies
              +--ro hello-adjacencies* [adjacent-address]
                +--ro adjacent-address
                  . . . .
            +--rw targeted
              +--rw address-families
                +--rw ipv4
                  +--ro hello-adjacencies
                    +--ro hello-adjacencies*
                      | [local-address adjacent-address]
                    +--ro local-address
                      +--ro adjacent-address
                        . . . .
                        . . . .

```

Figure 5: Adjacency state

6.2. Peer state

Peer related state is presented under peers tree. This is one of the core state that provides info on the session related parameters (mode, authentication, KA timeout etc.), TCP connection info, hello adjacencies for the peer, statistics related to messages and bindings, and capabilities exchange info.

Following captures high level tree hierarchy for peer state. The peer's hello adjacencies tree is shown for ipv4 address-family only; a similar tree exists for ipv6 address-family as well.

```

+--rw mpls-ldp!
  +--rw peers
    +--rw peer* [lsr-id label-space-id]
      +--rw lsr-id
      +--rw label-space-id
      +--ro label-advertisement-mode
      +--ro session-state
      +--ro tcp-connection
      +--ro session-holdtime?
      +--ro up-time
      +-- . . . .
      +--ro address-families
        +--ro ipv4
          +--ro hello-adjacencies
            +--ro hello-adjacencies*
              [local-address adjacent-address]
              . . . .
              . . . .
      +--ro received-peer-state
        +--ro . . . .
        +--ro capability
          +--ro . . . .
      +--ro statistics
        +-- . . . .
        +-- received
          +-- ...
        +-- sent
          +-- ...

```

Figure 6: Peer state

6.3. Bindings state

Binding state provides information on LDP FEC-label bindings as well as address binding for both inbound (received) as well as outbound (advertised) direction. FEC-label bindings are presented as a FEC-centric view, and address bindings are presented as an address-centric view:


```

FEC-Label bindings:
  FEC 203.0.113.1/32:
    advertised: local-label 16000
      peer 192.0.2.1:0
      peer 192.0.2.2:0
      peer 192.0.2.3:0
    received:
      peer 192.0.2.1:0, label 16002, used-in-forwarding=Yes
      peer 192.0.2.2:0, label 17002, used-in-forwarding=No
  FEC 203.0.113.2/32:
    . . . .
  FEC 198.51.100.0/24:
    . . . .
  FEC 2001:db8:0:2::
    . . . .
  FEC 2001:db8:0:3::
    . . . .

Address bindings:
  Addr 192.0.2.10:
    advertised
  Addr 2001:db8:0:10::
    advertised

  Addr 192.0.2.1:
    received, peer 192.0.2.1:0
  Addr 192.0.2.2:
    received, peer 192.0.2.2:0
  Addr 192.0.2.3:
    received, peer 192.0.2.3:0
  Addr 2001:db8:0:2::
    received, peer 192.0.2.2:0
  Addr 2001:db8:0:3::
    received, peer 192.0.2.3:0

```

Figure 7: Example Bindings

Note that all local addresses are advertised to all peers and hence no need to provide per-peer information for local address advertisement. Furthermore, note that it is easy to derive a peer-centric view for the bindings from the information already provided in this model.

Following captures high level tree hierarchy for bindings state. The tree shown below is for ipv4 address-family only; a similar tree exists for ipv6 address-family as well.

```

+--rw mpls-ldp!
  +--rw global
    +--rw address-families
      +--rw ipv4
        +--ro bindings
          +--ro address* [address]
            +--ro address (ipv4-address or ipv6-address)
            +--ro advertisement-type? advertised-received
            +--ro peer? leafref
          +--ro fec-label* [fec]
            +--ro fec (ipv4-prefix or ipv6-prefix)
            +--ro peer* [peer advertisement-type]
              +--ro peer leafref
              +--ro advertisement-type? advertised-received
              +--ro label? mpls:mpls-label
              +--ro used-in-forwarding? boolean

```

Figure 8: Bindings state

6.4. Capabilities state

LDP capabilities state comprise two types of information - global information (such as timer etc.), and per-peer information.

Following captures high level tree hierarchy for LDP capabilities state.

```

+--rw mpls-ldp!
  +--rw peers
    +--rw peer* [lsr-id label-space-id]
      +--rw lsr-id yang:dotted-quad
      +--rw label-space-id
      +--ro received-peer-state
        +--ro capability
          +--ro . . . .
          +--ro . . . .

```

Figure 9: Capabilities state

7. Notifications

This model defines a list of notifications to inform client of important events detected during the protocol operation. These events include events related to changes in the operational state of an LDP peer, hello adjacency, and FEC etc. It is to be noted that an LDP FEC is treated as operational (up) as long as it has at least 1 NHLFE (Next Hop Label Forwarding Entry) with outgoing label.

A simplified graphical representation of the data model for LDP notifications is shown in Figure 2.

8. Action

This model defines a list of rpcs that allow performing an action or executing a command on the protocol. For example, it allows to clear (reset) LDP peers, hello-adjacencies, and statistics. The model makes an effort to provide different level of control so that a user is able to either clear all, or clear all for a given type, or clear a specific entity.

A simplified graphical representation of the data model for LDP actions is shown in Figure 2.

9. YANG Specification

Following sections specify the actual YANG (module) specification for LDP constructs defined earlier in the document.

9.1. Base

This YANG module imports types defined in [RFC6991], [RFC8349], [RFC8294], [RFC8343], and [RFC8344].

```
<CODE BEGINS> file "ietf-mpls-ldp@2020-02-25.yang"
```

```
// RFC Editor: replace the above date 2020-02-25 with the date of  
// publication and remove this note.
```

```
module ietf-mpls-ldp {  
  yang-version 1.1;  
  
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp";  
  prefix "ldp";  
  
  import ietf-inet-types {
```

```
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
}

import ietf-routing {
    prefix "rt";
    reference
        "RFC 8349: A YANG Data Model for Routing Management (NMDA
        version)";
}

import ietf-routing-types {
    prefix "rt-types";
    reference
        "RFC 8294: Common YANG Data Types for the Routing Area";
}

import ietf-interfaces {
    prefix "if";
    reference "RFC 8343: A YANG Data Model for Interface Management";
}

import ietf-ip {
    prefix "ip";
    reference "RFC 7277: A YANG Data Model for IP Management";
}

import ietf-key-chain {
    prefix "key-chain";
    reference "RFC 8177: YANG Data Model for Key Chains";
}

organization
    "IETF MPLS Working Group";
contact
    "WG Web:  <http://tools.ietf.org/wg/mpls/>
    WG List:  <mailto:mpls@ietf.org>

    Editor:   Kamran Raza
              <mailto:skraza@cisco.com>

    Editor:   Rajiv Asati
              <mailto:rajiva@cisco.com>
```

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Santosh Esale
<mailto:sesale@juniper.net>

Editor: Xia Chen
<mailto:jescia.chenxia@huawei.com>

Editor: Himanshu Shah
<mailto:hshah@ciena.com>;

description

"This YANG module defines the essential components for the management of Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP). It is also the base model to be augmented for Multipoint LDP (mLDP).

Copyright (c) 2020 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

```
revision 2020-02-25 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Data Model for MPLS LDP.";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}

/*
 * Typedefs
 */
typedef advertised-received {
  type enumeration {
```

```
    enum advertised {
      description "Advertised information.";
    }
    enum received {
      description "Received information.";
    }
  }
  description
    "Received or advertised.";
}

typedef downstream-upstream {
  type enumeration {
    enum downstream {
      description "Downstream information.";
    }
    enum upstream {
      description "Upstream information.";
    }
  }
  description
    "Downstream or upstream.";
}

typedef label-adv-mode {
  type enumeration {
    enum downstream-unsolicited {
      description "Downstream Unsolicited.";
    }
    enum downstream-on-demand {
      description "Downstream on Demand.";
    }
  }
  description
    "Label Advertisement Mode.";
}

typedef oper-status-event-type {
  type enumeration {
    enum up {
      value 1;
      description
        "Operational status changed to up.";
    }
    enum down {
      value 2;
      description
        "Operational status changed to down.";
    }
  }
}
```

```
    }
  }
  description "Operational status event type for notifications.";
}

/*
 * Identities
 */
identity mpls-ldp {
  base rt:control-plane-protocol;
  description
    "LDP protocol.";
  reference
    "RFC 5036: LDP Specification";
}

identity adjacency-flag-base {
  description "Base type for adjacency flags.";
}

identity adjacency-flag-active {
  base adjacency-flag-base;
  description
    "This adjacency is configured and actively created.";
}

identity adjacency-flag-passive {
  base adjacency-flag-base;
  description
    "This adjacency is not configured and passively accepted.";
}

/*
 * Groupings
 */
grouping adjacency-state-attributes {
  description
    "The operational state attributes of an LDP Hello adjacency,
    which can used for basic and extended discoveris, in IPv4 and
    IPv6 address families.";

  leaf-list flag {
    type identityref {
      base adjacency-flag-base;
    }
    description
      "On or more flags to indicate whether the adjacency is
```

```
        actively created, passively accepted, or both.";
    }
    container hello-holdtime {
        description
            "Containing Hello holdtime state information.";
        leaf adjacent {
            type uint16;
            units seconds;
            description
                "The holdtime value learned from the adjacent LSR.";
        }
        leaf negotiated {
            type uint16;
            units seconds;
            description
                "The holdtime negotiated between this LSR and the adjacent
                LSR.";
        }
        leaf remaining {
            type uint16;
            units seconds;
            description
                "The time remaining until the holdtime timer expires.";
        }
    }

    leaf next-hello {
        type uint16;
        units seconds;
        description
            "The time when the next Hello message will be sent.";
    }

    container statistics {
        description
            "Statistics objects.";

        leaf discontinuity-time {
            type yang:date-and-time;
            mandatory true;
            description
                "The time on the most recent occasion at which any one or
                more of this interface's counters suffered a
                discontinuity.  If no such discontinuities have occurred
                since the last re-initialization of the local management
                subsystem, then this node contains the time the local
                management subsystem re-initialized itself.";
        }
    }
}
```



```
    leaf hello-received {
      type yang:counter64;
      description
        "The number of Hello messages received.";
    }
    leaf hello-dropped {
      type yang:counter64;
      description
        "The number of Hello messages dropped.";
    }
  } // statistics
} // adjacency-state-attributes

grouping basic-discovery-timers {
  description
    "The timer attributes for basic discovery, used in the
    per-interface setting and in the all-interface setting.";

  leaf hello-holdtime {
    type uint16 {
      range 15..3600;
    }
    units seconds;
    description
      "The time interval for which a LDP link Hello adjacency
      is maintained in the absence of link Hello messages from
      the LDP neighbor.
      This leaf may be configured at the per-interface level or
      the global level, with precedence given to the value at the
      per-interface level. If the leaf is not configured at
      either level, the default value at the global level is
      used.";
  }
  leaf hello-interval {
    type uint16 {
      range 5..1200;
    }
    units seconds;
    description
      "The interval between consecutive LDP link Hello messages
      used in basic LDP discovery.
      This leaf may be configured at the per-interface level or
      the global level, with precedence given to the value at the
      per-interface level. If the leaf is not configured at
      either level, the default value at the global level is
      used.";
  }
} // basic-discovery-timers
```

```
grouping binding-address-state-attributes {
  description
    "Operational state attributes of an address binding, used in
    IPv4 and IPv6 address families.";

  leaf advertisement-type {
    type advertised-received;
    description
      "Received or advertised.";
  }
  container peer {
    when "../advertisement-type = 'received'" {
      description
        "Applicable for received address.";
    }
    description
      "LDP peer from which this address is received.";
    uses ldp-peer-ref-from-binding;
  }
} // binding-address-state-attributes

grouping binding-label-state-attributes {
  description
    "Operational state attributes for a FEC-label binding, used in
    IPv4 and IPv6 address families.";

  list peer {
    key "lsr-id label-space-id advertisement-type";
    description
      "List of advertised and received peers.";
    uses ldp-peer-ref-from-binding {
      description
        "The LDP peer from which this binding is received, or to
        which this binding is advertised.
        The peer is identified by its LDP ID, which consists of
        the LSR ID and the Label Space ID.";
    }
    leaf advertisement-type {
      type advertised-received;
      description
        "Received or advertised.";
    }
    leaf label {
      type rt-types:mpls-label;
      description
        "Advertised (outbound) or received (inbound)
        label.";
    }
  }
}
```

```
    leaf used-in-forwarding {
        type boolean;
        description
            "'true' if the label is used in forwarding.";
    }
} // peer
} // binding-label-state-attributes

grouping graceful-restart-attributes-per-peer {
    description
        "Per peer graceful restart attributes.
        On the local side, these attributes are configuration and
        operational state data. On the peer side, these attributes
        are operational state data received from the peer.";

    container graceful-restart {
        description
            "Attributes for graceful restart.";
        leaf enabled {
            type boolean;
            description
                "Enable or disable graceful restart.
                This leaf may be configured at the per-peer level or the
                global level, with precedence given to the value at the
                per-peer level. If the leaf is not configured at either
                level, the default value at the global level is used.";
        }
        leaf reconnect-time {
            type uint16 {
                range 10..1800;
            }
            units seconds;
            description
                "Specifies the time interval that the remote LDP peer
                must wait for the local LDP peer to reconnect after the
                remote peer detects the LDP communication failure.
                This leaf may be configured at the per-peer level or the
                global level, with precedence given to the value at the
                per-peer level. If the leaf is not configured at either
                level, the default value at the global level is used.";
        }
        leaf recovery-time {
            type uint16 {
                range 30..3600;
            }
            units seconds;
            description
                "Specifies the time interval, in seconds, that the remote
```

```
        LDP peer preserves its MPLS forwarding state after
        receiving the Initialization message from the restarted
        local LDP peer.
        This leaf may be configured at the per-peer level or the
        global level, with precedence given to the value at the
        per-peer level. If the leaf is not configured at either
        level, the default value at the global level is used.";
    }
} // graceful-restart
} // graceful-restart-attributes-per-peer

grouping ldp-interface-ref {
    description
        "Defining a reference to LDP interface.";

    leaf name {
        type if:interface-ref;
        must "(/if:interfaces/if:interface[if:name=current()]/ip:ipv4)"
            + " or "
            + "(/if:interfaces/if:interface[if:name=current()]/ip:ipv6)"
        {
            description "Interface is IPv4 or IPv6.";
        }
        description
            "The name of an LDP interface.";
    }
}

grouping ldp-peer-ref-absolute {
    description
        "An absolute reference to an LDP peer, by the LDP ID, which
        consists of the LSR ID and the Label Space ID.";

    leaf protocol-name {
        type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
        }
        description
            "The name of the LDP protocol instance.";
    }
    leaf lsr-id {
        type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol"
                + "[rt:name=current()]/../protocol-name/"
                + "ldp:mpls-ldp/ldp:peers/ldp:peer/ldp:lsr-id";
        }
    }
}
```

```
        description
            "The LSR ID of the peer, as a portion of the peer LDP ID.";
    }
    leaf label-space-id {
        type leafref {
            path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol"
                + "[rt:name=current()/../protocol-name]/"
                + "ldp:mpls-ldp/ldp:peers/"
                + "ldp:peer[ldp:lsr-id=current()/../lsr-id]/"
                + "ldp:label-space-id";
        }
        description
            "The Label Space ID of the peer, as a portion of the peer
            LDP ID.";
    }
} // ldp-peer-ref-absolute

grouping ldp-peer-ref-from-binding {
    description
        "A relative reference to an LDP peer, by the LDP ID, which
        consists of the LSR ID and the Label Space ID.";

    leaf lsr-id {
        type leafref {
            path "../../../../../ldp:peers/ldp:peer/ldp:lsr-id";
        }
        description
            "The LSR ID of the peer, as a portion of the peer LDP ID.";
    }
    leaf label-space-id {
        type leafref {
            path "../../../../ldp:peers/"
                + "ldp:peer[ldp:lsr-id=current()/../lsr-id]/"
                + "ldp:label-space-id";
        }
        description
            "The Label Space ID of the peer, as a portion of the peer
            LDP ID.";
    }
} // ldp-peer-ref-from-binding

grouping ldp-peer-ref-from-interface {
    description
        "A relative reference to an LDP peer, by the LDP ID, which
        consists of the LSR ID and the Label Space ID.";

    container peer {
```

```

description
  "Reference to an LDP peer, by the LDP ID, which consists of
  the LSR ID and the Label Space ID.";
leaf lsr-id {
  type leafref {
    path "../../../../../ldp:peers/ldp:peer/"
      + "ldp:lsr-id";
  }
  description
    "The LSR ID of the peer, as a portion of the peer LDP ID.";
}
leaf label-space-id {
  type leafref {
    path "../../../../../ldp:peers/"
      + "ldp:peer[ldp:lsr-id=current()/../lsr-id]/"
      + "ldp:label-space-id";
  }
  description
    "The Label Space ID of the peer, as a portion of the peer
    LDP ID.";
}
} // peer
} // ldp-peer-ref-from-interface

grouping ldp-peer-ref-from-target {
  description
    "A relative reference to an LDP peer, by the LDP ID, which
    consists of the LSR ID and the Label Space ID.";

  container peer {
    description
      "Reference to an LDP peer, by the LDP ID, which consists of
      the LSR ID and the Label Space ID.";
    leaf lsr-id {
      type leafref {
        path "../../../../../ldp:peers/ldp:peer/"
          + "ldp:lsr-id";
      }
      description
        "The LSR ID of the peer, as a portion of the peer LDP ID.";
    }
    leaf label-space-id {
      type leafref {
        path "../../../../../ldp:peers/"
          + "ldp:peer[ldp:lsr-id=current()/../lsr-id]/"
          + "ldp:label-space-id";
      }
      description

```

```
        "The Label Space ID of the peer, as a portion of the peer
        LDP ID.";
    }
} // peer
} // ldp-peer-ref-from-target

grouping peer-attributes {
    description
        "Peer configuration attributes, used in the per-peer setting
        can in the all-peer setting.";

    leaf session-ka-holdtime {
        type uint16 {
            range 45..3600;
        }
        units seconds;
        description
            "The time interval after which an inactive LDP session
            terminates and the corresponding TCP session closes.
            Inactivity is defined as not receiving LDP packets from the
            peer.
            This leaf may be configured at the per-peer level or the
            global level, with precedence given to the value at the
            per-peer level. If the leaf is not configured at either
            level, the default value at the global level is used.";
    }
    leaf session-ka-interval {
        type uint16 {
            range 15..1200;
        }
        units seconds;
        description
            "The interval between successive transmissions of keepalive
            packets. Keepalive packets are only sent in the absence of
            other LDP packets transmitted over the LDP session.
            This leaf may be configured at the per-peer level or the
            global level, with precedence given to the value at the
            per-peer level. If the leaf is not configured at either
            level, the default value at the global level is used.";
    }
} // peer-attributes

grouping peer-authentication {
    description
        "Peer authentication container, used in the per-peer setting
        can in the all-peer setting.";

    container authentication {
```

```
description
  "Containing authentication information.";
choice authentication-type {
  description
    "Choice of authentication.";
  case password {
    leaf key {
      type string;
      description
        "This leaf specifies the authentication key. The length
        of the key may be dependent on the cryptographic
        algorithm.";
    }
    leaf crypto-algorithm {
      type identityref {
        base key-chain:crypto-algorithm;
      }
      description
        "Cryptographic algorithm associated with key.";
    }
  }
}
} // peer-authentication

grouping peer-state-derived {
  description
    "The peer state information derived from the LDP protocol
    operations.";

  container label-advertisement-mode {
    config false;
    description "Label advertisement mode state.";
    leaf local {
      type label-adv-mode;
      description
        "Local Label Advertisement Mode.";
    }
    leaf peer {
      type label-adv-mode;
      description
        "Peer Label Advertisement Mode.";
    }
    leaf negotiated {
      type label-adv-mode;
      description
        "Negotiated Label Advertisement Mode.";
    }
  }
}
```



```
    }
    leaf next-keep-alive {
        type uint16;
        units seconds;
        config false;
        description
            "Time duration from now until sending the next KeepAlive
            message.";
    }

    container received-peer-state {
        config false;
        description
            "Operational state information learned from the peer.";

        uses graceful-restart-attributes-per-peer;

        container capability {
            description "Peer capability information.";
            container end-of-lib {
                description
                    "Peer's end-of-lib capability.";
                leaf enabled {
                    type boolean;
                    description
                        "'true' if peer's end-of-lib capability is enabled.";
                }
            }
        }
        container typed-wildcard-fec {
            description
                "Peer's typed-wildcard-fec capability.";
            leaf enabled {
                type boolean;
                description
                    "'true' if peer's typed-wildcard-fec capability is
                    enabled.";
            }
        }
        container upstream-label-assignment {
            description
                "Peer's upstream label assignment capability.";
            leaf enabled {
                type boolean;
                description
                    "'true' if peer's upstream label assignment is
                    enabled.";
            }
        }
    }
}
```

```
    } // capability
  } // received-peer-state

  container session-holdtime {
    config false;
    description "Session holdtime state.";
    leaf peer {
      type uint16;
      units seconds;
      description "Peer holdtime.";
    }
    leaf negotiated {
      type uint16;
      units seconds;
      description "Negotiated holdtime.";
    }
    leaf remaining {
      type uint16;
      units seconds;
      description "Remaining holdtime.";
    }
  } // session-holdtime

  leaf session-state {
    type enumeration {
      enum non-existent {
        description "NON EXISTENT state. Transport disconnected.";
      }
      enum initialized {
        description "INITIALIZED state.";
      }
      enum openrec {
        description "OPENREC state.";
      }
      enum opensent {
        description "OPENSENT state.";
      }
      enum operational {
        description "OPERATIONAL state.";
      }
    }
    config false;
    description
      "Representing the operational status of the LDP session.";
    reference
      "RFC5036, Sec. 2.5.4.";
  }
}
```

```
container tcp-connection {
  config false;
  description "TCP connection state.";
  leaf local-address {
    type inet:ip-address;
    description "Local address.";
  }
  leaf local-port {
    type inet:port-number;
    description "Local port number.";
  }
  leaf remote-address {
    type inet:ip-address;
    description "Remote address.";
  }
  leaf remote-port {
    type inet:port-number;
    description "Remote port number.";
  }
} // tcp-connection

leaf up-time {
  type rt-types:timeticks64;
  config false;
  description
    "The number of time ticks (hundredths of a second) since the
    the state of the session with the peer changed to
    OPERATIONAL.";
}

container statistics {
  config false;
  description
    "Statistics objects.";

  leaf discontinuity-time {
    type yang:date-and-time;
    mandatory true;
    description
      "The time on the most recent occasion at which any one or
      more of this interface's counters suffered a
      discontinuity.  If no such discontinuities have occurred
      since the last re-initialization of the local management
      subsystem, then this node contains the time the local
      management subsystem re-initialized itself.";
  }

  container received {
```

```
        description "Inbound statistics.";
        uses statistics-peer-received-sent;
    }
    container sent {
        description "Outbound statistics.";
        uses statistics-peer-received-sent;
    }

    leaf total-addresses {
        type uint32;
        description
            "The number of learned addresses.";
    }
    leaf total-labels {
        type uint32;
        description
            "The number of learned labels.";
    }
    leaf total-fec-label-bindings {
        type uint32;
        description
            "The number of learned label-address bindings.";
    }
} // statistics
} // peer-state-derived

grouping statistics-peer-received-sent {
    description
        "Inbound and outbound statistic counters.";
    leaf total-octets {
        type yang:counter64;
        description
            "The total number of octets sent or received.";
    }
    leaf total-messages {
        type yang:counter64;
        description
            "The number of messages sent or received.";
    }
    leaf address {
        type yang:counter64;
        description
            "The number of address messages sent or received.";
    }
    leaf address-withdraw {
        type yang:counter64;
        description
            "The number of address-withdraw messages sent or received.";
```

```
    }
    leaf initialization {
        type yang:counter64;
        description
            "The number of initialization messages sent or received.";
    }
    leaf keepalive {
        type yang:counter64;
        description
            "The number of keepalive messages sent or received.";
    }
    leaf label-abort-request {
        type yang:counter64;
        description
            "The number of label-abort-request messages sent or
            received.";
    }
    leaf label-mapping {
        type yang:counter64;
        description
            "The number of label-mapping messages sent or received.";
    }
    leaf label-release {
        type yang:counter64;
        description
            "The number of label-release messages sent or received.";
    }
    leaf label-request {
        type yang:counter64;
        description
            "The number of label-request messages sent or received.";
    }
    leaf label-withdraw {
        type yang:counter64;
        description
            "The number of label-withdraw messages sent or received.";
    }
    leaf notification {
        type yang:counter64;
        description
            "The number of notification messages sent or received.";
    }
} // statistics-peer-received-sent

/*
 * Configuration data and operational state data nodes
 */
```

```
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol" {
    when "derived-from-or-self(rt:type, 'ldp:mpls-ldp')" {
      description
        "This augmentation is only valid for a control-plane
        protocol instance of LDP (type 'mpls-ldp').";
    }
    description
      "LDP augmentation to routing control-plane protocol
      configuration and state.";

    container mpls-ldp {
      must "not (../../rt:control-plane-protocol"
        + "[derived-from-or-self(rt:type, 'ldp:mpls-ldp')]"
        + "[rt:name!=current()/../../rt:name])"
      {
        description "Only one LDP instance is allowed.";
      }

      description
        "Containing configuration and operational data for the LDP
        protocol.";

      container global {
        description
          "Global attributes for LDP.";

        container capability {
          description
            "Containing the LDP capability data. The container is
            used for augmentations.";
          reference
            "RFC5036: Sec. 1.5.";
        }

        container graceful-restart {
          description
            "Attributes for graceful restart.";
          leaf enabled {
            type boolean;
            default false;
            description
              "Enable or disable graceful restart.";
          }
          leaf reconnect-time {
            type uint16 {
              range 10..1800;
            }
          }
        }
      }
    }
  }
```

```
    units seconds;
    default 120;
    description
        "Specifies the time interval that the remote LDP peer
        must wait for the local LDP peer to reconnect after
        the remote peer detects the LDP communication
        failure.";
}
leaf recovery-time {
    type uint16 {
        range 30..3600;
    }
    units seconds;
    default 120;
    description
        "Specifies the time interval, in seconds, that the
        remote LDP peer preserves its MPLS forwarding state
        after receiving the Initialization message from the
        restarted local LDP peer.";
}
leaf forwarding-holdtime {
    type uint16 {
        range 30..3600;
    }
    units seconds;
    default 180;
    description
        "Specifies the time interval, in seconds, before the
        termination of the recovery phase.";
}
} // graceful-restart

leaf lsr-id {
    type rt-types:router-id;
    description
        "Specify the value to act as the LDP LSR ID.
        If this attribute is not specified, LDP uses the router
        ID as determined by the system.";
}

container address-families {
    description
        "Per address family configuration and operational state.
        The address family can be either IPv4 or IPv6.";
    container ipv4 {
        presence
            "Present if IPv4 is enabled, unless the 'enabled'
            leaf is set to 'false'";
    }
}
```

```
description
  "Containing data related to the IPv4 address family.";

leaf enabled {
  type boolean;
  default true;
  description
    "'false' to disable the address family.";
}

leaf label-distribution-control-mode {
  type enumeration {
    enum independent {
      description
        "Independent label distribution control.";
    }
    enum ordered {
      description
        "Ordered label distribution control.";
    }
  }
  config false;
  description
    "Label distribution control mode.";
  reference
    "RFC5036: LDP Specification. Sec 2.6.";
}

// ipv4 bindings
container bindings {
  config false;
  description
    "LDP address and label binding information.";
  list address {
    key "address";
    description
      "List of address bindings learned by LDP.";
    leaf address {
      type inet:ipv4-address;
      description
        "The IPv4 address learned from an Address
        message received from or advertised to a peer.";
    }
    uses binding-address-state-attributes;
  }

  list fec-label {
    key "fec";
```



```
    description
      "List of FEC-label bindings learned by LDP.";
    leaf fec {
      type inet:ipv4-prefix;
      description
        "The prefix FEC value in the FEC-label binding,
         learned in a Label Mapping message received from
         or advertised to a peer.";
    }
    uses binding-label-state-attributes;
  }
} // bindings
} // ipv4
} // address-families
} // global

container discovery {
  description
    "Neighbor discovery configuration and operational state.";

  container interfaces {
    description
      "A list of interfaces for LDP Basic Discovery.";
    reference
      "RFC5036: LDP Specification. Sec 2.4.1.";

    uses basic-discovery-timers {
      refine "hello-holdtime" {
        default 15;
      }
      refine "hello-interval" {
        default 5;
      }
    }
  }

  list interface {
    key "name";
    description
      "List of LDP interfaces used for LDP Basic Discovery.";
    uses ldp-interface-ref;
    leaf next-hello {
      type uint16;
      units seconds;
      config false;
      description "Time to send the next Hello message.";
    }

    container address-families {
```

```
description
  "Container for address families.";
container ipv4 {
  presence
    "Present if IPv4 is enabled, unless the 'enabled'
    leaf is set to 'false'";
  description
    "IPv4 address family.";

  leaf enabled {
    type boolean;
    default true;
    description
      "Set to false to disable the address family on
      the interface.";
  }

  container hello-adjacencies {
    config false;
    description
      "Containing a list of Hello adjacencies.";

    list hello-adjacency {
      key "adjacent-address";
      config false;
      description "List of Hello adjacencies.";

      leaf adjacent-address {
        type inet:ipv4-address;
        description
          "Neighbor address of the Hello adjacency.";
      }

      uses adjacency-state-attributes;
      uses ldp-peer-ref-from-interface;
    }
  } // ipv4
} // address-families
} // interface
} // interfaces

container targeted
{
  description
    "A list of targeted neighbors for extended discovery.";

  leaf hello-holdtime {
```

```
    type uint16 {
      range 15..3600;
    }
    units seconds;
    default 45;
    description
      "The time interval for which LDP targeted Hello
      adjacency is maintained in the absence of targeted
      Hello messages from an LDP neighbor.";
  }
  leaf hello-interval {
    type uint16 {
      range 5..3600;
    }
    units seconds;
    default 15;
    description
      "The interval between consecutive LDP targeted Hello
      messages used in extended LDP discovery.";
  }

  container hello-accept {
    description
      "LDP policy to control the acceptance of extended
      neighbor discovery Hello messages.";

    leaf enabled {
      type boolean;
      default false;
      description
        "'true' to accept; 'false' to deny.";
    }
  }

  container address-families {
    description
      "Container for address families.";
    container ipv4 {
      presence
        "Present if IPv4 is enabled.";
      description
        "IPv4 address family.";

      container hello-adjacencies {
        config false;
        description
          "Containing a list of Hello adjacencies.";
      }
    }
  }
}
```

```
list hello-adjacency {
  key "local-address adjacent-address";
  description "List of Hello adjacencies.";

  leaf local-address {
    type inet:ipv4-address;
    description
      "Local address of the Hello adjacency.";
  }
  leaf adjacent-address {
    type inet:ipv4-address;
    description
      "Neighbor address of the Hello adjacency.";
  }

  uses adjacency-state-attributes;
  uses ldp-peer-ref-from-target;
}

list target {
  key "adjacent-address";
  description
    "Targeted discovery params.";

  leaf adjacent-address {
    type inet:ipv4-address;
    description
      "Configures a remote LDP neighbor for the
       extended LDP discovery.";
  }

  leaf enabled {
    type boolean;
    default true;
    description
      "'true' to enable the target.";
  }

  leaf local-address {
    type inet:ipv4-address;
    description
      "The local address used as the source address to
       send targeted Hello messages.
       If the value is not specified, the
       transport-address is used as the source
       address.";
  }
} // target
```

```
        } // ipv4
      } // address-families
    } // targeted
  } // discovery

  container peers {
    description
      "Peers configuration attributes.";

    uses peer-authentication;
    uses peer-attributes {
      refine session-ka-holdtime {
        default 180;
      }
      refine session-ka-interval {
        default 60;
      }
    }

    list peer {
      key "lsr-id label-space-id";
      description
        "List of peers.";

      leaf lsr-id {
        type rt-types:router-id;
        description
          "The LSR ID of the peer, to identify the globally
          unique LSR. This is the first four octets of the LDP
          ID. This leaf is used together with the leaf
          'label-space-id' to form the LDP ID.";
        reference
          "RFC5036. Sec 2.2.2.";
      }
      leaf label-space-id {
        type uint16;
        description
          "The Label Space ID of the peer, to identify a specific
          label space within the LSR. This is the last two
          octets of the LDP ID. This leaf is used together with
          the leaf 'lsr-id' to form the LDP ID.";
        reference
          "RFC5036. Sec 2.2.2.";
      }
    }

    uses peer-authentication;

    container address-families {
```

```
description
  "Per-vrf per-af params.";
container ipv4 {
  presence
    "Present if IPv4 is enabled.";
  description
    "IPv4 address family.";

  container hello-adjacencies {
    config false;
    description
      "Containing a list of Hello adjacencies.";

    list hello-adjacency {
      key "local-address adjacent-address";
      description "List of Hello adjacencies.";

      leaf local-address {
        type inet:ipv4-address;
        description
          "Local address of the Hello adjacency.";
      }
      leaf adjacent-address {
        type inet:ipv4-address;
        description
          "Neighbor address of the Hello adjacency.";
      }
    }

    uses adjacency-state-attributes;

    leaf interface {
      type if:interface-ref;
      description "Interface for this adjacency.";
    }
  }
} // ipv4
} // address-families

uses peer-state-derived;
} // list peer
} // peers
} // container mpls-ldp
}

/*
 * RPCs
 */
```

```
rpc mpls-ldp-clear-peer {
  description
    "Clears the session to the peer.";
  input {
    uses ldp-peer-ref-absolute {
      description
        "The LDP peer to be cleared. If this is not provided
        then all peers are cleared.
        The peer is identified by its LDP ID, which consists of
        the LSR ID and the Label Space ID.";
    }
  }
}

rpc mpls-ldp-clear-hello-adjacency {
  description
    "Clears the hello adjacency";
  input {
    container hello-adjacency {
      description
        "Link adjacency or targetttted adjacency. If this is not
        provided then all Hello adjacencies are cleared";
      leaf protocol-name {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rt:name";
        }
        description
          "The name of the LDP protocol instance.";
      }
      choice hello-adjacency-type {
        description "Adjacency type.";
        case targeted {
          container targeted {
            presence "Present to clear targeted adjacencies.";
            description
              "Clear targeted adjacencies.";
            leaf target-address {
              type inet:ip-address;
              description
                "The target address. If this is not provided then
                all targeted adjacencies are cleared";
            }
          }
        }
        case link {
          container link {
            presence "Present to clear link adjacencies.";
          }
        }
      }
    }
  }
}
```

```

    description
      "Clear link adjacencies.";
    leaf next-hop-interface {
      type leafref {
        path "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/mpls-ldp/discovery/"
          + "interfaces/interface/name";
      }
      description
        "Interface connecting to next-hop. If this is not
        provided then all link adjacencies are cleared.";
    }
    leaf next-hop-address {
      type inet:ip-address;
      must "../next-hop-interface" {
        description
          "Applicable when interface is specified.";
      }
      description
        "IP address of next-hop. If this is not provided
        then adjacencies to all next-hops on the given
        interface are cleared.";
    }
  }
} // hello-adjacency-type
} // hello-adjacency
} // input
} // mpls-ldp-clear-hello-adjacency

rpc mpls-ldp-clear-peer-statistics {
  description
    "Clears protocol statistics (e.g. sent and received
    counters).";
  input {
    uses ldp-peer-ref-absolute {
      description
        "The LDP peer whose statistics are to be cleared.
        If this is not provided then all peers' statistics are
        cleared.
        The peer is identified by its LDP ID, which consists of
        the LSR ID and the Label Space ID.";
    }
  }
}

/*
 * Notifications

```



```
*/
notification mpls-ldp-peer-event {

  description
    "Notification event for a change of LDP peer operational
    status.";
  leaf event-type {
    type oper-status-event-type;
    description "Event type.";
  }
  container peer {
    description
      "Reference to an LDP peer, by the LDP ID, which consists of
      the LSR ID and the Label Space ID.";
    uses ldp-peer-ref-absolute;
  }
}

notification mpls-ldp-hello-adjacency-event {
  description
    "Notification event for a change of LDP adjacency operational
    status.";
  leaf event-type {
    type oper-status-event-type;
    description "Event type.";
  }
  leaf protocol-name {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rt:name";
    }
    description
      "The name of the LDP protocol instance.";
  }
  choice hello-adjacency-type {
    description
      "Interface or targeted adjacency.";
    case targeted {
      container targeted {
        description
          "Targeted adjacency through LDP extended discovery.";
        leaf target-address {
          type inet:ip-address;
          description
            "The target adjacent address learned.";
        }
      }
    }
  }
}
```

```
    case link {
      container link {
        description
          "Link adjacency through LDP basic discovery.";
        leaf next-hop-interface {
          type if:interface-ref;
          description
            "The interface connecting to the adjacent next hop.";
        }
        leaf next-hop-address {
          type inet:ip-address;
          must "../next-hop-interface" {
            description
              "Applicable when interface is specified.";
          }
          description
            "IP address of the next hop. This can be IPv4 or IPv6
            address.";
        }
      }
    } // hello-adjacency-type
  } // mpls-ldp-hello-adjacency-event

notification mpls-ldp-fec-event {
  description
    "Notification event for a change of FEC status.";
  leaf event-type {
    type oper-status-event-type;
    description "Event type.";
  }
  leaf protocol-name {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rt:name";
    }
    description
      "The name of the LDP protocol instance.";
  }
  leaf fec {
    type inet:ip-prefix;
    description
      "The address prefix element of the FEC whose status
      has changed.";
  }
}
}
```

<CODE ENDS>

Figure 10: LDP base module

9.2. Extended

This YANG module imports types defined in [RFC6991], [RFC8349], [RFC8177], and [RFC8343].

```
<CODE BEGINS> file "ietf-mpls-ldp-extended@2020-02-25.yang"

// RFC Editor: replace the above date 2020-02-25 with the date of
// publication and remove this note.

module ietf-mpls-ldp-extended {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp-extended";
  prefix "ldp-ext";

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-routing {
    prefix "rt";
    reference
      "RFC 8349: A YANG Data Model for Routing Management (NMDA
      version)";
  }

  import ietf-key-chain {
    prefix "key-chain";
    reference "RFC 8177: YANG Data Model for Key Chains";
  }

  import ietf-mpls-ldp {
    prefix "ldp";
    reference "RFC XXXX: YANG Data Model for MPLS LDP";
    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
  }
}
```

```
import ietf-interfaces {
  prefix "if";
  reference "RFC 8343: A YANG Data Model for Interface Management";
}

import ietf-routing-policy {
  prefix rt-pol;
  reference
    "I-D.ietf-rtgwg-policy-model: A YANG Data Model for Routing
    Policy Management";
}

organization
  "IETF MPLS Working Group";
contact
  "WG Web:  <http://tools.ietf.org/wg/mppls/>
  WG List:  <mailto:mppls@ietf.org>

  Editor:    Kamran Raza
             <mailto:skraza@cisco.com>

  Editor:    Rajiv Asati
             <mailto:rajiva@cisco.com>

  Editor:    Xufeng Liu
             <mailto:xufeng.liu.ietf@gmail.com>

  Editor:    Santosh Esale
             <mailto:sesale@juniper.net>

  Editor:    Xia Chen
             <mailto:jescia.chenxia@huawei.com>

  Editor:    Himanshu Shah
             <mailto:hshah@ciena.com>";

description
  "This YANG module defines the extended components for the
  management of Multi-Protocol Label Switching (MPLS) Label
  Distribution Protocol (LDP). It is also the model to
  be augmented for extended Multipoint LDP (mLDP).

  Copyright (c) 2020 IETF Trust and the persons identified as
  authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
```

forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";

// RFC Editor: replace XXXX with actual RFC number and remove
// this note

```
revision 2020-02-25 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Data Model for MPLS LDP.";

    // RFC Editor: replace XXXX with actual RFC number and remove
    // this note
}

/*
 * Features
 */
feature capability-end-of-lib {
  description
    "This feature indicates that the system allows to configure
    LDP end-of-lib capability.";
}

feature capability-typed-wildcard-fec {
  description
    "This feature indicates that the system allows to configure
    LDP typed-wildcard-fec capability.";
}

feature capability-upstream-label-assignment {
  description
    "This feature indicates that the system allows to configure
    LDP upstream label assignment capability.";
}

feature forwarding-nexthop-config {
  description
    "This feature indicates that the system allows to configure
    forwarding nexthop on interfaces.";
}

feature graceful-restart-helper-mode {
```

```
    description
      "This feature indicates that the system supports graceful
      restart helper mode. We call an LSR to be operating in GR
      helper mode when it advertises 0 as its FT Reconnect Timeout
      in the FT Session TLV.
      Please refer RFC3478 section 2 for details.";
  }

  feature key-chain {
    description
      "This feature indicates that the system supports keychain for
      authentication.";
  }

  feature peers-dual-stack-transport-preference {
    description
      "This feature indicates that the system allows to configure
      the transport connection preference in a dual-stack setup
      for peers.";
  }

  feature per-interface-timer-config {
    description
      "This feature indicates that the system allows to configure
      interface Hello timers at the per-interface level.";
  }

  feature per-peer-admin-down {
    description
      "This feature indicates that the system allows to
      administratively disable a peer.";
  }

  feature per-peer-graceful-restart-config {
    description
      "This feature indicates that the system allows to configure
      graceful restart at the per-peer level.";
  }

  feature per-peer-session-attributes-config {
    description
      "This feature indicates that the system allows to configure
      session attributes at the per-peer level.";
  }

  feature policy-label-assignment-config {
    description
      "This feature indicates that the system allows to configure
```

```
        policies to assign labels according to certain prefixes.";
    }

    feature policy-ordered-label-config {
        description
            "This feature indicates that the system allows to configure
            ordered label policies.";
    }

    feature policy-targeted-discovery-config {
        description
            "This feature indicates that the system allows to configure
            policies to control the acceptance of targeted neighbor
            discovery Hello messages.";
    }

    feature session-downstream-on-demand-config {
        description
            "This feature indicates that the system allows to configure
            session downstream-on-demand";
    }

    /*
     * Typedefs
     */
    typedef neighbor-list-ref {
        type leafref {
            path "/rt-pol:routing-policy/rt-pol:defined-sets/"
                + "rt-pol:neighbor-sets/rt-pol:neighbor-set/rt-pol:name";
        }
        description
            "A type for a reference to a neighbor address list.
            The string value is the name identifier for uniquely
            identifying the referenced address list, which contains a list
            of addresses that a routing policy can applied.";
        reference
            "I-D.ietf-rtgwg-policy-model: A YANG Data Model for Routing
            Policy Management";
    }

    typedef prefix-list-ref {
        type leafref {
            path "/rt-pol:routing-policy/rt-pol:defined-sets/"
                + "rt-pol:prefix-sets/rt-pol:prefix-set/rt-pol:name";
        }
        description
            "A type for a reference to a prefix list.
            The string value is the name identifier for uniquely
```

```
        identifying the referenced prefix set, which contains a list
        of prefixes that a routing policy can applied.";
    reference
        "I-D.ietf-rtgwg-policy-model: A YANG Data Model for Routing
        Policy Management";
}

typedef peer-list-ref {
    type leafref {
        path "/rt-pol:routing-policy/rt-pol:defined-sets/"
            + "rt-pol:neighbor-sets/rt-pol:neighbor-set/rt-pol:name";
    }
    description
        "A type for a reference to a peer address list.
        The string value is the name identifier for uniquely
        identifying the referenced address list, which contains a list
        of addresses that a routing policy can applied.";
    reference
        "I-D.ietf-rtgwg-policy-model: A YANG Data Model for Routing
        Policy Management";
}

/*
 * Identities
 */

/*
 * Groupings
 */
grouping address-family-ipv4-augment {
    description "Augmentation to address family IPv4.";

    uses policy-container;

    leaf transport-address {
        type inet:ipv4-address;
        description
            "The transport address advertised in LDP Hello messages.
            If this value is not specified, the LDP LSR ID is used as
            the transport address.";
        reference
            "RFC5036. Sec. 3.5.2.";
    }
}

grouping authentication-keychain-augment {
    description "Augmentation to authentication to add keychain.";
```



```
    leaf key-chain {
      type key-chain:key-chain-ref;
      description
        "key-chain name.
        If not specified, no key chain is used.";
    }
  }

  grouping capability-augment {
    description "Augmentation to capability.";

    container end-of-lib {
      if-feature capability-end-of-lib;
      description
        "Configure end-of-lib capability.";
      leaf enabled {
        type boolean;
        default false;
        description
          "'true' to enable end-of-lib capability.";
      }
    }

    container typed-wildcard-fec {
      if-feature capability-typed-wildcard-fec;
      description
        "Configure typed-wildcard-fec capability.";
      leaf enabled {
        type boolean;
        default false;
        description
          "'true' to enable typed-wildcard-fec capability.";
      }
    }

    container upstream-label-assignment {
      if-feature capability-upstream-label-assignment;
      description
        "Configure upstream label assignment capability.";
      leaf enabled {
        type boolean;
        default false;
        description
          "'true' to enable upstream label assignment.";
      }
    }
  } // capability-augment

  grouping global-augment {
    description "Augmentation to global attributes.";
```

```
leaf igp-synchronization-delay {
  type uint16 {
    range "0 | 3..300";
  }
  units seconds;
  default 0;
  description
    "Sets the interval that the LDP waits before notifying the
    Interior Gateway Protocol (IGP) that label exchange is
    completed so that IGP can start advertising the normal
    metric for the link.
    If the value is not specified, there is no delay.";
}

grouping global-forwarding-nexthop-augment {
  description
    "Augmentation to global forwarding nexthop interfaces.";

  container forwarding-nexthop {
    if-feature forwarding-nexthop-config;
    description
      "Configuration for forwarding nexthop.";

    container interfaces {
      description
        "Containing a list of interfaces on which forwarding can be
        disabled.";

      list interface {
        key "name";
        description
          "List of LDP interfaces on which forwarding can be
          disabled.";
        uses ldp:ldp-interface-ref;
        list address-family {
          key "afi";
          description
            "Per-vrf per-af params.";
          leaf afi {
            type identityref {
              base rt:address-family;
            }
            description
              "Address family type value.";
          }
          leaf ldp-disable {
            type boolean;
          }
        }
      }
    }
  }
}
```

```
        default false;
        description
            "'true' to disable LDP forwarding on the interface.";
    }
} // interface
} // interfaces
} // forwarding-nexthop
} // global-forwarding-nexthop-augment

grouping graceful-restart-augment {
    description "Augmentation to graceful restart.";

    leaf helper-enabled {
        if-feature graceful-restart-helper-mode;
        type boolean;
        default false;
        description
            "Enable or disable graceful restart helper mode.";
    }
}

grouping interface-address-family-ipv4-augment {
    description "Augmentation to interface address family IPv4.";

    leaf transport-address {
        type union {
            type enumeration {
                enum "use-global-transport-address" {
                    description
                        "Use the transport address set at the global level
                        common for all interfaces for this address family.";
                }
                enum "use-interface-address" {
                    description
                        "Use interface address as the transport address.";
                }
            }
            type inet:ipv4-address;
        }
        default "use-global-transport-address";
        description
            "IP address to be advertised as the LDP transport address.";
    }
}

grouping interface-address-family-ipv6-augment {
    description "Augmentation to interface address family IPv6.";
```

```
leaf transport-address {
  type union {
    type enumeration {
      enum "use-global-transport-address" {
        description
          "Use the transport address set at the global level
           common for all interfaces for this address family.";
      }
      enum "use-interface-address" {
        description
          "Use interface address as the transport address.";
      }
    }
    type inet:ipv6-address;
  }
  default "use-global-transport-address";
  description
    "IP address to be advertised as the LDP transport address.";
}

grouping interface-augment {
  description "Augmentation to interface.";

  uses ldp:basic-discovery-timers {
    if-feature per-interface-timer-config;
  }
  leaf igp-synchronization-delay {
    if-feature per-interface-timer-config;
    type uint16 {
      range "0 | 3..300";
    }
    units seconds;
    description
      "Sets the interval that the LDP waits before notifying the
       Interior Gateway Protocol (IGP) that label exchange is
       completed so that IGP can start advertising the normal
       metric for the link.
       This leaf may be configured at the per-interface level or
       the global level, with precedence given to the value at the
       per-interface level.  If the leaf is not configured at
       either level, the default value at the global level is
       used.";
  }
}

grouping peer-af-policy-container {
  description
```

```
    "LDP policy attribute container under peer address-family.";
  container label-policy {
    description
      "Label policy attributes.";
    container advertise {
      description
        "Label advertising policies.";
      leaf prefix-list {
        type prefix-list-ref;
        description
          "Applies the prefix list to filter outgoing label
           advertisements.
           If the value is not specified, no prefix filter
           is applied.";
      }
    }
    container accept {
      description
        "Label advertisement acceptance policies.";
      leaf prefix-list {
        type prefix-list-ref;
        description
          "Applies the prefix list to filter incoming label
           advertisements.
           If the value is not specified, no prefix filter
           is applied.";
      }
    }
  }
} // peer-af-policy-container

grouping peer-augment {
  description "Augmentation to each peer list entry.";

  leaf admin-down {
    if-feature per-peer-admin-down;
    type boolean;
    default false;
    description
      "'true' to disable the peer.";
  }

  uses ldp:graceful-restart-attributes-per-peer {
    if-feature per-peer-graceful-restart-config;
  }

  uses ldp:peer-attributes {
    if-feature per-peer-session-attributes-config;
  }
}
```

```
    }  
  }  
  
  grouping peers-augment {  
    description "Augmentation to peers container.";  
  
    container session-downstream-on-demand {  
      if-feature session-downstream-on-demand-config;  
      description  
        "Session downstream-on-demand attributes.";  
      leaf enabled {  
        type boolean;  
        default false;  
        description  
          "'true' if session downstream-on-demand is enabled.";  
      }  
      leaf peer-list {  
        type peer-list-ref;  
        description  
          "The name of a peer ACL, to be applied to the  
          downstream-on-demand sessions.  
          If this value is not specified, no filter is applied to  
          any downstream-on-demand sessions.";  
      }  
    }  
  }  
  
  container dual-stack-transport-preference {  
    if-feature peers-dual-stack-transport-preference;  
    description  
      "The settings of peers to establish TCP connection in a  
      dual-stack setup.";  
    leaf max-wait {  
      type uint16 {  
        range "0..60";  
      }  
      default 30;  
      description  
        "The maximum wait time in seconds for preferred transport  
        connection establishment. 0 indicates no preference.";  
    }  
  }  
  
  container prefer-ipv4 {  
    presence  
      "Present if IPv4 is preferred for transport connection  
      establishment, subject to the 'peer-list' in this  
      container.";  
    description  
      "Uses IPv4 as the preferred address family for transport  
      connection establishment, subject to the 'peer-list' in  
      this container."
```

```
        If this container is not present, as a default, IPv6 is
        the preferred address family for transport connection
        establishment.";
    leaf peer-list {
        type peer-list-ref;
        description
            "The name of a peer ACL, to be applied to the IPv4
            transport connections.
            If this value is not specified, no filter is applied,
            and the IPv4 is preferred for all peers.";
    }
}
} // peers-augment

grouping policy-container {
    description
        "LDP policy attributes.";
    container label-policy {
        description
            "Label policy attributes.";
    }
    container advertise {
        description
            "Label advertising policies.";
    }
    container egress-explicit-null {
        description
            "Enables an egress router to advertise an
            explicit null label (value 0) in place of an
            implicit null label (value 3) to the
            penultimate hop router.";
        leaf enabled {
            type boolean;
            default false;
            description
                "'true' to enable explicit null.";
        }
    }
    leaf prefix-list {
        type prefix-list-ref;
        description
            "Applies the prefix list to filter outgoing label
            advertisements.
            If the value is not specified, no prefix filter
            is applied.";
    }
}
container accept {
    description
```

```
        "Label advertisement acceptance policies.";
    leaf prefix-list {
        type prefix-list-ref;
        description
            "Applies the prefix list to filter incoming label
             advertisements.
             If the value is not specified, no prefix filter
             is applied.";
    }
}
container assign {
    if-feature policy-label-assignment-config;
    description
        "Label assignment policies";
    container independent-mode {
        description
            "Independent label policy attributes.";
        leaf prefix-list {
            type prefix-list-ref;
            description
                "Assign labels according to certain prefixes.
                 If the value is not specified, no prefix filter
                 is applied (labels are assigned to all learned
                 routes).";
        }
    }
}
container ordered-mode {
    if-feature policy-ordered-label-config;
    description
        "Ordered label policy attributes.";
    leaf egress-prefix-list {
        type prefix-list-ref;
        description
            "Assign labels according to certain prefixes for
             egress LSR.";
    }
}
} // assign
} // label-policy
} // policy-container

/*
 * Configuration and state data nodes
 */
// Forwarding nexthop augmentation to the global tree
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global" {
```



```
    description "Forwarding nexthop augmentation.";
    uses global-forwarding-nexthop-augment;
}

// global/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/"
+ "ldp:address-families" {
    description "Global IPv6 augmentation.";

    container ipv6 {
        presence
            "Present if IPv6 is enabled, unless the 'enabled'
            leaf is set to 'false'";
        description
            "Containing data related to the IPv6 address family.";

        leaf enabled {
            type boolean;
            default true;
            description
                "'false' to disable the address family.";
        }

        uses policy-container;

        leaf transport-address {
            type inet:ipv6-address;
            mandatory true;
            description
                "The transport address advertised in LDP Hello messages.";
        }

        leaf label-distribution-control-mode {
            type enumeration {
                enum independent {
                    description
                        "Independent label distribution control.";
                }
                enum ordered {
                    description
                        "Ordered label distribution control.";
                }
            }
            config false;
            description
                "Label distribution control mode.";
            reference
```

```
    "RFC5036: LDP Specification. Sec 2.6.";
  }

  // ipv6 bindings
  container bindings {
    config false;
    description
      "LDP address and label binding information.";
    list address {
      key "address";
      description
        "List of address bindings learned by LDP.";
      leaf address {
        type inet:ipv6-address;
        description
          "The IPv6 address learned from an Address
            message received from or advertised to a peer.";
      }
      uses ldp:binding-address-state-attributes;
    }

    list fec-label {
      key "fec";
      description
        "List of FEC-label bindings learned by LDP.";
      leaf fec {
        type inet:ipv6-prefix;
        description
          "The prefix FEC value in the FEC-label binding,
            learned in a Label Mapping message received from
            or advertised to a peer.";
      }
      uses ldp:binding-label-state-attributes;
    }
  } // bindings
} // ipv6

// discovery/interfaces/interface/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:interfaces/ldp:interface/"
+ "ldp:address-families" {
  description "Interface IPv6 augmentation.";

  container ipv6 {
    presence
      "Present if IPv6 is enabled, unless the 'enabled'
```

```
        leaf is set to 'false'";
    description
        "IPv6 address family.";

    leaf enabled {
        type boolean;
        default true;
        description
            "'false' to disable the address family on the interface.";
    }

    container hello-adjacencies {
        config false;
        description
            "Containing a list of Hello adjacencies.";

        list hello-adjacency {
            key "adjacent-address";
            config false;
            description "List of Hello adjacencies.";

            leaf adjacent-address {
                type inet:ipv6-address;
                description
                    "Neighbor address of the Hello adjacency.";
            }

            uses ldp:adjacency-state-attributes;
            uses ldp:ldp-peer-ref-from-interface;
        }
    }
} // ipv6
}

// discovery/targeted/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:targeted/ldp:address-families" {
    description "Targeted discovery IPv6 augmentation.";

    container ipv6 {
        presence
            "Present if IPv6 is enabled.";
        description
            "IPv6 address family.";

        container hello-adjacencies {
            config false;
        }
    }
}
```

```
description
  "Containing a list of Hello adjacencies.";

list hello-adjacency {
  key "local-address adjacent-address";
  config false;
  description "List of Hello adjacencies.";

  leaf local-address {
    type inet:ipv6-address;
    description
      "Local address of the Hello adjacency.";
  }
  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Neighbor address of the Hello adjacency.";
  }

  uses ldp:adjacency-state-attributes;
  uses ldp:ldp-peer-ref-from-target;
}

list target {
  key "adjacent-address";
  description
    "Targeted discovery params.";

  leaf adjacent-address {
    type inet:ipv6-address;
    description
      "Configures a remote LDP neighbor for the
       extended LDP discovery.";
  }
  leaf enabled {
    type boolean;
    default true;
    description
      "'true' to enable the target.";
  }
  leaf local-address {
    type inet:ipv6-address;
    description
      "The local address used as the source address to send
       targeted Hello messages.
       If the value is not specified, the transport-address
       is used as the source address.";
```

```
    }
  } // target
} // ipv6
}

// /peers/peer/state/address-families/ipv6
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/"
+ "ldp:peer/ldp:address-families" {
  description "Peer state IPv6 augmentation.";

  container ipv6 {
    presence
      "Present if IPv6 is enabled.";
    description
      "IPv6 address family.";

    container hello-adjacencies {
      config false;
      description
        "Containing a list of Hello adjacencies.";

      list hello-adjacency {
        key "local-address adjacent-address";
        description "List of Hello adjacencies.";

        leaf local-address {
          type inet:ipv6-address;
          description
            "Local address of the Hello adjacency.";
        }
        leaf adjacent-address {
          type inet:ipv6-address;
          description
            "Neighbor address of the Hello adjacency.";
        }
      }

      uses ldp:adjacency-state-attributes;

      leaf interface {
        type if:interface-ref;
        description "Interface for this adjacency.";
      }
    }
  }
} // ipv6
}
```

```
/*
 * Configuration data and operational state data nodes
 */
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global" {
  description "Graceful restart augmentation.";
  uses global-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/"
+ "ldp:capability" {
  description "Capability augmentation.";
  uses capability-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/"
+ "ldp:graceful-restart" {
  description "Graceful restart augmentation.";
  uses graceful-restart-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:global/"
+ "ldp:address-families/ldp:ipv4" {
  description "Address family IPv4 augmentation.";
  uses address-family-ipv4-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:interfaces/ldp:interface" {
  description "Interface augmentation.";
  uses interface-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:interfaces/ldp:interface/ldp:address-families/"
+ "ldp:ipv4" {
  description "Interface address family IPv4 augmentation.";
  uses interface-address-family-ipv4-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:interfaces/ldp:interface/ldp:address-families/"
```

```
+ "ldp-ext:ipv6" {
  description "Interface address family IPv6 augmentation.";
  uses interface-address-family-ipv6-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:discovery/"
+ "ldp:targeted/ldp:hello-accept" {
  description "Targeted discovery augmentation.";
  leaf neighbor-list {
    if-feature policy-targeted-discovery-config;
    type neighbor-list-ref;
    description
      "The name of a neighbor ACL, to accept Hello messages from
       LDP peers as permitted by the neighbor-list policy.
       If this value is not specified, targeted Hello messages from
       any source are accepted.";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers" {
  description "Peers augmentation.";
  uses peers-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/"
+ "ldp:authentication/ldp:authentication-type" {
  if-feature key-chain;
  description "Peers authentication augmentation.";
  case key-chain {
    uses authentication-keychain-augment;
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/ldp:peer" {
  description "Peer list entry augmentation.";
  uses peer-augment;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/ldp:peer/"
+ "ldp:authentication/ldp:authentication-type" {
  if-feature key-chain;
  description "Peer list entry authentication augmentation.";
  case key-chain {
```

```
        uses authentication-keychain-augment;
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/ldp:peer/"
+ "ldp:address-families/ldp:ipv4" {
    description
        "Peer list entry IPv4 augmentation.";
    uses peer-af-policy-container;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/ldp:mpls-ldp/ldp:peers/ldp:peer/"
+ "ldp:address-families/ldp-ext:ipv6" {
    description
        "Peer list entry IPv6 augmentation.";
    uses peer-af-policy-container;
}
}

<CODE ENDS>
```

Figure 11: LDP extended module

10. Security Considerations

This specification inherits the security considerations captured in [RFC5920] and the LDP protocol specification documents, namely base LDP [RFC5036], LDP IPv6 [RFC7552], LDP Capabilities [RFC5561], Typed Wildcard FEC [RFC5918], LDP End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

10.1. YANG model

The YANG modules specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or

RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

10.1.1. Writable nodes

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

For LDP, the ability to modify MPLS LDP configuration may allow the entire MPLS LDP domain to be compromised including forming LDP adjacencies and/or peer sessions with unauthorized routers to mount a massive Denial-of-Service (DoS) attack. In particular, following are the subtrees and data nodes that are sensitive and vulnerable:

- * /mpls-ldp/discovery/interfaces/interface: Adding LDP on any unprotected interface could allow an LDP hello adjacency to be formed with an unauthorized and malicious neighbor. Once an hello adjacency is formed, a peer session could progress with this neighbor.
- * /mpls-ldp/discovery/targeted/hello-accept: Allowing acceptance of targeted-hellos could open LDP to DoS attacks related to incoming targeted hellos from malicious sources.
- * /mpls-ldp/peers/authentication: Allowing a peer session establishment is typically controlled via LDP authentication where a proper and secure authentication password/key management is warranted.
- * /mpls-ldp/peers/peer/authentication: Same as above.

10.1.2. Readable nodes

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

The exposure of LDP databases (such as hello adjacencies, peers, address bindings, and fec-label bindings) beyond the scope of the LDP admin domain may be undesirable. The relevant subtrees and data nodes are as follows:

- * /mpls-ldp/global/address-families/ipv4/bindings/address
- * /mpls-ldp/global/address-families/ipv6/bindings/address
- * /mpls-ldp/global/address-families/ipv4/bindings/fec-label
- * /mpls-ldp/global/address-families/ipv6/bindings/fec-label
- * /mpls-ldp/discovery/interfaces/interface/address-families/ipv4/hello-adjacencies
- * /mpls-ldp/discovery/interfaces/interface/address-families/ipv6/hello-adjacencies
- * /mpls-ldp/discovery/targeted/address-families/ipv4/hello-adjacencies
- * /mpls-ldp/discovery/targeted/address-families/ipv6/hello-adjacencies
- * /mpls-ldp/peers

The configuration for LDP peer authentication is supported via the specification of key-chain [RFC8040], or via direct specification of a key associated with a crypto algorithm (such as MD5). The relevant subtrees and data nodes are as follows:

- * /mpls-ldp/peers/authentication
- * /mpls-ldp/peers/peer/authentication

The actual authentication key data (whether locally specified or part of a key-chain) is sensitive and needs to be kept secret from unauthorized parties. For key-chain based authentication, this model inherits the security considerations of [RFC8040] (that includes the considerations with respect to the local storage and handling of authentication keys). A similar procedure for storage and access to direct key is warranted.

10.1.3. RPC operations

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations otherwise control plane flaps, network outages, and DoS attacks are possible. The RPC operations are:

- * mpls-ldp-clear-peer

* mpls-ldp-clear-hello-adjacency

10.1.4. Notifications

The model describes several notifications. The implementations must rate-limit the generation of these notifications to avoid creating significant notification load and possible side effects on the system stability.

11. IANA Considerations

This document requests the registration of the following URIs in the IETF "XML registry" [RFC3688]:

URI	Registrant	XML
urn:ietf:params:xml:ns:yang:ietf-mpls-ldp	The IESG	N/A
urn:ietf:params:xml:ns:yang:ietf-mpls-ldp-extended	The IESG	N/A

Table 1: URIs

This document requests the registration of the following YANG modules in the "YANG Module Names" registry [RFC6020]:

Name	Namespace	Prefix	Reference
ietf-mpls-ldp	urn:ietf:params:xml:ns:yang: :ietf-mpls-ldp	ldp	This document
ietf-mpls-ldp-extended	urn:ietf:params:xml:ns:yang: :ietf-mpls-ldp-extended	ldp- ext	This document

Table 2: YANG Modules

-- RFC Editor: Replace "this document" with the document RFC number at time of publication, and remove this note.

12. Acknowledgments

The authors would like to acknowledge Eddie Chami, Nagendra Kumar, Mannan Venkatesan, and Pavan Beeram for their contribution to this document.

We also acknowledge Ladislav Lhotka, Jan Lindblad, Tom Petch, Yingzhen Qu, and Benjamin Kaduk for their detailed review of the model during WG and IESG.

13. Contributors

Danial Johari
Cisco Systems
Email: dajohari@cisco.com

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Jeff Tantsura
Apstra
Email: jefftant.ietf@gmail.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Reshad Rahman
Cisco Systems
Email: rrahman@cisco.com

Stephane Litkowski
Cisco Systems
Email: slitkows@cisco.com

14. Normative References

- [I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy Management", Work in Progress, Internet-Draft, draft-ietf-rtgwg-policy-model-09, 4 March 2020, <<https://tools.ietf.org/html/draft-ietf-rtgwg-policy-model-09>>.
- [RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, DOI 10.17487/RFC3478, February 2003, <<https://www.rfc-editor.org/info/rfc3478>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,

- DOI 10.17487/RFC3688, January 2004,
<<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed.,
"LDP Specification", RFC 5036, DOI 10.17487/RFC5036,
October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream
Label Assignment and Context-Specific Label Space",
RFC 5331, DOI 10.17487/RFC5331, August 2008,
<<https://www.rfc-editor.org/info/rfc5331>>.
- [RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP
Synchronization", RFC 5443, DOI 10.17487/RFC5443, March
2009, <<https://www.rfc-editor.org/info/rfc5443>>.
- [RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL.
Le Roux, "LDP Capabilities", RFC 5561,
DOI 10.17487/RFC5561, July 2009,
<<https://www.rfc-editor.org/info/rfc5561>>.
- [RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution
Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class
(FEC)", RFC 5918, DOI 10.17487/RFC5918, August 2010,
<<https://www.rfc-editor.org/info/rfc5918>>.
- [RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas,
"Signaling LDP Label Advertisement Completion", RFC 5919,
DOI 10.17487/RFC5919, August 2010,
<<https://www.rfc-editor.org/info/rfc5919>>.
- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS
Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010,
<<https://www.rfc-editor.org/info/rfc5920>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
the Network Configuration Protocol (NETCONF)", RFC 6020,
DOI 10.17487/RFC6020, October 2010,
<<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
<<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,
<<https://www.rfc-editor.org/info/rfc6242>>.

- [RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, DOI 10.17487/RFC6389, November 2011, <<https://www.rfc-editor.org/info/rfc6389>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <<https://www.rfc-editor.org/info/rfc7277>>.
- [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <<https://www.rfc-editor.org/info/rfc7552>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.

- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <<https://www.rfc-editor.org/info/rfc8529>>.

15. Informative References

- [I-D.ietf-mpls-mldp-yang] Raza, K., Liu, X., Esale, S., Andersson, L., Tantsura, J., and S. Krishnaswamy, "YANG Data Model for MPLS mLDP", Work in Progress, Internet-Draft, draft-ietf-mpls-mldp-yang-06, 31 May 2019, <<https://tools.ietf.org/html/draft-ietf-mpls-mldp-yang-06>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC7307] Zhao, Q., Raza, K., Zhou, C., Fang, L., Li, L., and D. King, "LDP Extensions for Multi-Topology", RFC 7307, DOI 10.17487/RFC7307, July 2014, <<https://www.rfc-editor.org/info/rfc7307>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

Appendix A. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951], containing both configuration and state data.

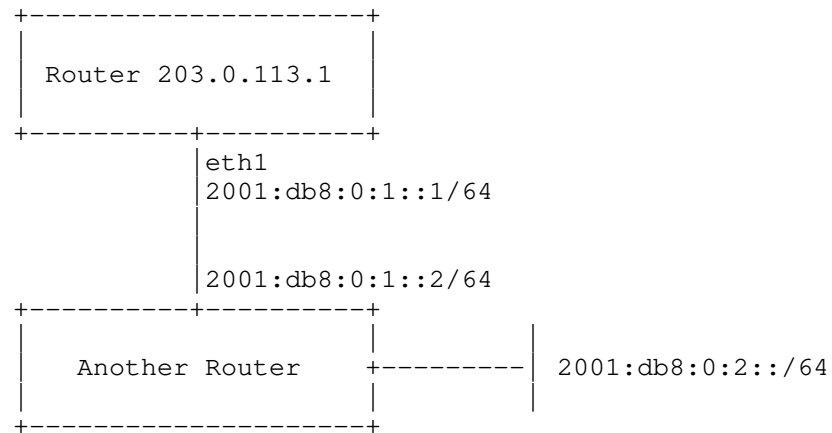


Figure 12: Example topology

The configuration instance data tree for Router 203.0.113.1 in the above figure could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "description": "An interface with LDP enabled.",
        "type": "iana-if-type:ethernetCsmacd",
        "ietf-ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:1::1",
              "prefix-length": 64
            }
          ]
        },
        "forwarding": true
      }
    ]
  },
  "ietf-routing:routing": {
    "router-id": "203.0.113.1",
    "control-plane-protocols": {

```



```

"control-plane-protocol": [
  {
    "type": "ietf-mpls-ldp:mpls-ldp",
    "name": "ldp-1",
    "ietf-mpls-ldp:mpls-ldp": {
      "global": {
        "address-families": {
          "ietf-mpls-ldp-extended:ipv6": {
            "enabled": true,
            "transport-address": "2001:db8:0:1::1"
          }
        }
      },
      "discovery": {
        "interfaces": {
          "interface": [
            {
              "name": "eth1",
              "address-families": {
                "ietf-mpls-ldp-extended:ipv6": {
                  "enabled": true
                }
              }
            }
          ]
        }
      }
    }
  }
]
}

```

Figure 13: Example Configuration data in JSON

The corresponding operational state data for Router 203.0.113.1 could be as follows:

```

{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth1",
        "description": "An interface with LDP enabled.",
        "type": "iana-if-type:ethernetCsmacd",
        "phys-address": "00:00:5e:00:53:01",

```

```
    "oper-status": "up",
    "statistics": {
      "discontinuity-time": "2018-09-10T15:16:27-05:00"
    },
    "ietf-ip:ipv6": {
      "forwarding": true,
      "mtu": 1500,
      "address": [
        {
          "ip": "2001:db8:0:1::1",
          "prefix-length": 64,
          "origin": "static",
          "status": "preferred"
        },
        {
          "ip": "fe80::200:5eff:fe00:5301",
          "prefix-length": 64,
          "origin": "link-layer",
          "status": "preferred"
        }
      ],
      "neighbor": [
        {
          "ip": "2001:db8:0:1::2",
          "link-layer-address": "00:00:5e:00:53:02",
          "origin": "dynamic",
          "is-router": [null],
          "state": "reachable"
        },
        {
          "ip": "fe80::200:5eff:fe00:5302",
          "link-layer-address": "00:00:5e:00:53:02",
          "origin": "dynamic",
          "is-router": [null],
          "state": "reachable"
        }
      ]
    }
  ],
  "ietf-routing:routing": {
    "router-id": "203.0.113.1",
    "interfaces": {
      "interface": [
        "eth1"
      ]
    }
  },
}
```

```

"control-plane-protocols": {
  "control-plane-protocol": [
    {
      "type": "ietf-mpls-ldp:mpls-ldp",
      "name": "ldp-1",
      "ietf-mpls-ldp:mpls-ldp": {
        "global": {
          "address-families": {
            "ietf-mpls-ldp-extended:ipv6": {
              "enabled": true,
              "transport-address": "2001:db8:0:1::1"
            }
          }
        },
        "discovery": {
          "interfaces": {
            "interface": [
              {
                "name": "eth1",
                "address-families": {
                  "ietf-mpls-ldp-extended:ipv6": {
                    "enabled": true,
                    "hello-adjacencies": {
                      "hello-adjacency": [
                        {
                          "adjacent-address":
                            "fe80::200:5eff:fe00:5302",
                          "flag": ["adjacency-flag-active"],
                          "hello-holdtime": {
                            "adjacent": 15,
                            "negotiated": 15,
                            "remaining": 9
                          },
                          "next-hello": 3,
                          "statistics": {
                            "discontinuity-time":
                              "2018-09-10T15:16:27-05:00"
                          },
                          "peer": {
                            "lsr-id": "203.0.113.2",
                            "label-space-id": 0
                          }
                        }
                      ]
                    }
                  }
                }
              ]
            }
          }
        }
      }
    }
  ]
}

```

```
]
}
},
"peers": {
  "peer": [
    {
      "lsr-id": "203.0.113.2",
      "label-space-id": 0,
      "label-advertisement-mode": {
        "local": "downstream-unsolicited",
        "peer": "downstream-unsolicited",
        "negotiated": "downstream-unsolicited"
      },
      "next-keep-alive": 5,
      "session-holdtime": {
        "peer": 180,
        "negotiated": 180,
        "remaining": 78
      },
      "session-state": "operational",
      "tcp-connection": {
        "local-address": "fe80::200:5eff:fe00:5301",
        "local-port": 646,
        "remote-address": "fe80::200:5eff:fe00:5302",
        "remote-port": 646
      },
      "up-time": 3438100,
      "statistics": {
        "discontinuity-time": "2018-09-10T15:16:27-05:00"
      }
    ]
  }
}
}
```

Figure 14: Example Operational data in JSON

Authors' Addresses

Kamran Raza (editor)
Cisco Systems

Canada
Email: skraza@cisco.com

Rajiv Asati
Cisco Systems
United States of America
Email: rajiva@cisco.com

Xufeng Liu
Volta Networks
United States of America
Email: xufeng.liu.ietf@gmail.com

Santosh Esale
Juniper Networks
United States of America
Email: sesale@juniper.net

Xia Chen
Huawei Technologies
China
Email: jescia.chenxia@huawei.com

Himanshu Shah
Ciena Corporation
United States of America
Email: hshah@ciena.com

MPLS WG
Internet-Draft
Intended status: Experimental
Expires: August 18, 2021

K. Kompella
Juniper Networks, Inc.
L. Contreras
Telefonica
February 14, 2021

Resilient MPLS Rings
draft-ietf-mpls-rmr-14

Abstract

This document describes the use of the MPLS control and data planes on ring topologies. It describes the special nature of rings, and proceeds to show how MPLS can be effectively used in such topologies. It describes how MPLS rings are configured, auto-discovered and signaled, as well as how the data plane works. Companion documents describe the details of discovery and signaling for specific protocols.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119][RFC8174].

This document is classified as an Experimental RFC. The parameters of this experiment have yet to be defined: how long the experiment runs, what criteria determine that the experiment is over -- does the doc then become Standards Track or Historical, etc. A future update will document these parameters.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Definitions	3
1.2. Changes from -12 in response to reviews	5
2. Motivation	6
3. Theory of Operation	6
3.1. Provisioning	7
3.2. Ring Nodes	7
3.3. Ring Links and Directions	8
3.3.1. Express Links	9
3.4. Ring LSPs	9
3.5. Installing Primary LFIB Entries	9
3.6. Protection	10
3.7. Installing FRR LFIB Entries	11
4. Autodiscovery	11
4.1. Overview	11
4.2. Ring Announcement Phase	13
4.3. Mastership Phase	14
4.4. Ring Identification Phase	14
4.5. Ring Changes	15
5. Ring OAM	16
6. Advanced Topics	16
6.1. Beyond the Ring	16
6.2. Half-rings	18
6.3. Hub Node Resilience	18
7. Security Considerations	18
8. Acknowledgments	19
9. IANA Considerations	19
10. References	19
10.1. Normative References	19
10.2. Informative References	19
Authors' Addresses	20

1. Introduction

Rings are a very common topology either at infrastructure level (e.g., physical ring fiber deployments in Layer 1 networks) or node interconnection structures (e.g., loops created in bridged interconnected infrastructures [IEEE.802.1D_2004]). A ring is the simplest topology offering link and node resilience. Rings are nearly ubiquitous in access and aggregation networks. As MPLS increases its presence in such networks, and takes on a greater role, it is imperative that MPLS handles rings well; this is not the case today.

This document describes the special nature of rings, and the special needs of MPLS on rings. It then shows how these needs can be met in several ways, some of which involve extensions to protocols such as IS-IS [RFC5305], OSPF [RFC3630], RSVP-TE [RFC3209] and LDP [RFC5036]. RMR LSPs can also be signaled with IGP [RFC8402]; that will be described in a future document.

The intent of this document is to handle rings that "occur naturally". Many access and aggregation networks in metros have their start as a simple ring. They may then grow into more complex topologies, for example, by adding parallel links to the ring, or by adding "express" links. The goal here is to discover these rings (with some guidance), and run MPLS over them efficiently. The intent is not to construct rings in a mesh network with the purpose of using them for protection.

In some other networking situations (e.g., interconnection of bridges), those rings could create loops making the network inoperable, and thus needing from signaling mechanisms (such the Spanning Tree Protocol) for preventing and eliminating such loops [IEEE.802.1D_2004]. Here it is followed a dual approach where the signaling methods are precisely created for automatically identifying and defining rings where efficiently create LSPs adapted to the formed ring topology.

1.1. Definitions

A (directed) graph $G = (V, E)$ consists of a set of vertices (or nodes) V and a set of edges (or links) E . An edge is an ordered pair of nodes (a, b) , where a and b are in V . (In this document, the terms node and link will be used instead of vertex and edge.)

A ring is a subgraph of G . A ring consists of a subset of n nodes $\{R_i, 0 \leq i < n\}$ of V . The directed edges $\{(R_i, R_{i+1}) \text{ and } (R_{i+1}, R_i), 0 \leq i < n-1\}$ must be a subset of E (note that index arithmetic is done modulo n). We define the direction from node R_i to R_{i+1} as

"clockwise" (CW) and the reverse direction as "anticlockwise" (AC). As there may be several rings in a graph, we number each ring with a distinct ring ID RID.



Figure 1: Ring with 8 nodes

The following terminology is used for ring LSPs:

Ring ID (RID): A non-negative number. When the RID identifies a ring, it must be positive and unique in some scope of a Service Provider's network. An RID of zero, when assigned to a node, indicates that the node must behave in "promiscuous mode" (see Section 3.2). A node may belong to multiple rings.

Ring node: A member of a ring. Note that a device may belong to several rings.

Node index: A logical numbering of nodes in a ring, from zero up to one less than the ring size. Used purely for exposition in this document.

Ring master: The ring master initiates the ring identification process. Mastership is indicated in the IGP by a two-bit field.

Ring neighbors: Nodes whose indices differ by one (modulo ring size).

Ring links: Links that connect ring neighbors.

Express links: Links that connect non-neighboring ring nodes.

Ring direction: A two-bit field in the IGP indicating the direction of a link. The choices are:

UN: 00 undefined link

CW: 01 clockwise ring link

AC: 10 anticlockwise ring link

EX: 11 express link

Ring Identification: The process of discovering ring nodes, ring links, link directions, and express links.

The following notation is used for ring LSPs:

R_k: A ring node with index k. R_k has AC neighbor R_(k-1) and CW neighbor R_(k+1).

RL_k: A (unicast) Ring LSP anchored on node R_k.

CL_jk: A label allocated by R_j for RL_k in the CW direction.

AL_jk: A label allocated by R_j for RL_k in the AC direction.

1.2. Changes from -12 in response to reviews

[Note to RFC Editor: this (sub-)section to be removed prior to publication.]

Reqs Lang: updated (response to Gen-ART review [Gen])

Section 1: updated "transport networks" to "Layer 1 networks" (response to Transport Area review [TAR])

Sec 1: replaced SPRING with IGP (response to OPS directorate [OPS])

Sec 1: rephrased last sentence [TAR]

Sec 2: added para on control plane resilience [TAR]

Sec 3.1: typo fixed [Gen]

Sec 3.2: added figure, caveats for promiscuous mode (response to Security Area Directorate review [SAD])

Sec 3.5: updated reference [OPS]

Sec 3.6: updated text on node protection, TTL [OPS]

Sec 4.1: changed Ring Neighbor TLV/flags to Ring Link TLV/flags; changed SPRING to IGP [OPS]

Sec 4.1: clean up [Gen]

Sec 4.2: updated text on timers T1, T2 [SAD]

Sec 4.3, 4.4: rewrote sections on Mastership, Ring Identification Phases for clarity [OPS]

Sec 4.5: removed "and" [Gen]

Sec 5: updated text on timers [TAR]

New Sec 6.1: added text on traffic transiting a ring [OPS]

Sec Cons: added text on compromised nodes [SAD]

2. Motivation

A ring is the simplest topology that offers resilience. This is perhaps the main reason to lay out fiber in a ring. Thus, effective mechanisms for fast failover on rings are needed. Furthermore, there are large numbers of rings. Thus, configuration of rings needs to be as simple as possible. Finally, bandwidth management on access rings is very important, as bandwidth is generally quite constrained here.

The goals of this document are to present mechanisms for improved MPLS-based resilience in ring networks (using ideas that are reminiscent of Bidirectional Line Switched Rings), for automatic bring-up of LSPs, better bandwidth management and for auto-hierarchy. These goals can be achieved using extensions to existing IGP and MPLS signaling protocols, using central provisioning, or in other ways.

Note that this document addresses data plane resilience. Control plane resilience, and robustness of protocol messaging, is managed by the protocols being used here (IS-IS, OSPF, LDP and RSVP-TE) and not described in this document.

3. Theory of Operation

Say a ring has ring ID RID. The ring is provisioned by choosing one or more ring masters for the ring and assigning them the RID. Other nodes in the ring may also be assigned this RID, or may be configured as "promiscuous". Ring discovery then kicks in. When each ring node knows its CW and AC ring neighbors and its ring links, and all express links have been identified, ring identification is complete.

Once ring identification is complete, each node signals one or more ring LSPs RL_i . RL_i , anchored on node R_i , consists of two counter-rotating unicast LSPs that start and end at R_i . A ring LSP is "multipoint": any node R_j can use RL_i to send traffic to R_i ; this can be in either the CW or AC directions, or both (i.e., load

balanced). Both of these counter-rotating LSPs are "active"; the choice of direction to send traffic to R_i is determined by policy at the node where traffic is injected into the ring. The default policy is to send traffic along the shortest path. Bidirectional connectivity between nodes R_i and R_j is achieved by using two different ring LSPs: R_i uses RL_j to reach R_j , and R_j uses RL_i to reach R_i .

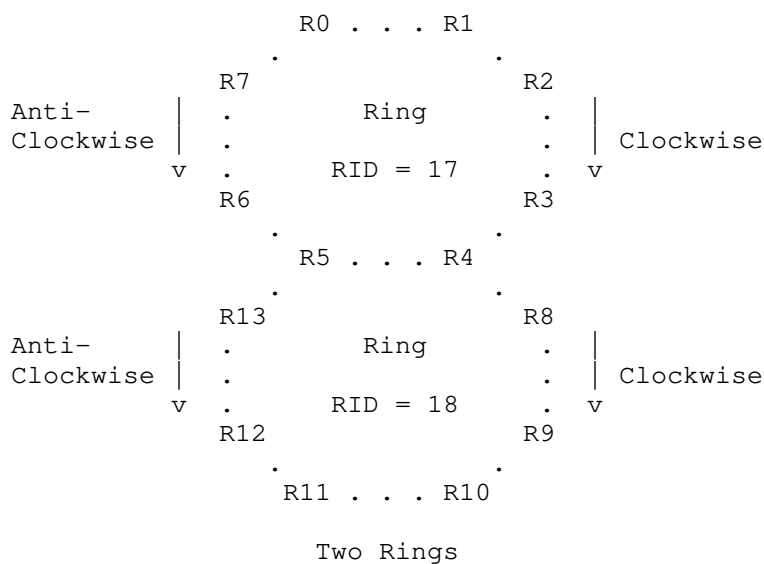
3.1. Provisioning

The goal here is to provision rings with the absolute minimum configuration. The exposition below aims to achieve that using auto-discovery via a link-state IGP (see Section 4). Of course, auto-discovery can be overridden by configuration. For example, a link that would otherwise be classified by auto-discovery as a ring link might be configured not to be used for ring LSPs.

3.2. Ring Nodes

Ring nodes have a loopback address, and run a link-state IGP and an MPLS signaling protocol. To provision a node as a ring node for ring RID, the node is simply assigned that RID. A node may be part of several rings, and thus may be assigned several ring IDs.

To simplify ring provisioning even further, a node N may be made "promiscuous" by being assigned an RID of 0. A promiscuous node listens to RIDs in its IGP neighbors' link-state updates. For every non-zero RID N hears from a neighbor, N joins the corresponding ring by taking on that RID. In many situations, the use of promiscuous mode means that only one or two nodes in a ring needs to be provisioned; everything else is auto-discovered. However, this feature should be used with care. Consider the following:



If R3 and R6 are configured with RID 17, R8 and R13 with RID 18, and all other nodes with RID 0, this will end up as two rings with R4 and R5 in both. However, other permutations of RID configurations could easily end up with all nodes being in both rings 17 and 18, whereupon the maximal ring will consist of R0 to R4, R8 to R13, R5 to R7 (and the link from R4 to R5 will be an express link). In cases such as these, one should eschew promiscuous mode in favor of simply configuring all nodes with the appropriate RIDs.

A ring node indicates in its IGP updates the ring LSP signaling protocols it supports. This can be LDP and/or RSVP-TE. Ideally, each node should support both.

3.3. Ring Links and Directions

Ring links must be MPLS-capable. They are by default unnumbered, point-to-point (from the IGP point of view) and "auto-bundled". The "auto-bundled" attribute means that parallel links between ring neighbors are considered as a single link, without the need for explicit configuration for bundling (such as a Link Aggregation Group). Note that each component may be advertised separately in the IGP; however, signaling messages and labels across one component link apply to all components. Parallel links between a pair of ring nodes is often the result of having multiple lambdas or fibers between those nodes. RMR is primarily intended for operation at the packet layer; however, parallel links at the lambda or fiber layer may result in parallel links at the packet layer.

A ring link is not provisioned as belonging to the ring; it is discovered to belong to ring RID if both its adjacent nodes belong to RID. A ring link's direction (CW or AC) is also discovered; this process is initiated by the ring's ring master. Note that the above two attributes can be overridden by provisioning if needed; it is then up to the provisioning system to maintain consistency across the ring.

3.3.1. Express Links

Express links are discovered once ring nodes, ring links and directions have been established. As defined earlier, express links are links joining non-neighboring ring nodes; often, this may be the result of optically bypassing ring nodes.

3.4. Ring LSPs

Ring LSPs are not provisioned. Once a ring node R_i knows its RID, its ring links and directions, it kicks off ring LSP signaling automatically. R_i allocates CW and AC labels for each ring LSP RL_k . R_i also initiates the creation of RL_i . As the signaling propagates around the ring, CW and AC labels are exchanged. When R_i receives CW and AC labels for RL_k from its ring neighbors, primary and fast reroute (FRR) paths for RL_k are installed at R_i .

For RSVP-TE LSPs, bandwidths may be signaled in both directions. However, these are not provisioned either; rather, one does "reverse call admission control". When a service needs to use an LSP, the ring node where the traffic enters the ring attempts to increase the bandwidth on the LSP to the egress. If successful, the service is admitted to the ring.

3.5. Installing Primary LFIB Entries

In setting up RL_k , a node R_j sends out two labels: CL_{jk} to R_{j-1} and AL_{jk} to R_{j+1} . R_j also receives two labels: $CL_{j+1,k}$ from R_{j+1} , and $AL_{j-1,k}$ from R_{j-1} . R_j can now set up the forwarding entries for RL_k . In the CW direction, R_j swaps incoming label CL_{jk} with $CL_{j+1,k}$ with next hop R_{j+1} ; these allow R_j to act as LSR for RL_k . R_j also installs an LFIB entry to push $CL_{j+1,k}$ with next hop R_{j+1} to act as ingress for RL_k . Similarly, in the AC direction, R_j swaps incoming label AL_{jk} with $AL_{j-1,k}$ with next hop R_{j-1} (as LSR), and an entry to push $AL_{j-1,k}$ with next hop R_{j-1} (as ingress).

Clearly, R_k does not act as ingress for its own LSPs. However, R_k can send OAM messages, for example, an MPLS ping or traceroute ([RFC8029]), using labels $CL_{k,k+1}$ and $AL_{k-1,k}$, to test the entire

ring LSP anchored at R_k in both directions. Furthermore, if these LSPs use Ultimate Hop Popping, then R_k installs LFIB entries to pop CL_k,k for packets received from R_{k-1} and to pop AL_k,k for packets received from R_{k+1} .

3.6. Protection

In this scheme, there are no protection LSPs as such -- no node or link bypass LSPs, no standby LSPs, no detours, and no LFA-type protection. Protection is via the "other" direction around the ring, which is why ring LSPs are in counter-rotating pairs. Protection works in the same way for link, node and ring LSP failures.

If a node R_j detects a failure from R_{j+1} -- either all links to R_{j+1} fail, or R_{j+1} itself fails, R_j switches traffic on all CW ring LSPs to the AC direction using the FRR LFIB entries. If the failure is specific to a single ring LSP, R_j switches traffic just for that LSP. In either case, this switchover can be very fast, as the FRR LFIB entries can be preprogrammed. Fast detection and fast switchover lead to minimal traffic loss.

R_j then sends an indication to R_{j-1} that the CW direction is not working, so that R_{j-1} can similarly switch traffic to the AC direction. For RSVP-TE, this indication can be a PathErr or a Notify; other signaling protocols have similar indications. These indications propagate AC until each traffic source on the ring AC of the failure uses the AC direction. Thus, within a short period, traffic will be flowing in the optimal path, given that there is a failure on the ring. This contrasts with (say) bypass protection, where until the ingress recomputes a new path, traffic will be suboptimal.

Note that the failure of a node or a link will not necessarily affect all ring LSPs. Thus, it is important to identify the affected LSPs (and switch them), but to leave the rest alone.

One point to note is that when a ring node, say R_j , fails, RL_j is clearly unusable. However, the above protection scheme will cause a traffic loop: R_{j-1} detects a failure CW, and protects by sending CW traffic on RL_j back all the way to R_{j+1} , which in turn sends traffic to R_{j-1} , etc. There are three proposals to avoid this:

1. Each ring node acting as ingress sends traffic with a TTL of at most $2*n$, where n is the number of nodes in the ring.
2. A ring node sends protected traffic (i.e., traffic switched from CW to AC or vice versa) with TTL just large enough to reach the egress.

3. A ring node sends protected traffic with a special purpose label below the ring LSP label. A protecting node first checks for the presence of this label; if present, it means that the traffic is looping and MUST be dropped.

Approaches 1 and 2 work for traffic that remains on the ring or terminates on a ring node (see Section 6.1); for traffic transiting the ring, playing with TTL may affect forwarding beyond the ring. Approach 3 is the most general and is the one we advocate; however, this will require the allocation and definition of a new special purpose label.

3.7. Installing FRR LFIB Entries

At the same time that R_j sets up its primary CW and AC LFIB entries, it can also set up the protection forwarding entries for RL_k. In the CW direction, R_j sets up an FRR LFIB entry to swap incoming label CL_jk with AL_{j-1,k} with next hop R_{j-1}. In the AC direction, R_j sets up an FRR LFIB entry to swap incoming label AL_jk with CL_{j+1,k} with next hop R_{j+1}. Again, R_k does not install FRR LFIB entries in this manner.

Say R1 receives label L42 from R2 to reach R4 in the clockwise direction, and receives label L40 from R0 to reach R4 in the anti-clockwise direction. Say R1 also receives label L52 from R2 to reach R5 in the clockwise direction, and receives label L50 from R0 to reach R5 in the anti-clockwise direction. R1 makes the following LFIB entries:

Dest	CW/NH	CW FRR/NH	AC/NH	AC FRR/NH
...				
R4	L42/R2	L40/R0	L40/R0	L42/R2
R5	L52/R2	L50/R0	L50/R0	L52/R2
...				

R1's LFIB

4. Autodiscovery

4.1. Overview

Auto-discovery proceeds in three phases. The first phase is the announcement phase. The second phase is the mastership phase. The third phase is the ring identification phase.

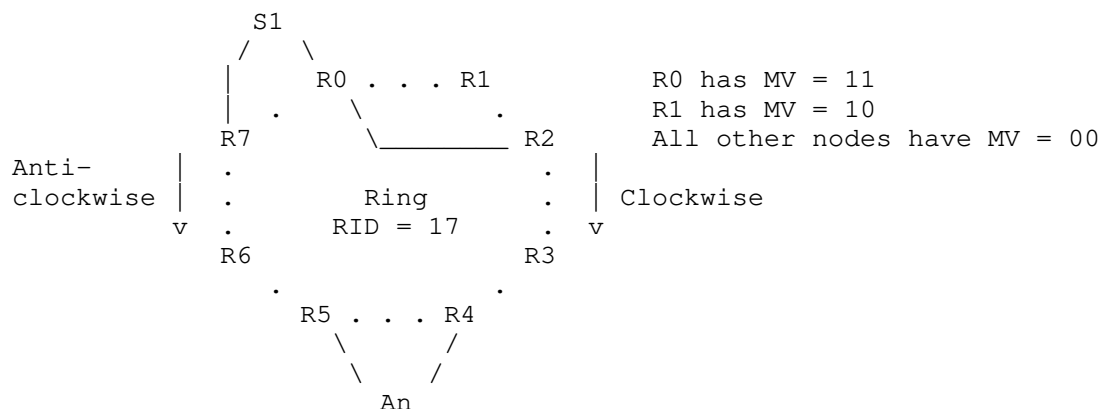


Figure 2: Ring with non-ring nodes and links

We use three concepts below:

ring nodes: all nodes that announce ring node TLVs with a given RID.

IGP neighbors: all nodes which are IGP neighbors of a given node.

ring neighbors: ring nodes that are IGP neighbors of a given node. Exactly one is the CW neighbor and one is the AC neighbor; all other ring neighbors are express neighbors.

In Figure 2, R0 through R7 are ring nodes belonging to ring 17. R0 has IGP neighbors R1, R2, R7 and S1. R0 has ring neighbors R1 (CW), R2 (express) and R7 (AC). Autodiscovery aims to identify ring nodes of a given ring, ring neighbors of each ring node, and the CW and AC node for each ring node.

The format of an RMR Node Type-Length-Value (TLV) is given below. It consists of information pertaining to the node and optionally, sub-TLVs. A Neighbor sub-TLV contains information pertaining to the node's neighbors. Other sub-TLVs may be defined in the future. Details of the format specific to IS-IS and OSPF will be given in the corresponding IGP documents.

[RMR Node Type][RMR Node Length][RID][Node Flags][sub-TLVs]

Ring Node TLV Format

[My Intf Inx][Rem Intf Inx][RID 1][Flags for RID 1]
 [RID 2][Flags for RID 2]...

Ring Link Sub-TLV Format

```

      0                                     1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+
|MV | SS | SO | MBZ | SU | M |
+---+---+---+---+---+---+---+---+
MV: Mastership Value
SS: Supported Signaling Protocols
    (100 = RSVP-TE; 010 = LDP; 001 = IGP)
MBZ: Must be zero
SO: Supported OAM Protocols (100 = BFD; 010 = CFM; 001 = EFM)
SU: Signaling Protocol to Use (00: none; 01: LDP; 10: RSVP-TE;
    11: IGP)
M : Elected Master (0 = no, 1 = yes)

```

Flags for a Ring Node TLV

```

      0                                     1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+
|RD |OAM| MBZ |
+---+---+---+---+---+---+---+---+
RD: Ring Direction (00 = none; 01 = CW; 10 = AC; 11 = express)
OAM: OAM Protocol to use (00 = none; 01 = BFD; 10 = CFM; 11 = EFM)
MBZ: Must be zero

```

Flags for a Ring Link TLV

4.2. Ring Announcement Phase

Each node participating in an MPLS ring is assigned an RID; in the example, RID = 17. A node is also provisioned with a mastership value. Each node advertises a ring node TLV for each ring it is participating in, along with the associated flags. It then starts timer T1; this timer is to allow each node time to hear from all other nodes in the ring. [The settings for timers T1 and T2 (below) are particular to the specific IGP used for signaling; they will be discussed in the IGP document that defines the ring node/link TLVs.] The settings for timers T1 and T2 (below) will be discussed in the IGP document that defines the ring node/link TLVs.]

A node in promiscuous mode doesn't advertise any ring node TLVs. However, when it hears a ring node TLV from an IGP neighbor, it joins that ring, and sends its own ring node TLV with that RID.

The announcement phase allows a ring node to discover other ring nodes in the same ring so that a ring master can be elected.

4.3. Mastership Phase

When timer T1 fires, a node enters the mastership phase. In this phase, each ring node N starts timer T2 and checks if it is master as follows. N examines the MV value of all ring nodes and selects those with the highest MV value. Among these nodes, N finds the node with the lowest loopback address. If that node is N, N declares itself master to the entire ring by readvertising its ring node TLV with the M bit set.

When timer T2 fires, each node examines the ring node TLVs from all other nodes in the ring to identify the ring master. There should be exactly one; if not, each node restarts timer T2 and tries again.

Barring software bugs or malicious code, the principal reason for multiple nodes for setting their M bit is late-arriving ring announcements. Say nodes N1 and N2 have the highest mastership values, and N1 has the lowest loopback address, while N2 has the second lowest loopback address. If N1 makes its ring announcement just as N2's T1 timer fires, both N1 and N2 will think they are the master (since N2 will not have heard N1's announcement in time). However, in the next round, N2 will realize that N1 is indeed the master. In the worst case, the mastership phase will occur as many times as there are nodes in the ring.

4.4. Ring Identification Phase

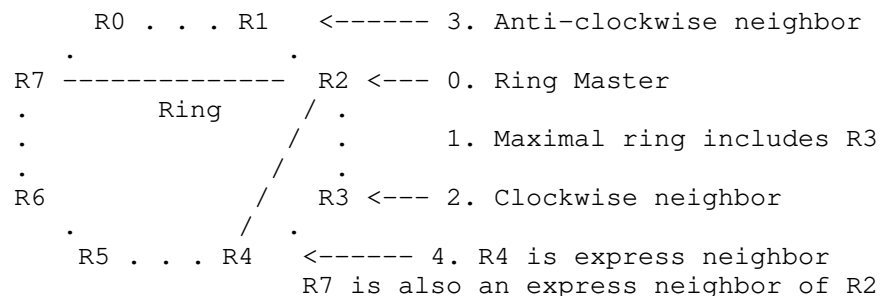


Figure 3: Ring Identification

When there is exactly one ring master M (here, R2), M enters the Ring Identification Phase. M indicates that it has successfully completed this phase by advertising ring link TLVs. This is the trigger for

M's CW neighbor to enter the Ring Identification Phase. This phase passes CW until all ring nodes have completed ring identification.

The Ring Identification Phase proceeds as follows:

1. M identifies all ring nodes for ring RID, i.e., those that have announced ring node TLVs with the ring ID = RID.
2. M computes a maximal ring among these nodes.
3. Based on that, M picks a CW neighbor and an AC neighbor.
4. M then inserts ring link TLVs with ring direction CW for each link to its CW neighbor; M also inserts a ring link TLV with direction AC for each link to its AC neighbor. (Note that there may be multiple links from M to each of its neighbors.)
5. Finally, M determines its express links. These are links to IGP neighbors that are ring nodes but neither the CW or AC neighbor. M advertises ring link TLVs for express links by setting the link direction to "express link".

This process passes on to the CW neighbor X as follows:

1. Each node Y listens for ring link TLVs. The set of nodes S consists of those that have announced ring link TLVs.
2. If a node Z announces a ring link TLV with Y as the CW neighbor, then Y is next.

X follows the same procedure as M with two small changes:

1. when X computes a maximal ring, it MUST include all nodes in S.
2. X knows its AC neighbor (Z above), and doesn't have to pick it.

Here, R2 (the master) knows R0 through R7 are ring nodes (Step 1). R1, R3, R4 and R7 are its ring neighbors. R2 computes a maximal ring (Step 2). It then picks R3 as its CW neighbor and R1 as its AC neighbor (Step 3). Finally, it declares the links to R4 and R7 as express links (Step 5).

4.5. Ring Changes

The main changes to a ring are:

ring link addition;

ring link deletion;
ring node addition;
ring node deletion.

The main goal of handling ring changes is (as much as possible) not to perturb existing ring operation. Thus, if the ring master hasn't changed, all of the above changes should be local to the point of change. Link adds just update the IGP; signaling should take advantage of the new capacity as soon as it learns. Link deletions in the case of parallel links also show up as a change in capacity (until the last link in the bundle is removed.)

The removal of the last ring link between two nodes, or the removal of a ring node is an event that triggers protection switching. In a simple ring, the result is a broken ring. However, if a ring has express links, then it may be able to converge to a smaller ring with protection.

The addition of a new ring node can also be handled incrementally.

5. Ring OAM

Each ring node should advertise in its ring node TLV the OAM protocols it supports. Each ring node is expected to run a link-level OAM over each ring link. This should be an OAM protocol that both neighbors agree on. The default hello time is that of the protocol chosen.

Each ring node also sends OAM messages over each direction of its ring LSP. This is a multi-hop OAM to check LSP liveness; typically, BFD would be used for this. Each node chooses the hello interval, the choice of which should be based on the size of the ring (as each node would have to send out twice that many hello messages every interval) and the desired failure detection time.

6. Advanced Topics

6.1. Beyond the Ring

The discourse above discusses traffic that originates and terminates on a ring. However, in many cases, traffic may come originate on a ring node and terminate at a non-ring node; other traffic may originate on a non-ring node and terminate on a ring node; and in yet other cases, traffic may transit a ring, i.e., originate on a non-ring node, arrive at a ring node, traverse the ring, and leave for a

non-ring destination. This section discusses these cases, and how traffic traversing a ring can profit from ring protection.

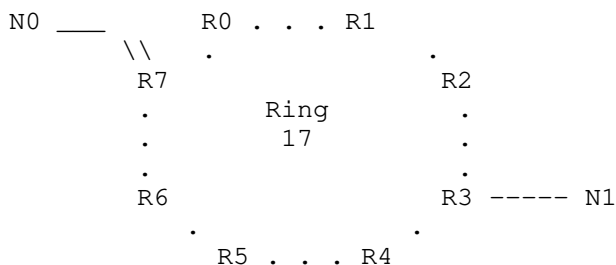


Figure 4: Beyond the Ring

In all these cases, the "end-to-end" path needs to be either stitched with, or overlaid on, the ring path. The latter approach is recommended, using hierarchy in both the control and data planes. In the figure above, traffic from N0 to N1 (both non-ring nodes) traverses Ring 17. If nodes outside Ring 17 use LDP to signal LSPs, here's one way to accomplish this: R7 and R3 have targeted LDP sessions to exchange labels. The following LDP label exchanges occur (among others):

1. N1 sends an "egress label" L0 for its loopback N1 to R3 and inserts a "pop L0 and forward" entry in its LFIB.
2. R3 sends a label L1 for N1 to R7 over the targeted LDP session and inserts a "swap L1 with L0" in its LFIB.
3. R7 sends label L2 for N1 to N0 and inserts a "swap L2 with L1" entry in its LFIB.
4. N0 inserts a "push L2" entry in its LFIB for traffic destined to N1.

In parallel, nodes in Ring 17 exchange labels for traffic within the ring.

To send a packet to N1, N0 pushes label L2. When this reaches R7, R7 swaps L2 with L1 and additionally pushes a ring label to reach R3. Ring forwarding occurs between R7 and R3. R3 pops the ring label, swaps L2 with L1 and forwards the packet to N1. If a failure occurs on the ring, ring protection kicks in. A failure of R7, R3 or any non-ring node will be dealt with by the non-ring label distribution protocol (in this case, LDP).

6.2. Half-rings

In some cases, a ring H may be incomplete, either because H is permanently missing a link (not just because of a failure), or because the link required to complete H is in a different IGP area. Either way, the ring discovery algorithm will fail. We call such a ring a "half-ring". Half-rings are sufficiently common that finding a way to deal with them effectively is a useful problem to solve. This topic will not be addressed in this document; that task is left for a future document.

6.3. Hub Node Resilience

Let's call the node(s) that connect a ring to the rest of the network "hub node(s)" (usually, there are a pair of hub nodes.) Suppose a ring has two hub nodes H1 and H2. Suppose further that a non-hub ring node X wants to send traffic to some node Z outside the ring. This could be done, say, by having targeted LDP (T-LDP) sessions from H1 and H2 to X advertising LDP reachability to Z via H1 (H2); there would be a two-label stack from X to reach Z. Say that to reach Z, X prefers H1; thus, traffic from X to Z will first go to H1 via a ring LSP, then to Z via LDP.

If H1 fails, traffic from X to Z will drop until the T-LDP session from H1 to Z fails, the IGP reconverges, and H2's label to Z is chosen. Thereafter, traffic will go from X to H2 via a ring LSP, then to Z via LDP. However, this convergence could take a long time. Since this is a very common and important situation, it is again a useful problem to solve. However, this topic too will not be addressed in this document; that task is left for a future document.

7. Security Considerations

This document proposes extensions to IS-IS, OSPF, LDP and RSVP-TE, all of which have mechanisms to secure them. The extensions proposed do not represent per se a compromise to network security when the control plane is secured, since any manipulation of the content of the messages or even the control plane misinterpretation of the semantics are avoided.

A compromised or otherwise misbehaving node can foil the autodiscovery process Section 4, leading to a ring never transitioning to a usable state.

8. Acknowledgments

Many thanks to Pierre Bichon whose exemplar of self-organizing networks and whose urging for ever simpler provisioning led to the notion of promiscuous nodes.

9. IANA Considerations

There are no requests as yet to IANA for this document.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

10.2. Informative References

- [IEEE.802.1D_2004] IEEE, "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges", IEEE 802.1D-2004, DOI 10.1109/ieeestd.2004.94569, July 2004, <<http://ieeexplore.ieee.org/servlet/opac?punumber=9155>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <<https://www.rfc-editor.org/info/rfc5036>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.

- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

Authors' Addresses

Kireeti Kompella
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: kireeti.ietf@gmail.com

Luis M. Contreras
Telefonica
Ronda de la Comunicacion
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: <http://lmcontreras.com>

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2017

D. Dhody, Ed.
Huawei Technologies
J. Hardwick
Metaswitch
V. Beeram
Juniper Networks
J. Tantsura
March 12, 2017

A YANG Data Model for Path Computation Element Communications Protocol
(PCEP)
draft-ietf-pce-pcep-yang-02

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration data and state data (status information and counters for the collection of statistics).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 13, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. Terminology and Notation	3
3.1. Tree Diagrams	4
3.2. Prefixes in Data Node Names	5
4. Objectives	6
5. The Design of PCEP Data Model	6
5.1. The Entity	17
5.2. The Peer Lists	17
5.3. The Session Lists	18
5.4. Notifications	18
5.5. RPC	19
6. The Design of PCEP Statistics Data Model	19
7. Advanced PCE Features	22
7.1. Stateful PCE's LSP-DB	22
8. Open Issues and Next Step	22
8.1. The PCE-Initiated LSP	23
8.2. PCEP over TLS (PCEPS)	23
9. PCEP YANG Modules	23
9.1. ietf-pcep module	23
9.2. ietf-pcep-stats module	83
10. Security Considerations	103
11. IANA Considerations	104
12. Acknowledgements	105
13. References	105
13.1. Normative References	105
13.2. Informative References	106
Appendix A. Contributor Addresses	108
Authors' Addresses	109

1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE). [I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC6020] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data (status information and counters for the collection of statistics).

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- o Path Computation Request message (PCReq).
- o Path Computation Reply message (PCRep).
- o Notification message (PCNtf).
- o Error message (PCErr).
- o Request Parameters object (RP).
- o Synchronization Vector object (SVEC).
- o Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- o PCEP entity: a local PCEP speaker.
- o PCEP peer: to refer to a remote PCEP speaker.

- o PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [I-D.ietf-pce-stateful-pce] :

- o Stateful PCE, Passive Stateful PCE, Active Stateful PCE
- o Delegation, Revocation, Redelegation
- o LSP State Report, Path Computation Report message (PCRpt).
- o LSP State Update, Path Computation Update message (PCUpd).

[I-D.ietf-pce-pce-initiated-lsp] :

- o PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[I-D.ietf-pce-lsp-setup-type] :

- o Path Setup Type (PST).

[I-D.ietf-pce-segment-routing] :

- o Segment Routing (SR).

3.1. Tree Diagrams

A graphical representation of the complete data tree is presented in Section 5. The meaning of the symbols in these diagrams is as follows and as per [I-D.ietf-netmod-rfc6087bis]. Each node is printed as:

<status> <flags> <name> <opts> <type> <if-features>

<status> is one of:

- + for current
- x for deprecated
- o for obsolete

<flags> is one of:

- rw for configuration data
- ro for non-configuration data
- x for rpcs and actions
- n for notifications

<name> is the name of the node

- (<name>) means that the node is a choice node
- :(<name>) means that the node is a case node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>.

<opts> is one of:

- ? for an optional leaf, choice, anydata or anyxml
- ! for a presence container
- * for a leaf-list or list
- [<keys>] for a list's keys

<type> is the name of the type for leafs and leaf-lists

If the type is a leafref, the type is printed as "-> TARGET", where TARGET is either the leafref path, with prefixed removed if possible.

<if-features> is the list of features this node depends on, printed within curly brackets and a question mark "{...}?"

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
te	ietf-te	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te]
key-chain	ietf-key-chain	[I-D.ietf-rtgwg-yang-key-chain]
nacm	ietf-netconf-acm	[RFC6536]

Table 1: Prefixes and corresponding YANG modules

4. Objectives

This section describes some of the design objectives for the model:

- o In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.
- o The data model should be suitable for new implementations to use as is.
- o Mapping to the PCEP MIB Module should be clear.
- o The data model should allow for static configurations of peers.
- o The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors. This could be in a separate model which augments the base data model.
- o It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker.

```

module: ietf-pcep
  +--rw pcep!
  |   +--rw entity
  |   |   +--rw addr                               inet:ip-address
  |   |   +--rw enabled?                          boolean

```

```

+--rw role                                pcep-role
+--rw description?                        string
+--rw speaker-entity-id?                  string
|   {stateful-sync-opt}?
+--rw domain
|   +--rw domain* [domain-type domain]
|       +--rw domain-type    domain-type
|       +--rw domain         domain
+--rw capability
|   +--rw gmpls?                boolean {gmpls}?
|   +--rw bi-dir?               boolean
|   +--rw diverse?              boolean
|   +--rw load-balance?         boolean
|   +--rw synchronize?         boolean {svec}?
|   +--rw objective-function?   boolean
|       {objective-function}?
|   +--rw add-path-constraint?   boolean
|   +--rw prioritization?        boolean
|   +--rw multi-request?        boolean
|   +--rw gco?                  boolean {gco}?
|   +--rw p2mp?                 boolean {p2mp}?
|   +--rw stateful {stateful}?
|       +--rw enabled?          boolean
|       +--rw active?           boolean
|       +--rw pce-initiated?    boolean
|           {pce-initiated}?
|       +--rw include-db-ver?    boolean
|           {stateful-sync-opt}?
|       +--rw trigger-resync?     boolean
|           {stateful-sync-opt}?
|       +--rw trigger-initial-sync? boolean
|           {stateful-sync-opt}?
|       +--rw incremental-sync?   boolean
|           {stateful-sync-opt}?
|   +--rw sr {sr}?
|       +--rw enabled?          boolean
+--rw pce-info
|   +--rw scope
|       +--rw intra-area-scope?    boolean
|       +--rw intra-area-pref?     uint8
|       +--rw inter-area-scope?    boolean
|       +--rw inter-area-scope-default? boolean
|       +--rw inter-area-pref?     uint8
|       +--rw inter-as-scope?      boolean
|       +--rw inter-as-scope-default? boolean
|       +--rw inter-as-pref?       uint8
|       +--rw inter-layer-scope?   boolean
|       +--rw inter-layer-pref?    uint8

```



```

+--rw neigh-domains
|   +--rw domain* [domain-type domain]
|       +--rw domain-type    domain-type
|       +--rw domain         domain
+--rw path-key {path-key}?
|   +--rw enabled?           boolean
|   +--rw discard-timer?     uint32
|   +--rw reuse-time?        uint32
|   +--rw pce-id?            inet:ip-address
+--rw (auth-type-selection)?
|   +--:(auth-key-chain)
|   |   +--rw key-chain?
|   |   |   key-chain:key-chain-ref
|   +--:(auth-key)
|   |   +--rw crypto-algorithm          identityref
|   |   +--rw key-string
|   |   |   +--rw (key-string-style)?
|   |   |   |   +--:(keystring)
|   |   |   |   |   +--rw keystring?          string
|   |   |   |   +--:(hexadecimal)
|   |   |   |   |   {key-chain:hex-key-string}?
|   |   |   +--rw hexadecimal-string?    yang:hex-string
|   +--:(auth-tls) {tls}?
|   |   +--rw tls
+--rw connect-timer?          uint32
+--rw connect-max-retry?      uint32
+--rw init-backoff-timer?     uint32
+--rw max-backoff-timer?      uint32
+--rw open-wait-timer?        uint32
+--rw keep-wait-timer?        uint32
+--rw keep-alive-timer?       uint32
+--rw dead-timer?             uint32
+--rw allow-negotiation?      boolean
+--rw max-keep-alive-timer?    uint32
+--rw max-dead-timer?         uint32
+--rw min-keep-alive-timer?    uint32
+--rw min-dead-timer?         uint32
+--rw sync-timer?             uint32 {svec}?
+--rw request-timer?          uint32
+--rw max-sessions?           uint32
+--rw max-unknown-reqs?       uint32
+--rw max-unknown-msgs?       uint32
+--rw pcep-notification-max-rate uint32
+--rw stateful-parameter {stateful}?
|   +--rw state-timeout?       uint32
|   +--rw redelegation-timeout? uint32
|   +--rw rpt-non-pcep-lsp?    boolean
+--rw of-list {objective-function}?

```

```

|   +--rw objective-function* [of]
|   |   +--rw of      objective-function
+--rw peers
|   +--rw peer* [addr]
|   |   +--rw addr      inet:ip-address
|   |   +--rw description? string
|   |   +--rw domain
|   |   |   +--rw domain* [domain-type domain]
|   |   |   |   +--rw domain-type domain-type
|   |   |   |   +--rw domain domain
|   |   +--rw capability
|   |   |   +--rw gmpls? boolean {gmpls}?
|   |   |   +--rw bi-dir? boolean
|   |   |   +--rw diverse? boolean
|   |   |   +--rw load-balance? boolean
|   |   |   +--rw synchronize? boolean {svec}?
|   |   |   +--rw objective-function? boolean
|   |   |   |   {objective-function}?
|   |   |   +--rw add-path-constraint? boolean
|   |   |   +--rw prioritization? boolean
|   |   |   +--rw multi-request? boolean
|   |   |   +--rw gco? boolean {gco}?
|   |   |   +--rw p2mp? boolean {p2mp}?
|   |   +--rw stateful {stateful}?
|   |   |   +--rw enabled? boolean
|   |   |   +--rw active? boolean
|   |   |   +--rw pce-initiated? boolean
|   |   |   |   {pce-initiated}?
|   |   |   +--rw include-db-ver? boolean
|   |   |   |   {stateful-sync-opt}?
|   |   |   +--rw trigger-resync? boolean
|   |   |   |   {stateful-sync-opt}?
|   |   |   +--rw trigger-initial-sync? boolean
|   |   |   |   {stateful-sync-opt}?
|   |   |   +--rw incremental-sync? boolean
|   |   |   |   {stateful-sync-opt}?
|   |   +--rw sr {sr}?
|   |   |   +--rw enabled? boolean
+--rw scope
|   +--rw intra-area-scope? boolean
|   +--rw intra-area-pref? uint8
|   +--rw inter-area-scope? boolean
|   +--rw inter-area-scope-default? boolean
|   +--rw inter-area-pref? uint8
|   +--rw inter-as-scope? boolean
|   +--rw inter-as-scope-default? boolean
|   +--rw inter-as-pref? uint8
|   +--rw inter-layer-scope? boolean

```

```

|      |--rw inter-layer-pref?          uint8
+--rw neigh-domains
|      |--rw domain* [domain-type domain]
|      |      |--rw domain-type      domain-type
|      |      |--rw domain          domain
+--rw delegation-pref?      uint8 {stateful}?
+--rw (auth-type-selection)?
+--:(auth-key-chain)
|      |--rw key-chain?
|      |      key-chain:key-chain-ref
+--:(auth-key)
|      |--rw crypto-algorithm      identityref
|      |--rw key-string
|      |      |--rw (key-string-style)?
|      |      |      |--:(keystring)
|      |      |      |      |--rw keystring?          string
|      |      |      |      |--:(hexadecimal)
|      |      |      |      {key-chain:hex-key-string}?
|      |      |      |--rw hexadecimal-string?
|      |      |      |      yang:hex-string
+--:(auth-tls) {tls}?
+--rw tls

+--ro pcep-state
+--ro entity
+--ro addr?          inet:ip-address
+--ro index?         uint32
+--ro admin-status?  pcep-admin-status
+--ro oper-status?   pcep-admin-status
+--ro role?          pcep-role
+--ro description?   string
+--ro speaker-entity-id? string
|      {stateful-sync-opt}?
+--ro domain
|      |--ro domain* [domain-type domain]
|      |      |--ro domain-type      domain-type
|      |      |--ro domain          domain
+--ro capability
|      |--ro gmpls?          boolean {gmpls}?
|      |--ro bi-dir?        boolean
|      |--ro diverse?        boolean
|      |--ro load-balance?    boolean
|      |--ro synchronize?    boolean {svec}?
|      |--ro objective-function? boolean
|      |      {objective-function}?
|      |--ro add-path-constraint? boolean
|      |--ro prioritization?  boolean
|      |--ro multi-request?   boolean
|      |--ro gco?            boolean {gco}?

```

```

|   +--ro p2mp?                               boolean {p2mp}?
|   +--ro stateful {stateful}?
|   |   +--ro enabled?                         boolean
|   |   +--ro active?                         boolean
|   |   +--ro pce-initiated?                   boolean
|   |   |   {pce-initiated}?
|   |   +--ro include-db-ver?                 boolean
|   |   |   {stateful-sync-opt}?
|   |   +--ro trigger-resync?                 boolean
|   |   |   {stateful-sync-opt}?
|   |   +--ro trigger-initial-sync?           boolean
|   |   |   {stateful-sync-opt}?
|   |   +--ro incremental-sync?               boolean
|   |   |   {stateful-sync-opt}?
|   +--ro sr {sr}?
|   |   +--ro enabled?    boolean
+--ro pce-info
|   +--ro scope
|   |   +--ro intra-area-scope?                boolean
|   |   +--ro intra-area-pref?                uint8
|   |   +--ro inter-area-scope?               boolean
|   |   +--ro inter-area-scope-default?       boolean
|   |   +--ro inter-area-pref?                uint8
|   |   +--ro inter-as-scope?                 boolean
|   |   +--ro inter-as-scope-default?         boolean
|   |   +--ro inter-as-pref?                  uint8
|   |   +--ro inter-layer-scope?              boolean
|   |   +--ro inter-layer-pref?               uint8
|   +--ro neigh-domains
|   |   +--ro domain* [domain-type domain]
|   |   |   +--ro domain-type    domain-type
|   |   |   +--ro domain        domain
|   +--ro path-key {path-key}?
|   |   +--ro enabled?            boolean
|   |   +--ro discard-timer?     uint32
|   |   +--ro reuse-time?        uint32
|   |   +--ro pce-id?            inet:ip-address
+--ro (auth-type-selection)?
|   +--:(auth-key-chain)
|   |   +--ro key-chain?
|   |   |   key-chain:key-chain-ref
|   +--:(auth-key)
|   |   +--ro crypto-algorithm    identityref
|   |   +--ro key-string
|   |   |   +--ro (key-string-style)?
|   |   |   |   +--:(keystring)
|   |   |   |   |   +--ro keystring?            string
|   |   |   |   |   +--:(hexadecimal)

```

```

| | | | {key-chain:hex-key-string}?
| | | | +--ro hexadecimal-string? yang:hex-string
| | | +--:(auth-tls) {tls}?
| | | | +--ro tls
+--ro connect-timer? uint32
+--ro connect-max-retry? uint32
+--ro init-backoff-timer? uint32
+--ro max-backoff-timer? uint32
+--ro open-wait-timer? uint32
+--ro keep-wait-timer? uint32
+--ro keep-alive-timer? uint32
+--ro dead-timer? uint32
+--ro allow-negotiation? boolean
+--ro max-keep-alive-timer? uint32
+--ro max-dead-timer? uint32
+--ro min-keep-alive-timer? uint32
+--ro min-dead-timer? uint32
+--ro sync-timer? uint32 {svec}?
+--ro request-timer? uint32
+--ro max-sessions? uint32
+--ro max-unknown-reqs? uint32
+--ro max-unknown-msgs? uint32
+--ro stateful-parameter {stateful}?
| | +--ro state-timeout? uint32
| | +--ro redelegation-timeout? uint32
| | +--ro rpt-non-pcep-lsp? boolean
+--ro lsp-db {stateful}?
| | +--ro db-ver? uint64
| | | {stateful-sync-opt}?
+--ro association-list*
| | | [id source global-source extended-id]
| | | +--ro type? assoc-type
| | | +--ro id uint16
| | | +--ro source inet:ip-address
| | | +--ro global-source uint32
| | | +--ro extended-id string
| | | +--ro lsp* [plsp-id pcc-id]
| | | | +--ro plsp-id leafref
| | | | +--ro pcc-id leafref
+--ro lsp* [plsp-id pcc-id]
| | +--ro plsp-id uint32
| | +--ro pcc-id inet:ip-address
| | +--ro lsp-ref
| | | +--ro source?
| | | | -> /te:te/lsp-state/lsp/source
| | | +--ro destination?
| | | | -> /te:te/lsp-state/lsp/destination
| | +--ro tunnel-id?

```

```

|         -> /te:te/lsp-state/lsp/tunnel-id
+--ro lsp-id?
|         -> /te:te/lsp-state/lsp/lsp-id
+--ro extended-tunnel-id?  leafref
+--ro type?
|         -> /te:te/lsp-state/lsp/type
+--ro admin-state?        boolean
+--ro operational-state?   operational-state
+--ro delegated
|   +--ro enabled?        boolean
|   +--ro peer?
|   |   -> /pcep-state/entity/peers/peer/addr
|   +--ro srp-id?         uint32
+--ro initiation {pce-initiated}?
|   +--ro enabled?        boolean
|   +--ro peer?
|   |   -> /pcep-state/entity/peers/peer/addr
+--ro symbolic-path-name?  string
+--ro last-error?          lsp-error
+--ro pst?                 pst
+--ro association-list*
|   [id source global-source extended-id]
|   +--ro id                leafref
|   +--ro source            leafref
|   +--ro global-source     leafref
|   +--ro extended-id      leafref
+--ro path-keys {path-key}?
+--ro path-keys* [path-key]
+--ro path-key            uint16
+--ro cps
|   +--ro explicit-route-objects* [index]
|   |   +--ro explicit-route-usage?  identityref
|   |   +--ro index                  uint32
|   |   +--ro (type)?
|   |   |   +--:(ip-address)
|   |   |   |   +--ro ip-address-hop
|   |   |   |   |   +--ro address?    inet:ip-address
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   +--:(as-number)
|   |   |   |   +--ro as-number-hop
|   |   |   |   |   +--ro as-number?   binary
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   +--:(unnumbered-link)
|   |   |   |   +--ro unnumbered-hop
|   |   |   |   |   +--ro router-id?   inet:ip-address
|   |   |   |   |   +--ro interface-id? uint32
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   +--:(label)

```

```

| | | +--ro label-hop
| | |     +--ro value?
| | |         rt-types:generalized-label
+---:(sid)
| | |   +--ro sid-hop
| | |     +--ro sid?
| | |         rt-types:generalized-label
+--ro pcc-original?
| |   -> /pcep-state/entity/peers/peer/addr
+--ro req-id?      uint32
+--ro retrieved?    boolean
+--ro pcc-retrieved?
| |   -> /pcep-state/entity/peers/peer/addr
+--ro creation-time?  yang:timestamp
+--ro discard-time?   uint32
+--ro reuse-time?     uint32
+--ro of-list {objective-function}?
|   +--ro objective-function* [of]
|       +--ro of      objective-function
+--ro peers
|   +--ro peer* [addr]
|       +--ro addr                      inet:ip-address
|       +--ro role?                    pcep-role
|       +--ro domain
|           +--ro domain* [domain-type domain]
|               +--ro domain-type      domain-type
|               +--ro domain          domain
+--ro capability
|   +--ro gmpls?                        boolean {gmpls}?
|   +--ro bi-dir?                      boolean
|   +--ro diverse?                     boolean
|   +--ro load-balance?                 boolean
|   +--ro synchronize?                 boolean {svec}?
|   +--ro objective-function?          boolean
|       | {objective-function}?
|   +--ro add-path-constraint?         boolean
|   +--ro prioritization?              boolean
|   +--ro multi-request?               boolean
|   +--ro gco?                        boolean {gco}?
|   +--ro p2mp?                       boolean {p2mp}?
+--ro stateful {stateful}?
|   +--ro enabled?                    boolean
|   +--ro active?                    boolean
|   +--ro pce-initiated?             boolean
|       | {pce-initiated}?
|   +--ro include-db-ver?            boolean
|       | {stateful-sync-opt}?
|   +--ro trigger-resync?            boolean

```

```

|   |   |   {stateful-sync-opt}?
|   |   +---ro trigger-initial-sync?   boolean
|   |   |   {stateful-sync-opt}?
|   |   +---ro incremental-sync?       boolean
|   |   |   {stateful-sync-opt}?
+---ro sr {sr}?
|   +---ro enabled?   boolean
+---ro pce-info
|   +---ro scope
|   |   +---ro intra-area-scope?       boolean
|   |   +---ro intra-area-pref?       uint8
|   |   +---ro inter-area-scope?       boolean
|   |   +---ro inter-area-scope-default? boolean
|   |   +---ro inter-area-pref?       uint8
|   |   +---ro inter-as-scope?         boolean
|   |   +---ro inter-as-scope-default? boolean
|   |   +---ro inter-as-pref?         uint8
|   |   +---ro inter-layer-scope?      boolean
|   |   +---ro inter-layer-pref?      uint8
+---ro neigh-domains
|   +---ro domain* [domain-type domain]
|   |   +---ro domain-type   domain-type
|   |   +---ro domain       domain
+---ro delegation-pref?   uint8 {stateful}?
+---ro (auth-type-selection)?
|   +---:(auth-key-chain)
|   |   +---ro key-chain?
|   |   |   key-chain:key-chain-ref
+---:(auth-key)
|   +---ro crypto-algorithm   identityref
|   +---ro key-string
|   |   +---ro (key-string-style)?
|   |   |   +---:(keystring)
|   |   |   |   +---ro keystring?   string
|   |   |   +---:(hexadecimal)
|   |   |   |   {key-chain:hex-key-string}?
|   |   |   +---ro hexadecimal-string?
|   |   |   |   yang:hex-string
+---:(auth-tls) {tls}?
|   +---ro tls
+---ro discontinuity-time?   yang:timestamp
+---ro initiate-session?    boolean
+---ro session-exists?      boolean
+---ro session-up-time?     yang:timestamp
+---ro session-fail-time?   yang:timestamp
+---ro session-fail-up-time? yang:timestamp
+---ro sessions
|   +---ro session* [initiator]

```



```

+--ro initiator                pcep-initiator
+--ro role?                    -> ../../../../role
+--ro state-last-change?      yang:timestamp
+--ro state?                   pcep-sess-state
+--ro session-creation?       yang:timestamp
+--ro connect-retry?          yang:counter32
+--ro local-id?                uint32
+--ro remote-id?              uint32
+--ro keepalive-timer?        uint32
+--ro peer-keepalive-timer?   uint32
+--ro dead-timer?             uint32
+--ro peer-dead-timer?        uint32
+--ro ka-hold-time-rem?       uint32
+--ro overloaded?             boolean
+--ro overload-time?          uint32
+--ro peer-overloaded?        boolean
+--ro peer-overload-time?     uint32
+--ro lspdb-sync?             sync-state
|                               {stateful}?
+--ro recv-db-ver?            uint64
|                               {stateful,stateful-sync-opt}?
+--ro of-list {objective-function}?
|   +--ro objective-function* [of]
|   |   +--ro of objective-function
+--ro speaker-entity-id?      string
|                               {stateful-sync-opt}?
+--ro discontinuity-time?     yang:timestamp

```

rpcs:

```

+---x trigger-resync {stateful,stateful-sync-opt}?
+---w input
+---w pcc?    -> /pcep-state/entity/peers/peer/addr

```

notifications:

```

+---n pcep-session-up
|   +--ro peer-addr?
|   |   -> /pcep-state/entity/peers/peer/addr
|   +--ro session-initiator? leafref
|   +--ro state-last-change? yang:timestamp
|   +--ro state?             pcep-sess-state
+---n pcep-session-down
|   +--ro peer-addr?
|   |   -> /pcep-state/entity/peers/peer/addr
|   +--ro session-initiator? pcep-initiator
|   +--ro state-last-change? yang:timestamp
|   +--ro state?             pcep-sess-state
+---n pcep-session-local-overload
|   +--ro peer-addr?

```

```

| |           -> /pcep-state/entity/peers/peer/addr
| +---ro session-initiator?   leafref
| +---ro overloaded?          boolean
| +---ro overload-time?       uint32
+---n pcep-session-local-overload-clear
| +---ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
| +---ro overloaded?          boolean
+---n pcep-session-peer-overload
| +---ro peer-addr?
| |           -> /pcep-state/entity/peers/peer/addr
| +---ro session-initiator?   leafref
| +---ro peer-overloaded?      boolean
| +---ro peer-overload-time?   uint32
+---n pcep-session-peer-overload-clear
| +---ro peer-addr?
| |           -> /pcep-state/entity/peers/peer/addr
| +---ro peer-overloaded?      boolean

```

5.1. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using `ietf-inet-types` [RFC6991]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [RFC7420] uses an entity list and a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding `pcePcepEntityIndex` and only one entity is assumed.

5.2. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address (addr-type, addr).

There is one list for static peer configuration ("/pcep/entity/peers"), and a separate list for the operational state of all peers (i.e. static as well as discovered)("/pcep-state/entity/peers"). The former is used to enable remote PCE configuration at PCC (or PCE) while the latter has the operational state of these peers as well as the remote PCE peer which were discovered and PCC peers that have initiated session.

5.3. The Session Lists

The session list contains PCEP session that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. If either of these sessions reaches active state first, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

There is only one list for the operational state of all sessions ("/pcep-state/entity/peers/peer/sessions/session").

5.4. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB notifications.

5.5. RPC

This YANG model defines a RPC to trigger state resynchronization to a particular PCEP peer.

6. The Design of PCEP Statistics Data Model

The module, "ietf-pcep-stats", augments the ietf-pcep module to include statistics at the PCEP peer and session level.

```
module: ietf-pcep-stats
  augment /p:pcep-state/p:entity/p:peers/p:peer:
    +--ro num-sess-setup-ok?      yang:counter32
    +--ro num-sess-setup-fail?    yang:counter32
    +--ro pcep-stats
      +--ro avg-rsp-time?          uint32
      +--ro lwm-rsp-time?          uint32
      +--ro hwm-rsp-time?          uint32
      +--ro num-pcreq-sent?        yang:counter32
      +--ro num-pcreq-rcvd?        yang:counter32
      +--ro num-pcrep-sent?        yang:counter32
      +--ro num-pcrep-rcvd?        yang:counter32
      +--ro num-pcerr-sent?        yang:counter32
      +--ro num-pcerr-rcvd?        yang:counter32
      +--ro num-pcntf-sent?        yang:counter32
      +--ro num-pcntf-rcvd?        yang:counter32
      +--ro num-keepalive-sent?    yang:counter32
      +--ro num-keepalive-rcvd?    yang:counter32
      +--ro num-unknown-rcvd?      yang:counter32
      +--ro num-corrupt-rcvd?      yang:counter32
      +--ro num-req-sent?          yang:counter32
      +--ro num-req-sent-pend-rep? yang:counter32
      +--ro num-req-sent-ero-rcvd? yang:counter32
      +--ro num-req-sent-nopath-rcvd? yang:counter32
      +--ro num-req-sent-cancel-rcvd? yang:counter32
      +--ro num-req-sent-error-rcvd? yang:counter32
      +--ro num-req-sent-timeout?   yang:counter32
      +--ro num-req-sent-cancel-sent? yang:counter32
      +--ro num-req-rcvd?          yang:counter32
      +--ro num-req-rcvd-pend-rep?  yang:counter32
      +--ro num-req-rcvd-ero-sent?   yang:counter32
      +--ro num-req-rcvd-nopath-sent? yang:counter32
      +--ro num-req-rcvd-cancel-sent? yang:counter32
      +--ro num-req-rcvd-error-sent? yang:counter32
      +--ro num-req-rcvd-cancel-rcvd? yang:counter32
      +--ro num-rep-rcvd-unknown?    yang:counter32
      +--ro num-req-rcvd-unknown?    yang:counter32
      +--ro svec {p:svec}?
```

```

    |   +---ro num-svec-sent?          yang:counter32
    |   +---ro num-svec-req-sent?     yang:counter32
    |   +---ro num-svec-rcvd?        yang:counter32
    |   +---ro num-svec-req-rcvd?     yang:counter32
+---ro stateful {p:stateful}?
    |   +---ro num-pcrpt-sent?        yang:counter32
    |   +---ro num-pcrpt-rcvd?       yang:counter32
    |   +---ro num-pcupd-sent?       yang:counter32
    |   +---ro num-pcupd-rcvd?       yang:counter32
    |   +---ro num-rpt-sent?         yang:counter32
    |   +---ro num-rpt-rcvd?         yang:counter32
    |   +---ro num-rpt-rcvd-error-sent? yang:counter32
    |   +---ro num-upd-sent?         yang:counter32
    |   +---ro num-upd-rcvd?         yang:counter32
    |   +---ro num-upd-rcvd-unknown? yang:counter32
    |   +---ro num-upd-rcvd-undelagated? yang:counter32
    |   +---ro num-upd-rcvd-error-sent? yang:counter32
    |   +---ro initiation {p:pce-initiated}?
    |       +---ro num-pcinitiate-sent?        yang:counter32
    |       +---ro num-pcinitiate-rcvd?       yang:counter32
    |       +---ro num-initiate-sent?         yang:counter32
    |       +---ro num-initiate-rcvd?         yang:counter32
    |       +---ro num-initiate-rcvd-error-sent? yang:counter32
+---ro path-key {p:path-key}?
    |   +---ro num-unknown-path-key?        yang:counter32
    |   +---ro num-exp-path-key?            yang:counter32
    |   +---ro num-dup-path-key?            yang:counter32
    |   +---ro num-path-key-no-attempt?     yang:counter32
+---ro num-req-sent-closed?        yang:counter32
+---ro num-req-rcvd-closed?       yang:counter32
augment /p:pcep-state/p:entity/p:peers/p:peer/p:sessions/p:session:
+---ro pcep-stats
    +---ro avg-rsp-time?            uint32
    +---ro lwm-rsp-time?            uint32
    +---ro hwm-rsp-time?            uint32
    +---ro num-pcreq-sent?          yang:counter32
    +---ro num-pcreq-rcvd?          yang:counter32
    +---ro num-pcrep-sent?          yang:counter32
    +---ro num-pcrep-rcvd?          yang:counter32
    +---ro num-pcerr-sent?          yang:counter32
    +---ro num-pcerr-rcvd?          yang:counter32
    +---ro num-pcntf-sent?          yang:counter32
    +---ro num-pcntf-rcvd?          yang:counter32
    +---ro num-keepalive-sent?      yang:counter32
    +---ro num-keepalive-rcvd?      yang:counter32
    +---ro num-unknown-rcvd?        yang:counter32
    +---ro num-corrupt-rcvd?        yang:counter32
    +---ro num-req-sent?            yang:counter32

```

```

+--ro num-req-sent-pend-rep?          yang:counter32
+--ro num-req-sent-ero-rcvd?          yang:counter32
+--ro num-req-sent-nopath-rcvd?       yang:counter32
+--ro num-req-sent-cancel-rcvd?       yang:counter32
+--ro num-req-sent-error-rcvd?        yang:counter32
+--ro num-req-sent-timeout?           yang:counter32
+--ro num-req-sent-cancel-sent?       yang:counter32
+--ro num-req-rcvd?                   yang:counter32
+--ro num-req-rcvd-pend-rep?          yang:counter32
+--ro num-req-rcvd-ero-sent?          yang:counter32
+--ro num-req-rcvd-nopath-sent?       yang:counter32
+--ro num-req-rcvd-cancel-sent?       yang:counter32
+--ro num-req-rcvd-error-sent?        yang:counter32
+--ro num-req-rcvd-cancel-rcvd?       yang:counter32
+--ro num-rep-rcvd-unknown?           yang:counter32
+--ro num-req-rcvd-unknown?           yang:counter32
+--ro svec {p:svec}?
|   +--ro num-svec-sent?               yang:counter32
|   +--ro num-svec-req-sent?           yang:counter32
|   +--ro num-svec-rcvd?               yang:counter32
|   +--ro num-svec-req-rcvd?           yang:counter32
+--ro stateful {p:stateful}?
|   +--ro num-pcrpt-sent?               yang:counter32
|   +--ro num-pcrpt-rcvd?               yang:counter32
|   +--ro num-pcupd-sent?               yang:counter32
|   +--ro num-pcupd-rcvd?               yang:counter32
|   +--ro num-rpt-sent?                 yang:counter32
|   +--ro num-rpt-rcvd?                 yang:counter32
|   +--ro num-rpt-rcvd-error-sent?      yang:counter32
|   +--ro num-upd-sent?                 yang:counter32
|   +--ro num-upd-rcvd?                 yang:counter32
|   +--ro num-upd-rcvd-unknown?         yang:counter32
|   +--ro num-upd-rcvd-undelegated?     yang:counter32
|   +--ro num-upd-rcvd-error-sent?      yang:counter32
|   +--ro initiation {p:pce-initiated}?
|       +--ro num-pcinitiate-sent?      yang:counter32
|       +--ro num-pcinitiate-rcvd?      yang:counter32
|       +--ro num-initiate-sent?        yang:counter32
|       +--ro num-initiate-rcvd?        yang:counter32
|       +--ro num-initiate-rcvd-error-sent? yang:counter32
+--ro path-key {p:path-key}?
|   +--ro num-unknown-path-key?         yang:counter32
|   +--ro num-exp-path-key?             yang:counter32
|   +--ro num-dup-path-key?             yang:counter32
|   +--ro num-path-key-no-attempt?      yang:counter32

```

7. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

This document further handles advanced PCE features like -

- o Capability and Scope
- o Domain information (local/neighbour)
- o Path-Key
- o OF
- o GCO
- o P2MP
- o GMPLS
- o Inter-Layer
- o Stateful PCE
- o Segment Routing
- o Authentication including PCEPS (TLS)

[Editor's Note - TLS would be added in a future revision]

7.1. Stateful PCE's LSP-DB

In the operational state of PCEP which supports stateful PCE mode, the list of LSP state are maintained in LSP-DB. The key is the PLSP-ID and the PCC IP address.

The PCEP data model contains the operational state of LSPs (/pcep-state/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [I-D.ietf-teas-yang-te]. A reference to LSP state in TE model is maintained.

8. Open Issues and Next Step

This section is added so that open issues can be tracked. This section would be removed when the document is ready for publication.

8.1. The PCE-Initiated LSP

The TE Model at [I-D.ietf-teas-yang-te] should support creationg of tunnels at the controller (PCE) and marking them as PCE-Initiated. The LSP-DB in the PCEP Yang (/pcep-state/entity/lsp-db/lsp/initiation) also marks the LSPs which are PCE-initiated.

8.2. PCEP over TLS (PCEPS)

A future version of this document would add TLS related configurations.

9. PCEP YANG Modules

9.1. ietf-pcep module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2017-03-12.yang"
module ietf-pcep {

    yang-version 1.1;

    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
    prefix pcep;

    import ietf-inet-types {
        prefix "inet";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import ietf-te {
        prefix "te";
    }

    import ietf-te-types {
        prefix "te-types";
    }

    import ietf-key-chain {
```



```
    prefix "key-chain";
}

import ietf-netconf-acm {
    prefix nacm;
}

organization
    "IETF PCE (Path Computation Element) Working Group";

contact
    "WG Web:      <http://tools.ietf.org/wg/pce/>
    WG List:      <mailto:pce@ietf.org>
    WG Chair:     JP Vasseur
                  <mailto:jpv@cisco.com>
    WG Chair:     Julien Meuric
                  <mailto:julien.meuric@orange.com>
    WG Chair:     Jonathan Hardwick
                  <mailto:Jonathan.Hardwick@metaswitch.com>
    Editor:       Dhruv Dhody
                  <mailto:dhruv.ietf@gmail.com>";

description
    "The YANG module defines a generic configuration and
    operational model for PCEP common across all of the
    vendor implementations.";

revision 2017-03-12 {
    description "Initial revision.";
    reference
        "RFC XXXX: A YANG Data Model for Path Computation
        Element Communications Protocol
        (PCEP)";
}

/*
 * Identities
 */

identity pcep {
    description "Identity for the PCEP protocol.";
}

/*
 * Typedefs
 */
```

```
typedef pcep-role {
  type enumeration {
    enum unknown {
      value "0";
      description
        "An unknown role";
    }
    enum pcc {
      value "1";
      description
        "The role of a Path Computation Client";
    }
    enum pce {
      value "2";
      description
        "The role of Path Computation Element";
    }
    enum pcc-and-pce {
      value "3";
      description
        "The role of both Path Computation Client and
        Path Computation Element";
    }
  }
  description
    "The role of a PCEP speaker.
    Takes one of the following values
    - unknown(0): the role is not known.
    - pcc(1): the role is of a Path Computation
      Client (PCC).
    - pce(2): the role is of a Path Computation
      Server (PCE).
    - pccAndPce(3): the role is of both a PCC and
      a PCE.";
}

typedef pcep-admin-status {
  type enumeration {
    enum admin-status-up {
      value "1";
      description
        "Admin Status is Up";
    }
    enum admin-status-down {
      value "2";
      description
```

```
        "Admin Status is Down";
    }
}

description
"The Admin Status of the PCEP entity.
Takes one of the following values
- admin-status-up(1): Admin Status is Up.
- admin-status-down(2): Admin Status is Down";
}

typedef pcep-oper-status {
    type enumeration {
        enum oper-status-up {
            value "1";
            description
            "The PCEP entity is active";
        }
        enum oper-status-down {
            value "2";
            description
            "The PCEP entity is inactive";
        }
        enum oper-status-going-up {
            value "3";
            description
            "The PCEP entity is activating";
        }
        enum oper-status-going-down {
            value "4";
            description
            "The PCEP entity is deactivating";
        }
        enum oper-status-failed {
            value "5";
            description
            "The PCEP entity has failed and will recover
            when possible.";
        }
        enum oper-status-failed-perm {
            value "6";
            description
            "The PCEP entity has failed and will not recover
            without operator intervention";
        }
    }
}
description
"The operational status of the PCEP entity.
```

```
    Takes one of the following values
    - oper-status-up(1): Active
    - oper-status-down(2): Inactive
    - oper-status-going-up(3): Activating
    - oper-status-going-down(4): Deactivating
    - oper-status-failed(5): Failed
    - oper-status-failed-perm(6): Failed Permanantly";
}

typedef pcep-initiator {
  type enumeration {
    enum local {
      value "1";
      description
        "The local PCEP entity initiated the session";
    }

    enum remote {
      value "2";
      description
        "The remote PCEP peer initiated the session";
    }
  }
  description
    "The initiator of the session, that is, whether the TCP
    connection was initiated by the local PCEP entity or
    the remote peer.
    Takes one of the following values
    - local(1): Initiated locally
    - remote(2): Initiated remotely";
}

typedef pcep-sess-state {
  type enumeration {
    enum tcp-pending {
      value "1";
      description
        "The tcp-pending state of PCEP session.";
    }

    enum open-wait {
      value "2";
      description
        "The open-wait state of PCEP session.";
    }

    enum keep-wait {
      value "3";
    }
  }
}
```

```
        description
            "The keep-wait state of PCEP session.";
    }

    enum session-up {
        value "4";
        description
            "The session-up state of PCEP session.";
    }
}
description
    "The current state of the session.
    The set of possible states excludes the idle state
    since entries do not exist in the idle state.
    Takes one of the following values
    - tcp-pending(1): PCEP TCP Pending state
    - open-wait(2): PCEP Open Wait state
    - keep-wait(3): PCEP Keep Wait state
    - session-up(4): PCEP Session Up state";
}

typedef domain-type {
    type enumeration {
        enum ospf-area {
            value "1";
            description
                "The OSPF area.";
        }
        enum isis-area {
            value "2";
            description
                "The IS-IS area.";
        }
        enum as {
            value "3";
            description
                "The Autonomous System (AS).";
        }
    }
    description
        "The PCE Domain Type";
}

typedef domain-ospf-area {
    type union {
        type uint32;
        type yang:dotted-quad;
    }
}
```

```
        description
            "OSPF Area ID.";
    }

    typedef domain-isis-area {
        type string {
            pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
        }
        description
            "IS-IS Area ID.";
    }

    typedef domain-as {
        type uint32;
        description
            "Autonomous System number.";
    }

    typedef domain {
        type union {
            type domain-ospf-area;
            type domain-isis-area;
            type domain-as;
        }
        description
            "The Domain Information";
    }

    typedef operational-state {
        type enumeration {
            enum down {
                value "0";
                description
                    "not active.";
            }
            enum up {
                value "1";
                description
                    "signalled.";
            }
            enum active {
                value "2";
                description
                    "up and carrying traffic.";
            }
            enum going-down {
                value "3";
            }
        }
    }
```

```
        description
            "LSP is being torn down, resources are
            being released.";
    }
    enum going-up {
        value "4";
        description
            "LSP is being signalled.";
    }
}
description
    "The operational status of the LSP";
}

typedef lsp-error {
    type enumeration {
        enum no-error {
            value "0";
            description
                "No error, LSP is fine.";
        }
        enum unknown {
            value "1";
            description
                "Unknown reason.";
        }
        enum limit {
            value "2";
            description
                "Limit reached for PCE-controlled LSPs.";
        }
        enum pending {
            value "3";
            description
                "Too many pending LSP update requests.";
        }
        enum unacceptable {
            value "4";
            description
                "Unacceptable parameters.";
        }
        enum internal {
            value "5";
            description
                "Internal error.";
        }
        enum admin {
            value "6";
        }
    }
}
```

```
        description
            "LSP administratively brought down.";
    }
    enum preempted {
        value "7";
        description
            "LSP preempted.";
    }
    enum rsvp {
        value "8";
        description
            "RSVP signaling error.";
    }
}
description
    "The LSP Error Codes.";
}

typedef sync-state {
    type enumeration {
        enum pending {
            value "0";
            description
                "The state synchronization
                 has not started.";
        }
        enum ongoing {
            value "1";
            description
                "The state synchronization
                 is ongoing.";
        }
        enum finished {
            value "2";
            description
                "The state synchronization
                 is finished.";
        }
    }
}
description
    "The LSP-DB state synchronization operational
    status.";
}

typedef pst{
    type enumeration{
        enum rsvp-te{
            value "0";
```



```
        description
            "RSVP-TE signaling protocol";
    }
    enum sr{
        value "1";
        description
            "Segment Routing Traffic Engineering";
    }
}
description
    "The Path Setup Type";
}

typedef assoc-type{
    type enumeration{
        enum protection{
            value "1";
            description
                "Path Protection Association Type";
        }
        enum policy{
            value "2";
            description
                "Policy Association Type";
        }
        enum diversity{
            value "3";
            description
                "Diversity Association Type";
        }
    }
    description
        "The PCEP Association Type";
}

typedef objective-function{
    type enumeration{
        enum mcp{
            value "1";
            description
                "Minimum Cost Path (MCP)";
        }
        enum mlp{
            value "2";
            description
                "Minimum Load Path (MLP)";
        }
        enum mbp{
```

```
        value "3";
        description
            "Maximum residual Bandwidth Path (MBP)";
    }
    enum mbc{
        value "4";
        description
            "Minimize aggregate Bandwidth Consumption
            (MBC)";
    }
    enum mll{
        value "5";
        description
            "Minimize the Load of the most loaded Link
            (MLL)";
    }
    enum mcc{
        value "6";
        description
            "Minimize the Cumulative Cost of a set of
            paths (MCC)";
    }
    enum spt{
        value "7";
        description
            "Shortest Path Tree (SPT)";
    }
    enum mct{
        value "8";
        description
            "Minimum Cost Tree (MCT)";
    }
    enum mplp{
        value "9";
        description
            "Minimum Packet Loss Path (MPLP)";
    }
    enum mup{
        value "10";
        description
            "Maximum Under-Utilized Path (MUP)";
    }
    enum mrup{
        value "11";
        description
            "Maximum Reserved Under-Utilized Path
            (MRUP)";
    }
}
```

```
    }
    description
        "The PCEP Objective functions";
}

/*
 * Features
 */

feature svec {
    description
        "Support synchronized path computation.";
}

feature gmpls {
    description
        "Support GMPLS.";
}

feature objective-function {
    description
        "Support OF as per RFC 5541.";
}

feature gco {
    description
        "Support GCO as per RFC 5557.";
}

feature path-key {
    description
        "Support path-key as per RFC 5520.";
}

feature p2mp {
    description
        "Support P2MP as per RFC 6006.";
}

feature stateful {
    description
        "Support stateful PCE.";
}

feature stateful-sync-opt {
    description
        "Support stateful sync optimization";
}
```

```
feature pce-initiated {
  description
    "Support PCE-Initiated LSP.";
}

feature tls {
  description
    "Support PCEP over TLS.";
}

feature sr {
  description
    "Support Segment Routing for PCE.";
}

/*
 * Groupings
 */

grouping pcep-entity-info{
  description
    "This grouping defines the attributes for PCEP entity.";
  leaf connect-timer {
    type uint32 {
      range "1..65535";
    }
    units "seconds";
    default 60;
    description
      "The time in seconds that the PCEP entity will wait
      to establish a TCP connection with a peer. If a
      TCP connection is not established within this time
      then PCEP aborts the session setup attempt.";
    reference
      "RFC 5440: Path Computation Element (PCE)
      Communication Protocol (PCEP)";
  }

  leaf connect-max-retry {
    type uint32;
    default 5;
    description
      "The maximum number of times the system tries to
      establish a TCP connection to a peer before the
      session with the peer transitions to the idle
      state.";
  }
}
```

```
        reference
          "RFC 5440: Path Computation Element (PCE)
            Communication Protocol (PCEP)";
    }

    leaf init-backoff-timer {
      type uint32 {
        range "1..65535";
      }
      units "seconds";
      description
        "The initial back-off time in seconds for retrying
        a failed session setup attempt to a peer.
        The back-off time increases for each failed
        session setup attempt, until a maximum back-off
        time is reached. The maximum back-off time is
        max-backoff-timer.";
    }

    leaf max-backoff-timer {
      type uint32;
      units "seconds";
      description
        "The maximum back-off time in seconds for retrying
        a failed session setup attempt to a peer.
        The back-off time increases for each failed session
        setup attempt, until this maximum value is reached.
        Session setup attempts then repeat periodically
        without any further increase in back-off time.";
    }

    leaf open-wait-timer {
      type uint32 {
        range "1..65535";
      }
      units "seconds";
      default 60;
      description
        "The time in seconds that the PCEP entity will wait
        to receive an Open message from a peer after the
        TCP connection has come up.
        If no Open message is received within this time then
        PCEP terminates the TCP connection and deletes the
        associated sessions.";
      reference
        "RFC 5440: Path Computation Element (PCE)
          Communication Protocol (PCEP)";
    }
  }
```

```
leaf keep-wait-timer {
  type uint32 {
    range "1..65535";
  }
  units "seconds";
  default 60;
  description
    "The time in seconds that the PCEP entity will wait
    to receive a Keepalive or PCErr message from a peer
    during session initialization after receiving an
    Open message. If no Keepalive or PCErr message is
    received within this time then PCEP terminates the
    TCP connection and deletes the associated
    sessions.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}

leaf keep-alive-timer {
  type uint32 {
    range "0..255";
  }
  units "seconds";
  default 30;
  description
    "The keep alive transmission timer that this PCEP
    entity will propose in the initial OPEN message of
    each session it is involved in. This is the
    maximum time between two consecutive messages sent
    to a peer. Zero means that the PCEP entity prefers
    not to send Keepalives at all.
    Note that the actual Keepalive transmission
    intervals, in either direction of an active PCEP
    session, are determined by negotiation between the
    peers as specified by RFC 5440, and so may differ
    from this configured value.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}

leaf dead-timer {
  type uint32 {
    range "0..255";
  }
  units "seconds";
  must "(. > ../keep-alive-timer)" {
```

```
        error-message "The dead timer must be "
            + "larger than the keep alive timer";
        description
            "This value MUST be greater than
            keep-alive-timer.";
    }
    default 120;
    description
        "The dead timer that this PCEP entity will propose
        in the initial OPEN message of each session it is
        involved in. This is the time after which a peer
        should declare a session down if it does not
        receive any PCEP messages. Zero suggests that the
        peer does not run a dead timer at all." ;
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}

leaf allow-negotiation{
    type boolean;
    description
        "Whether the PCEP entity will permit negotiation of
        session parameters.";
}

leaf max-keep-alive-timer{
    type uint32 {
        range "0..255";
    }
    units "seconds";
    description
        "In PCEP session parameter negotiation in seconds,
        the maximum value that this PCEP entity will
        accept from a peer for the interval between
        Keepalive transmissions. Zero means that the PCEP
        entity will allow no Keepalive transmission at
        all." ;
}

leaf max-dead-timer{
    type uint32 {
        range "0..255";
    }
    units "seconds";
    description
        "In PCEP session parameter negotiation in seconds,
```

```
        the maximum value that this PCEP entity will accept
        from a peer for the Dead timer.  Zero means that
        the PCEP entity will allow not running a Dead
        timer.";
    }

    leaf min-keep-alive-timer{
        type uint32 {
            range "0..255";
        }
        units "seconds";
        description
            "In PCEP session parameter negotiation in seconds,
            the minimum value that this PCEP entity will
            accept for the interval between Keepalive
            transmissions. Zero means that the PCEP entity
            insists on no Keepalive transmission at all.";
    }

    leaf min-dead-timer{
        type uint32 {
            range "0..255";
        }
        units "seconds";
        description
            "In PCEP session parameter negotiation in
            seconds, the minimum value that this PCEP entity
            will accept for the Dead timer.  Zero means that
            the PCEP entity insists on not running a Dead
            timer.";
    }

    leaf sync-timer{
        if-feature svec;
        type uint32 {
            range "0..65535";
        }
        units "seconds";
        default 60;
        description
            "The value of SyncTimer in seconds is used in the
            case of synchronized path computation request
            using the SVEC object. Consider the case where a
            PCReq message is received by a PCE that contains
            the SVEC object referring to M synchronized path
            computation requests. If after the expiration of
            the SyncTimer all the M path computation requests
            have not been, received a protocol error is
```



```
        triggered and the PCE MUST cancel the whole set
        of path computation requests.
        The aim of the SyncTimer is to avoid the storage
        of unused synchronized requests should one of
        them get lost for some reasons (for example, a
        misbehaving PCC).
        Zero means that the PCEP entity does not use the
        SyncTimer.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}

leaf request-timer{
    type uint32 {
        range "1..65535";
    }
    units "seconds";
    description
        "The maximum time that the PCEP entity will wait
        for a response to a PCReq message.";
}

leaf max-sessions{
    type uint32;
    description
        "Maximum number of sessions involving this PCEP
        entity that can exist at any time.";
}

leaf max-unknown-reqs{
    type uint32;
    default 5;
    description
        "The maximum number of unrecognized requests and
        replies that any session on this PCEP entity is
        willing to accept per minute before terminating
        the session.
        A PCRep message contains an unrecognized reply
        if it contains an RP object whose request ID
        does not correspond to any in-progress request
        sent by this PCEP entity.
        A PCReq message contains an unrecognized request
        if it contains an RP object whose request ID is
        zero.";
    reference
        "RFC 5440: Path Computation Element (PCE)";
}
```

```

        Communication Protocol (PCEP)";
    }

    leaf max-unknown-msgs{
        type uint32;
        default 5;
        description
            "The maximum number of unknown messages that any
             session on this PCEP entity is willing to accept
             per minute before terminating the session.";
        reference
            "RFC 5440: Path Computation Element (PCE)
             Communication Protocol (PCEP)";
    }
}

} // pcep-entity-info

grouping pce-scope{
    description
        "This grouping defines PCE path computation scope
         information which maybe relevant to PCE selection.
         This information corresponds to PCE auto-discovery
         information.";
    reference
        "RFC 5088: OSPF Protocol Extensions for Path
         Computation Element (PCE)
         Discovery
         RFC 5089: IS-IS Protocol Extensions for Path
         Computation Element (PCE)
         Discovery";
    leaf intra-area-scope{
        type boolean;
        default true;
        description
            "PCE can compute intra-area paths.";
    }
    leaf intra-area-pref{
        type uint8{
            range "0..7";
        }
        description
            "The PCE's preference for intra-area TE LSP
             computation.";
    }
    leaf inter-area-scope{
        type boolean;
        default false;
        description

```

```
        "PCE can compute inter-area paths.";
    }
    leaf inter-area-scope-default{
        type boolean;
        default false;
        description
            "PCE can act as a default PCE for inter-area
            path computation.";
    }
    leaf inter-area-pref{
        type uint8{
            range "0..7";
        }
        description
            "The PCE's preference for inter-area TE LSP
            computation.";
    }
    leaf inter-as-scope{
        type boolean;
        default false;
        description
            "PCE can compute inter-AS paths.";
    }
    leaf inter-as-scope-default{
        type boolean;
        default false;
        description
            "PCE can act as a default PCE for inter-AS
            path computation.";
    }
    leaf inter-as-pref{
        type uint8{
            range "0..7";
        }
        description
            "The PCE's preference for inter-AS TE LSP
            computation.";
    }
    leaf inter-layer-scope{
        type boolean;
        default false;
        description
            "PCE can compute inter-layer paths.";
    }
    leaf inter-layer-pref{
        type uint8{
            range "0..7";
        }
    }
```

```
        description
          "The PCE's preference for inter-layer TE LSP
          computation.";
      }
  }//pce-scope

  grouping domain{
    description
      "This grouping specifies a Domain where the
      PCEP speaker has topology visibility.";
    leaf domain-type{
      type domain-type;
      description
        "The domain type.";
    }
    leaf domain{
      type domain;
      description
        "The domain Information.";
    }
  }//domain

  grouping capability{
    description
      "This grouping specifies a capability
      information of local PCEP entity. This maybe
      relevant to PCE selection as well. This
      information corresponds to PCE auto-discovery
      information.";
    reference
      "RFC 5088: OSPF Protocol Extensions for Path
      Computation Element (PCE)
      Discovery
      RFC 5089: IS-IS Protocol Extensions for Path
      Computation Element (PCE)
      Discovery";
    leaf gmpls{
      if-feature gmpls;
      type boolean;
      description
        "Path computation with GMPLS link
        constraints.";
    }
    leaf bi-dir{
      type boolean;
      description
        "Bidirectional path computation.";
    }
  }
```

```
leaf diverse{
  type boolean;
  description
    "Diverse path computation.";
}
leaf load-balance{
  type boolean;
  description
    "Load-balanced path computation.";
}
leaf synchronize{
  if-feature svec;
  type boolean;
  description
    "Synchronized paths computation.";
}
leaf objective-function{
  if-feature objective-function;
  type boolean;
  description
    "Support for multiple objective functions.";
}
leaf add-path-constraint{
  type boolean;
  description
    "Support for additive path constraints (max
    hop count, etc.).";
}
leaf prioritization{
  type boolean;
  description
    "Support for request prioritization.";
}
leaf multi-request{
  type boolean;
  description
    "Support for multiple requests per message.";
}
leaf gco{
  if-feature gco;
  type boolean;
  description
    "Support for Global Concurrent Optimization
    (GCO).";
}
leaf p2mp{
  if-feature p2mp;
  type boolean;
```

```
        description
        "Support for P2MP path computation.";
    }

    container stateful{
        if-feature stateful;
        description
            "If stateful PCE feature is present";
        leaf enabled{
            type boolean;
            description
                "Enabled or Disabled";
        }
        leaf active{
            type boolean;
            description
                "Support for active stateful PCE.";
        }
        leaf pce-initiated{
            if-feature pce-initiated;
            type boolean;
            description
                "Support for PCE-initiated LSP.";
        }
        leaf include-db-ver{
            if-feature stateful-sync-opt;
            type boolean;
            description
                "Support inclusion of LSP-DB-VERSION
                in LSP object";
        }
        leaf trigger-resync{
            if-feature stateful-sync-opt;
            type boolean;
            description
                "Support PCE triggered re-synchronization";
        }
        leaf trigger-initial-sync{
            if-feature stateful-sync-opt;
            type boolean;
            description
                "PCE triggered initial synchronization";
        }
        leaf incremental-sync{
            if-feature stateful-sync-opt;
            type boolean;
            description
                "Support incremental (delta) sync";
        }
    }
}
```

```
    }
  }
  container sr{
    if-feature sr;
    description
      "If segment routing is supported";
    leaf enabled{
      type boolean;
      description
        "Enabled or Disabled";
    }
  }
}
} //capability

grouping info{
  description
    "This grouping specifies all information which
    maybe relevant to both PCC and PCE.
    This information corresponds to PCE auto-discovery
    information.";
  container domain{
    description
      "The local domain for the PCEP entity";
    list domain{
      key "domain-type domain";
      description
        "The local domain.";
      uses domain{
        description
          "The local domain for the PCEP entity.";
      }
    }
  }
  container capability{
    description
      "The PCEP entity capability";
    uses capability{
      description
        "The PCEP entity supported
        capabilities.";
    }
  }
}

} //info

grouping pce-info{
```

```
description
  "This grouping specifies all PCE information
  which maybe relevant to the PCE selection.
  This information corresponds to PCE auto-discovery
  information.";
container scope{
  description
    "The path computation scope";
  uses pce-scope;
}

container neigh-domains{
  description
    "The list of neighbour PCE-Domain
    toward which a PCE can compute
    paths";
  list domain{
    key "domain-type domain";

    description
      "The neighbour domain.";
    uses domain{
      description
        "The PCE neighbour domain.";
    }
  }
}
} //pce-info

grouping lsp-state{
  description
    "This grouping defines the attributes for LSP in LSP-DB.
    These are the attributes specifically from the PCEP
    perspective";
  leaf plsp-id{
    type uint32{
      range "1..1048575";
    }
    description
      "A PCEP-specific identifier for the LSP. A PCC
      creates a unique PLSP-ID for each LSP that is
      constant for the lifetime of a PCEP session.
      PLSP-ID is 20 bits with 0 and 0xFFFFF are
      reserved";
  }
  leaf pcc-id{
    type inet:ip-address;
    description
```



```
        "The local internet address of the PCC, that
        generated the PLSP-ID.";
    }

    container lsp-ref{
        description
            "reference to ietf-te lsp state";

        leaf source {
            type leafref {
                path "/te:te/te:lsps-state/te:lsp/te:source";
            }
            description
                "Tunnel sender address extracted from
                SENDER_TEMPLATE object";
            reference "RFC3209";
        }
        leaf destination {
            type leafref {
                path "/te:te/te:lsps-state/te:lsp/te:"
                    + "destination";
            }
            description
                "Tunnel endpoint address extracted from
                SESSION object";
            reference "RFC3209";
        }
        leaf tunnel-id {
            type leafref {
                path "/te:te/te:lsps-state/te:lsp/te:tunnel-id";
            }
            description
                "Tunnel identifier used in the SESSION
                that remains constant over the life
                of the tunnel.";
            reference "RFC3209";
        }
        leaf lsp-id {
            type leafref {
                path "/te:te/te:lsps-state/te:lsp/te:lsp-id";
            }
            description
                "Identifier used in the SENDER_TEMPLATE
                and the FILTER_SPEC that can be changed
                to allow a sender to share resources with
                itself.";
            reference "RFC3209";
        }
    }
```

```
    leaf extended-tunnel-id {
      type leafref {
        path "/te:te/te:lsps-state/te:lsp/te:"
          + "extended-tunnel-id";
      }
      description
        "Extended Tunnel ID of the LSP.";
      reference "RFC3209";
    }
    leaf type {
      type leafref {
        path "/te:te/te:lsps-state/te:lsp/te:type";
      }
      description "LSP type P2P or P2MP";
    }
  }

  leaf admin-state{
    type boolean;
    description
      "The desired operational state";
  }
  leaf operational-state{
    type operational-state;
    description
      "The operational status of the LSP";
  }
  container delegated{
    description
      "The delegation related parameters";
    leaf enabled{
      type boolean;
      description
        "LSP is delegated or not";
    }
    leaf peer{
      type leafref {
        path "/pcep-state/entity/peers/peer/addr";
      }
      must "(../enabled = true())"
      {
        error-message
          "The LSP must be delegated";
        description
          "When LSP is a delegated LSP";
      }
      description
        "At the PCC, the reference to the PCEP peer to
```

```
        which LSP is delegated; At the PCE, the
        reference to the PCEP peer which delegated this
        LSP";
    }
    leaf srp-id{
        type uint32;
        description
            "The last SRP-ID-number associated with this
            LSP.";
    }
}
container initiation {
    if-feature pce-initiated;
    description
        "The PCE initiation related parameters";
    leaf enabled{
        type boolean;
        description
            "LSP is PCE-initiated or not";
    }
    leaf peer{
        type leafref {
            path "/pcep-state/entity/peers/peer/addr";
        }
        must "(../enabled = true())"
        {
            error-message
                "The LSP must be PCE-Initiated";
            description
                "When the LSP must be PCE-Initiated";
        }
        description
            "At the PCC, the reference to the PCEP peer
            that initiated this LSP; At the PCE, the
            reference to the PCEP peer where the LSP
            is initiated";
    }
}
leaf symbolic-path-name{
    type string;
    description
        "The symbolic path name associated with the LSP.";
}
leaf last-error{
    type lsp-error;
    description
        "The last error for the LSP.";
}
```

```
        leaf pst{
            type pst;
            default "rsvp-te";
            description
                "The Path Setup Type";
        }

    }//lsp-state

    grouping notification-instance-hdr {
        description
            "This group describes common instance specific data
            for notifications.";

        leaf peer-addr {
            type leafref {
                path "/pcep-state/entity/peers/peer/addr";
            }
            description
                "Reference to peer address";
        }
    }

}// notification-instance-hdr

grouping notification-session-hdr {
    description
        "This group describes common session instance specific
        data for notifications.";

    leaf session-initiator {
        type leafref {
            path "/pcep-state/entity/peers/peer/sessions/" +
                "session/initiator";
        }
        description
            "Reference to pcep session initiator leaf";
    }
}

}// notification-session-hdr

grouping stateful-pce-parameter {
    description
        "This group describes stateful PCE specific
        parameters.";
    leaf state-timeout{
        type uint32;
        units "seconds";
        description
```

```
        "When a PCEP session is terminated, a PCC
        waits for this time period before flushing
        LSP state associated with that PCEP session
        and reverting to operator-defined default
        parameters or behaviours.";
    }
    leaf redelegation-timeout{
        when "../../../role = 'pcc'" +
            "or " +
            "../../../role = 'pcc-and-pce'"
        {
            description
                "Valid at PCC";
        }
        type uint32;
        units "seconds";
        description
            "When a PCEP session is terminated, a PCC
            waits for this time period before revoking
            LSP delegation to a PCE and attempting to
            redelegate LSPs associated with the
            terminated PCEP session to an alternate
            PCE.";
    }
    leaf rpt-non-pcep-lsp{
        when "../../../role = 'pcc'" +
            "or " +
            "../../../role = 'pcc-and-pce'"
        {
            description
                "Valid at PCC";
        }
        type boolean;
        default true;
        description
            "If set, a PCC reports LSPs that are not
            controlled by any PCE (for example, LSPs
            that are statically configured at the
            PCC). ";
    }
}

grouping authentication {
    description "Authentication Information";
    choice auth-type-selection {
        description
            "Options for expressing authentication setting.";
    }
}
```

```
case auth-key-chain {
  leaf key-chain {
    type key-chain:key-chain-ref;
    description
      "key-chain name.";
  }
}
case auth-key {
  leaf crypto-algorithm {
    type identityref {
      base key-chain:crypto-algorithm;
    }
    mandatory true;
    description
      "Cryptographic algorithm associated with key.";
  }
  container key-string {
    description
      "The key string.";
    nacm:default-deny-all;
    choice key-string-style {
      description
        "Key string styles";
      case keystack {
        leaf keystack {
          type string;
          description
            "Key string in ASCII format.";
        }
      }
      case hexadecimal {
        if-feature "key-chain:hex-key-string";
        leaf hexadecimal-string {
          type yang:hex-string;
          description
            "Key in hexadecimal string format. When
            compared to ASCII, specification in
            hexadecimal affords greater key
            entropy with the same number of
            octets. Additionally, it discourages
            usage of well-known words or
            numbers.";
        }
      }
    }
  }
}
case auth-tls {
```

```
        if-feature tls;
        container tls {
            description
                "TLS related information - TBD";
        }
    }
}

grouping path-key {
    description "Path-key related information";
    leaf enabled {
        type boolean;
        description
            "Enabled or Disabled";
    }
    leaf discard-timer {
        type uint32;
        units "minutes";
        default 10;
        description
            "A timer to discard unwanted path-keys";
    }
    leaf reuse-time {
        type uint32;
        units "minutes";
        default 30;
        description
            "A time after which the path-keys could be reused";
    }
    leaf pce-id {
        type inet:ip-address;
        description
            "PCE Address to be used in each Path-Key Subobject
            (PKS)";
    }
}

grouping path-key-state {
    description "Table to allow inspection of path-keys";
    list path-keys {
        key "path-key";

        description
            "The list of path-keys generated by the PCE";

        leaf path-key {
            type uint16;
        }
    }
}
```

```
    description
      "The identifier, or token used to represent
       the Confidential Path Segment (CPS) within
       the context of the PCE";
  }
  container cps {
    description
      "The Confidential Path Segment (CPS)";
    list explicit-route-objects {
      key "index";
      description
        "List of explicit route objects";

      leaf explicit-route-usage {
        type identityref {
          base te-types:route-usage-type;
        }
        description
          "An explicit-route hop action.";
      }

      uses te-types:explicit-route-hop_config;
    }
  }
  leaf pcc-original {
    type leafref {
      path "/pcep-state/entity/peers/peer/addr";
    }
    description
      "Reference to PCC peer address of
       the original request";
  }
  leaf req-id {
    type uint32;
    description
      "The request ID of the original PCReq.";
  }
  leaf retrieved {
    type boolean;
    description
      "If path-key has been retrieved yet";
  }
  leaf pcc-retrieved {
    type leafref {
      path "/pcep-state/entity/peers/peer/addr";
    }
    must "(../retrieved = true())"
    {
```



```

        error-message
            "The Path-key should be retrieved";
        description
            "When Path-Key has been retrieved";
    }
    description
        "Reference to PCC peer address which
        retrieved the path-key";
}
leaf creation-time {
    type yang:timestamp;
    description
        "The timestamp value at the time this Path-Key
        was created.";
}
leaf discard-time {
    type uint32;
    units "minutes";
    description
        "A time after which this path-keys will be
        discarded";
}
leaf reuse-time {
    type uint32;
    units "minutes";
    description
        "A time after which this path-keys could be
        reused";
}
}
}

grouping of-list {
    description "List of OF";
    list objective-function {
        key "of";

        description
            "The list of authorized OF";

        leaf of {
            type objective-function;
            description
                "The OF authorized";
        }
    }
}
grouping association {

```

```
description
  "Generic Association parameters";
leaf type {
  type "assoc-type";
  description
    "The PCEP association type";
}
leaf id {
  type uint16;
  description
    "PCEP Association ID";
}
leaf source {
  type inet:ip-address;
  description
    "PCEP Association Source.";
}
leaf global-source {
  type uint32;
  description
    "PCEP Association Global
    Source.";
}
leaf extended-id{
  type string;
  description
    "Additional information to
    support unique identification.";
}
}
grouping association-ref {
  description
    "Generic Association parameters";
  leaf id {
    type leafref {
      path "/pcep-state/entity/lsp-db/"
        + "association-list/id";
    }
    description
      "PCEP Association ID";
  }
  leaf source {
    type leafref {
      path "/pcep-state/entity/lsp-db/"
        + "association-list/source";
    }
    description
      "PCEP Association Source.";
  }
}
```

```
    }
    leaf global-source {
      type leafref {
        path "/pcep-state/entity/lsp-db/"
          + "association-list/global-source";
      }
      description
        "PCEP Association Global
        Source.";
    }
    leaf extended-id{
      type leafref {
        path "/pcep-state/entity/lsp-db/"
          + "association-list/extended-id";
      }
      description
        "Additional information to
        support unique identification.";
    }
  }
  /*
  * Configuration data nodes
  */
  container pcep{
    presence
      "The PCEP is enabled";

    description
      "Parameters for list of configured PCEP entities
      on the device.";

    container entity {

      description
        "The configured PCEP entity on the device.";

      leaf addr {
        type inet:ip-address;
        mandatory true;
        description
          "The local Internet address of this PCEP
          entity.
          If operating as a PCE server, the PCEP
          entity listens on this address.
          If operating as a PCC, the PCEP entity
          binds outgoing TCP connections to this
          address.
```

```
        It is possible for the PCEP entity to
        operate both as a PCC and a PCE Server, in
        which case it uses this address both to
        listen for incoming TCP connections and to
        bind outgoing TCP connections.";
    }

    leaf enabled {
        type boolean;
        default true;
        description
            "The administrative status of this PCEP
            Entity.";
    }

    leaf role {
        type pcep-role;
        mandatory true;
        description
            "The role that this entity can play.
            Takes one of the following values.
            - unknown(0): this PCEP Entity role is not
              known.
            - pcc(1): this PCEP Entity is a PCC.
            - pce(2): this PCEP Entity is a PCE.
            - pcc-and-pce(3): this PCEP Entity is both
              a PCC and a PCE.";
    }

    leaf description {
        type string;
        description
            "Description of the PCEP entity configured
            by the user";
    }

    leaf speaker-entity-id {
        if-feature stateful-sync-opt;
        type string;
        description
            "The Speaker Entity Identifier";
    }

    uses info {
        description
            "Local PCEP entity information";
    }
}
```

```
container pce-info {
  when "../role = 'pce'" +
    "or " +
    "../role = 'pcc-and-pce'"
  {
    description
      "Valid at PCE";
  }
  uses pce-info {
    description
      "Local PCE information";
  }
  container path-key {
    if-feature path-key;
    uses path-key {
      description
        "Path-Key Configuration";
    }
    description
      "Path-Key Configuration";
  }

  description
    "The Local PCE Entity PCE information";
}

uses authentication {
  description
    "Local PCEP entity authentication information";
}

uses pcep-entity-info {
  description
    "The configuration related to the PCEP
    entity.";
}

leaf pcep-notification-max-rate {
  type uint32;
  mandatory true;
  description
    "This variable indicates the maximum number of
    notifications issued per second. If events
    occur more rapidly, the implementation may
    simply fail to emit these notifications during
```

```
        that period, or may queue them until an
        appropriate time. A value of 0 means no
        notifications are emitted and all should be
        discarded (that is, not queued).";
    }

    container stateful-parameter{
        if-feature stateful;
        must "(/pcep/entity/capability/stateful/enabled" +
            " = true())"
        {
            error-message
                "The Stateful PCE must be enabled";
            description
                "When PCEP entity is stateful
                enabled";
        }
        uses stateful-pce-parameter;

        description
            "The configured stateful parameters";
    }

    container of-list{
        when "../role = 'pce'" +
            "or " +
            "../role = 'pcc-and-pce'"
        {
            description
                "Valid at PCE";
        }
        if-feature objective-function;

        uses of-list;

        description
            "The authorized OF-List at PCE for all peers";
    }

    container peers{
        when "../role = 'pcc'" +
            "or " +
            "../role = 'pcc-and-pce'"
        {
            description
                "Valid at PCC";
        }
    }
```

```
description
  "The list of configured peers for the
  entity (remote PCE)";
list peer{
  key "addr";

  description
    "The peer configured for the entity.
    (remote PCE)";

  leaf addr {
    type inet:ip-address;
    description
      "The local Internet address of this
      PCEP peer.";
  }

  leaf description {
    type string;
    description
      "Description of the PCEP peer
      configured by the user";
  }
  uses info {
    description
      "PCE Peer information";
  }
  uses pce-info {
    description
      "PCE Peer information";
  }

  leaf delegation-pref{
    if-feature stateful;
    type uint8{
      range "0..7";
    }
    must "(/pcep/entity/capability/stateful" +
      "/active = true())"
    {
      error-message
        "The Active Stateful PCE must be
        enabled";
      description
        "When PCEP entity is active stateful
        enabled";
    }
    description

```

```
        "The PCE peer delegation preference.";
    }
    uses authentication {
        description
            "PCE Peer authentication";
    }
    }//peer
} //peers
} //entity
} //pcep

/*
 * Operational data nodes
 */

container pcep-state{
    config false;
    description
        "The list of operational PCEP entities on the
        device.";

    container entity{
        description
            "The operational PCEP entity on the device.";

        leaf addr {
            type inet:ip-address;
            description
                "The local Internet address of this PCEP
                entity.
                If operating as a PCE server, the PCEP
                entity listens on this address.
                If operating as a PCC, the PCEP entity
                binds outgoing TCP connections to this
                address.
                It is possible for the PCEP entity to
                operate both as a PCC and a PCE Server, in
                which case it uses this address both to
                listen for incoming TCP connections and to
                bind outgoing TCP connections.";
        }

        leaf index{
            type uint32;
            description
                "The index of the operational PCEP
                entity";
        }
    }
}
```



```
leaf admin-status {
  type pcep-admin-status;
  description
    "The administrative status of this PCEP Entity.
    This is the desired operational status as
    currently set by an operator or by default in
    the implementation. The value of enabled
    represents the current status of an attempt
    to reach this desired status.";
}

leaf oper-status {
  type pcep-admin-status;
  description
    "The operational status of the PCEP entity.
    Takes one of the following values.
    - oper-status-up(1): the PCEP entity is
      active.
    - oper-status-down(2): the PCEP entity is
      inactive.
    - oper-status-going-up(3): the PCEP entity is
      activating.
    - oper-status-going-down(4): the PCEP entity is
      deactivating.
    - oper-status-failed(5): the PCEP entity has
      failed and will recover when possible.
    - oper-status-failed-perm(6): the PCEP entity
      has failed and will not recover without
      operator intervention.";
}

leaf role {
  type pcep-role;
  description
    "The role that this entity can play.
    Takes one of the following values.
    - unknown(0): this PCEP entity role is
      not known.
    - pcc(1): this PCEP entity is a PCC.
    - pce(2): this PCEP entity is a PCE.
    - pcc-and-pce(3): this PCEP entity is
      both a PCC and a PCE.";
}

leaf description {
  type string;
  description
    "Description of the PCEP entity configured
```

```
        by the user";
    }

    leaf speaker-entity-id{
        if-feature stateful-sync-opt;
        type string;
        description
            "The Speaker Entity Identifier";
    }

    uses info {
        description
            "Local PCEP entity information";
    }

    container pce-info {
        when "../role = 'pce'" +
            "or " +
            "../role = 'pcc-and-pce'"
        {
            description
                "Valid at PCE";
        }
        uses pce-info {
            description
                "Local PCE information";
        }

        container path-key {
            if-feature path-key;
            uses path-key {
                description
                    "Path-Key Configuration";
            }
            description
                "Path-Key Configuration";
        }

        description
            "The Local PCE Entity PCE information";
    }

    uses authentication {
        description
            "Local PCEP Entity authentication information";
    }

    uses pcep-entity-info{
```

```
        description
            "The operational information related to the
            PCEP entity.";
    }

    container stateful-parameter{
        if-feature stateful;
        must "(/pcep/entity/capability/stateful/enabled" +
            " = true())"
        {
            error-message
                "The Stateful PCE must be enabled";
            description
                "When PCEP entity is stateful
                enabled";
        }
        uses stateful-pce-parameter;

        description
            "The operational stateful parameters";
    }

    container lsp-db{
        if-feature stateful;
        description
            "The LSP-DB";
        leaf db-ver{
            when "../role = 'pcc'" +
                "or " +
                "../role = 'pcc-and-pce'"
            {
                description
                    "Valid at PCC";
            }
            if-feature stateful-sync-opt;
            type uint64;
            description
                "The LSP State Database Version Number";
        }
        list association-list {
            key "id source global-source extended-id";
            description
                "List of all PCEP associations";
            uses association {
                description
                    "The Association attributes";
            }
        }
    }
}
```

```

    }
    list lsp {
        key "plsp-id pcc-id";
        description
            "List of all LSP in this association";
        leaf plsp-id {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                    + "lsp/plsp-id";
            }
            description
                "Reference to PLSP-ID in LSP-DB";
        }
        leaf pcc-id {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                    + "lsp/pcc-id";
            }
            description
                "Reference to PCC-ID in LSP-DB";
        }
    }
}
list lsp{
    key "plsp-id pcc-id";
    description
        "List of all LSPs in LSP-DB";
    uses lsp-state{
        description
            "The PCEP specific attributes for
            LSP-DB.";
    }
    list association-list {
        key "id source global-source extended-id";
        description
            "List of all PCEP associations";
        uses association-ref {
            description
                "Reference to the Association
                attributes";
        }
    }
}
}
container path-keys {
    when "../role = 'pce'" +
        "or " +

```

```
        "../role = 'pcc-and-pce'"
    {
        description
            "Valid at PCE";
    }
    if-feature path-key;
    uses path-key-state;
    description
        "The path-keys generated by the PCE";
}
container of-list{
    when "../role = 'pce'" +
        "or " +
        "../role = 'pcc-and-pce'"
    {
        description
            "Valid at PCE";
    }
    if-feature objective-function;

    uses of-list;

    description
        "The authorized OF-List at PCE for all peers";
}
container peers{
    description
        "The list of peers for the entity";

    list peer{
        key "addr";

        description
            "The peer for the entity.";

        leaf addr {
            type inet:ip-address;
            description
                "The local Internet address of this PCEP
                peer.";
        }

        leaf role {
            type pcep-role;
            description
                "The role of the PCEP Peer.
                Takes one of the following values.
                - unknown(0): this PCEP peer role";
        }
    }
}
```

```
        is not known.
        - pcc(1): this PCEP peer is a PCC.
        - pce(2): this PCEP peer is a PCE.
        - pcc-and-pce(3): this PCEP peer
          is both a PCC and a PCE.";
    }

    uses info {
        description
            "PCEP peer information";
    }

    container pce-info {
        when "../role = 'pce'" +
            "or " +
            "../role = 'pcc-and-pce'"
        {
            description
                "When the peer is PCE";
        }

        uses pce-info {
            description
                "PCE Peer information";
        }
    }
    description
        "The PCE Peer information";
}

leaf delegation-pref{
    when "../role = 'pce'" +
        "or " +
        "../role = 'pcc-and-pce'"
    {
        description
            "When the peer is PCE";
    }
    if-feature stateful;
    type uint8{
        range "0..7";
    }
    must "(/pcep/entity/capability/stateful" +
        "/active = true())"
    {
        error-message
            "The Active Stateful PCE must be
```

```
        enabled";
    description
        "When PCEP entity is active stateful
        enabled";
    }
    description
        "The PCE peer delegation preference.";
}

uses authentication {
    description
        "PCE Peer authentication";
}

leaf discontinuity-time {
    type yang:timestamp;
    description
        "The timestamp of the time when the
        information and statistics were
        last reset.";
}

leaf initiate-session {
    type boolean;
    description
        "Indicates whether the local PCEP
        entity initiates sessions to this peer,
        or waits for the peer to initiate a
        session.";
}

leaf session-exists{
    type boolean;
    description
        "Indicates whether a session with
        this peer currently exists.";
}

leaf session-up-time{
    type yang:timestamp;
    description
        "The timestamp value of the last time a
        session with this peer was successfully
        established.";
}

leaf session-fail-time{
    type yang:timestamp;
```

```
    description
      "The timestamp value of the last time a
       session with this peer failed to be
       established.";
  }

  leaf session-fail-up-time{
    type yang:timestamp;
    description
      "The timestamp value of the last time a
       session with this peer failed from
       active.";
  }

  container sessions {
    description
      "This entry represents a single PCEP
       session in which the local PCEP entity
       participates.
       This entry exists only if the
       corresponding PCEP session has been
       initialized by some event, such as
       manual user configuration, auto-
       discovery of a peer, or an incoming
       TCP connection.";

    list session {
      key "initiator";

      description
        "The list of sessions, note that
         for a time being two sessions
         may exist for a peer";

      leaf initiator {
        type pcep-initiator;
        description
          "The initiator of the session,
           that is, whether the TCP
           connection was initiated by
           the local PCEP entity or the
           peer.
           There is a window during
           session initialization where
```



```
        two sessions can exist between
        a pair of PCEP speakers, each
        initiated by one of the
        speakers. One of these
        sessions is always discarded
        before it leaves OpenWait
        state. However, before it is
        discarded, two sessions to the
        given peer appear transiently
        in this MIB module. The
        sessions are distinguished by
        who initiated them, and so this
        field is the key.";
    }
    leaf role {
        type leafref {
            path "../.../role";
        }
        description
            "The reference to peer role .";
    }

    leaf state-last-change {
        type yang:timestamp;
        description
            "The timestamp value at the
            time this session entered its
            current state as denoted by
            the state leaf.";
    }

    leaf state {
        type pcep-sess-state;
        description
            "The current state of the
            session.
            The set of possible states
            excludes the idle state since
            entries do not exist in the
            idle state.";
    }

    leaf session-creation {
        type yang:timestamp;
        description
            "The timestamp value at the
            time this session was
            created.";
```

```
}  
  
leaf connect-retry {  
    type yang:counter32;  
    description  
        "The number of times that the  
        local PCEP entity has  
        attempted to establish a TCP  
        connection for this session  
        without success. The PCEP  
        entity gives up when this  
        reaches connect-max-retry.";  
}  
  
leaf local-id {  
    type uint32 {  
        range "0..255";  
    }  
    description  
        "The value of the PCEP session  
        ID used by the local PCEP  
        entity in the Open message  
        for this session.  
        If state is tcp-pending then  
        this is the session ID that  
        will be used in the Open  
        message. Otherwise, this is  
        the session ID that was sent  
        in the Open message.";  
}  
  
leaf remote-id {  
    type uint32 {  
        range "0..255";  
    }  
    must "((../state != 'tcp-pending'" +  
        "and " +  
        "../state != 'open-wait' )" +  
        "or " +  
        "((../state = 'tcp-pending'" +  
        " or " +  
        "../state = 'open-wait' )" +  
        "and (. = 0)))" {  
        error-message  
            "Invalid remote-id";  
        description  
            "If state is tcp-  
            pending or open-wait
```

```

        then this leaf is not
        used and MUST be set
        to zero.";
    }
    description
        "The value of the PCEP session
        ID used by the peer in its
        Open message for this
        session.";
}

leaf keepalive-timer {
    type uint32 {
        range "0..255";
    }
    units "seconds";
    must "(../state = 'session-up' +
        'or ' +
        '(../state != 'session-up' +
        'and (.= 0)))" {
        error-message
            "Invalid keepalive
            timer";
        description
            "This field is used if
            and only if state is
            session-up. Otherwise,
            it is not used and
            MUST be set to
            zero.";
    }
    description
        "The agreed maximum interval at
        which the local PCEP entity
        transmits PCEP messages on
        this PCEP session. Zero means
        that the local PCEP entity
        never sends Keepalives on this
        session.";
}

leaf peer-keepalive-timer {
    type uint32 {
        range "0..255";
    }
    units "seconds";
    must "(../state = 'session-up' +
        'or ' +

```

```

        "(../state != 'session-up'" +
        "and " +
        "(. = 0)))" {
            error-message
                "Invalid Peer keepalive
                timer";
            description
                "This field is used if
                and only if state is
                session-up. Otherwise,
                it is not used and MUST
                be set to zero.";
        }
    description
        "The agreed maximum interval at
        which the peer transmits PCEP
        messages on this PCEP session.
        Zero means that the peer never
        sends Keepalives on this
        session.";
}

leaf dead-timer {
    type uint32 {
        range "0..255";
    }
    units "seconds";
    description
        "The dead timer interval for
        this PCEP session.";
}

leaf peer-dead-timer {
    type uint32 {
        range "0..255";
    }
    units "seconds";
    must "((../state != 'tcp-pending'" +
        "and " +
        "../state != 'open-wait' )" +
        "or " +
        "((../state = 'tcp-pending'" +
        " or " +
        "../state = 'open-wait' )" +
        "and " +
        "(. = 0)))" {
        error-message
            "Invalid Peer Dead

```

```
        timer";
    description
        "If state is tcp-
        pending or open-wait
        then this leaf is not
        used and MUST be set to
        zero.";
    }
    description
        "The peer's dead-timer interval
        for this PCEP session.";
}

leaf ka-hold-time-rem {
    type uint32 {
        range "0..255";
    }
    units "seconds";
    must "((../state != 'tcp-pending' " +
        "and " +
        "../state != 'open-wait' ) " +
        "or " +
        "((../state = 'tcp-pending' " +
        "or " +
        "../state = 'open-wait' )" +
        "and " +
        "(. = 0)))" {
        error-message
            "Invalid Keepalive hold
            time remaining";
        description
            "If state is tcp-pending
            or open-wait then this
            field is not used and
            MUST be set to zero.";
    }
    description
        "The keep alive hold time
        remaining for this session.";
}

leaf overloaded {
    type boolean;
    description
        "If the local PCEP entity has
        informed the peer that it is
        currently overloaded, then
        this is set to true.
```

```
        Otherwise, it is set to
        false.";
    }

    leaf overload-time {
        type uint32;
        units "seconds";
        must "((../overloaded = true()) " +
            "or ((../overloaded != true()) " +
            "and (. = 0)))" {
            error-message
                "Invalid overload-time";
            description
                "This field is only used
                if overloaded is set to
                true. Otherwise, it is
                not used and MUST be
                set to zero.";
        }
        description
            "The interval of time that is
            remaining until the local PCEP
            entity will cease to be
            overloaded on this session.";
    }

    leaf peer-overloaded {
        type boolean;
        description
            "If the peer has informed the
            local PCEP entity that it is
            currently overloaded, then
            this is set to true.
            Otherwise, it is set to
            false.";
    }

    leaf peer-overload-time {
        type uint32;
        units "seconds";
        must "((../peer-overloaded = " +
            "true()) or " +
            "((../peer-overloaded != " +
            "true())" +
            " and " +
            "(. = 0)))" {
            error-message
                "Invalid peer overload
```

```
        time";
    description
        "This field is only used
        if peer-overloaded is
        set to true. Otherwise,
        it is not used and MUST
        be set to zero.";
    }
    description
        "The interval of time that is
        remaining until the peer will
        cease to be overloaded. If it
        is not known how long the peer
        will stay in overloaded state,
        this leaf is set to zero.";
}
leaf lspdb-sync {
    if-feature stateful;
    type sync-state;
    description
        "The LSP-DB state
        synchronization status.";
}
leaf recv-db-ver{
    when "../role = 'pcc'" +
        "or " +
        "../role = 'pcc-and-pce'"
    {
        description
            "Valid for PCEP Peer as
            PCC";
    }

    if-feature stateful;
    if-feature stateful-sync-opt;

    type uint64;

    description
        "The last received LSP State
        Database Version Number";
}

container of-list{
    when "../role = 'pce'" +
        "or " +
        "../role = 'pcc-and-pce'"
    {
```

```

        description
            "Valid for PCEP Peer as
            PCE";
    }
    if-feature objective-function;

    uses of-list;

    description
        "Indicate the list of supported
        OF on this session";
    }

    leaf speaker-entity-id{
        if-feature stateful-sync-opt;
        type string;
        description
            "The Speaker Entity Identifier";
    }

    leaf discontinuity-time {
        type yang:timestamp;
        description
            "The timestamp value of the
            time when the statistics were
            last reset.";
    }
    } // session
    } // sessions
    } // peer
    } // peers
    } // entity
} // pcep-state

/*
 * Notifications
 */
notification pcep-session-up {
    description
        "This notification is sent when the value of
        '/pcep/pcep-state/peers/peer/sessions/session/state'
        enters the 'session-up' state.";

    uses notification-instance-hdr;

    uses notification-session-hdr;

    leaf state-last-change {

```



```
        type yang:timestamp;
        description
            "The timestamp value at the time this session
            entered its current state as denoted by the state
            leaf.";
    }

    leaf state {
        type pcep-sess-state;
        description
            "The current state of the session.
            The set of possible states excludes the idle state
            since entries do not exist in the idle state.";
    }
} //notification

notification pcep-session-down {
    description
        "This notification is sent when the value of
        '/pcep/pcep-state/peers/peer/sessions/session/state'
        leaves the 'session-up' state.";

    uses notification-instance-hdr;

    leaf session-initiator {
        type pcep-initiator;
        description
            "The initiator of the session.";
    }

    leaf state-last-change {
        type yang:timestamp;
        description
            "The timestamp value at the time this session
            entered its current state as denoted by the state
            leaf.";
    }

    leaf state {
        type pcep-sess-state;
        description
            "The current state of the session.
            The set of possible states excludes the idle state
            since entries do not exist in the idle state.";
    }
} //notification

notification pcep-session-local-overload {
```

```
    description
      "This notification is sent when the local PCEP entity
       enters overload state for a peer.";

    uses notification-instance-hdr;

    uses notification-session-hdr;

    leaf overloaded {
      type boolean;
      description
        "If the local PCEP entity has informed the peer
         that it is currently overloaded, then this is set
         to true. Otherwise, it is set to false.";
    }

    leaf overload-time {
      type uint32;
      units "seconds";
      description
        "The interval of time that is remaining until the
         local PCEP entity will cease to be overloaded on
         this session.";
    }
  } //notification

notification pcep-session-local-overload-clear {
  description
    "This notification is sent when the local PCEP entity
     leaves overload state for a peer.";

  uses notification-instance-hdr;

  leaf overloaded {
    type boolean;
    description
      "If the local PCEP entity has informed the peer
       that it is currently overloaded, then this is set
       to true. Otherwise, it is set to false.";
  }
} //notification

notification pcep-session-peer-overload {
  description
    "This notification is sent when a peer enters overload
     state.";

  uses notification-instance-hdr;
```

```
    uses notification-session-hdr;

    leaf peer-overloaded {
        type boolean;
        description
            "If the peer has informed the local PCEP entity that
             it is currently overloaded, then this is set to
             true. Otherwise, it is set to false.";
    }

    leaf peer-overload-time {
        type uint32;
        units "seconds";
        description
            "The interval of time that is remaining until the
             peer will cease to be overloaded. If it is not
             known how long the peer will stay in overloaded
             state, this leaf is set to zero.";
    }
} //notification

notification pcep-session-peer-overload-clear {
    description
        "This notification is sent when a peer leaves overload
         state.";

    uses notification-instance-hdr;

    leaf peer-overloaded {
        type boolean;
        description
            "If the peer has informed the local PCEP entity that
             it is currently overloaded, then this is set to
             true. Otherwise, it is set to false.";
    }
} //notification

/*
 * RPC
 */

rpc trigger-resync {
    if-feature stateful;
    if-feature stateful-sync-opt;
    description
        "Trigger the resyncrinization at the PCE";
    input {
        leaf pcc {
```

```
        type leafref {
            path "/pcep-state/entity/peers/peer/addr";
        }
        description
            "The IP address to identify the PCC. The state
            synchronization is re-triggered for all LSPs from
            the PCC. The rpc on the PCC will be ignored.";
    }
}
} //rpc
} //module
```

<CODE ENDS>

9.2. ietf-pcep-stats module

```
<CODE BEGINS> file "ietf-pcep-stats@2017-03-12.yang"
module ietf-pcep-stats {

    yang-version 1.1;

    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep-stats";
    prefix ps;

    import ietf-pcep {
        prefix p;
    }

    import ietf-yang-types {
        prefix "yang";
    }

    organization
        "IETF PCE (Path Computation Element) Working Group";

    contact
        "WG Web:    <http://tools.ietf.org/wg/pce/>
        WG List:    <mailto:pce@ietf.org>
        WG Chair:   JP Vasseur
                   <mailto:jpv@cisco.com>
```

```
WG Chair: Julien Meuric
          <mailto:julien.meuric@orange.com>
WG Chair: Jonathan Hardwick
          <mailto:Jonathan.Hardwick@metaswitch.com>
Editor:   Dhruv Dhody
          <mailto:dhruv.ietf@gmail.com>;
```

description

```
"The YANG module augments the PCEP yang operational
model with statistics, counters and telemetry data";
```

revision 2017-03-12 {

```
  description "Initial revision.";
```

reference

```
    "RFC XXXX:  A YANG Data Model for Path Computation
                Element Communications Protocol
                (PCEP)";
```

}

```
/*
```

```
 * Groupings
```

```
*/
```

grouping pcep-stats{

description

```
    "This grouping defines statistics for PCEP. It is used
    for both peer and current session.";
```

leaf avg-rsp-time{

```
    when "../p:role = 'pce' " +
        "or " +
        "../p:role = 'pcc-and-pce' "
```

```
    {
```

description

```
        "Valid for PCEP Peer as PCE";
```

```
    }
```

```
    type uint32;
```

```
    units "milliseconds";
```

description

```
    "The average response time.
    If an average response time has not been
    calculated then this leaf has the value zero.";
```

```
}
```

```
leaf lwm-rsp-time{
  when "../..//p:role = 'pce'" +
    "or " +
    "../..//p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type uint32;
  units "milliseconds";

  description
    "The smallest (low-water mark) response time seen.
    If no responses have been received then this
    leaf has the value zero.";
}

leaf hwm-rsp-time{
  when "../..//p:role = 'pce'" +
    "or " +
    "../..//p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type uint32;
  units "milliseconds";

  description
    "The greatest (high-water mark) response time seen.
    If no responses have been received then this object
    has the value zero.";
}

leaf num-pcreq-sent{
  when "../..//p:role = 'pce'" +
    "or " +
    "../..//p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCReq messages sent.";
}

leaf num-pcreq-rcvd{
```

```
    when "../..//p:role = 'pcc'" +
      "or " +
      "../..//p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of PCReq messages received.";
  }

  leaf num-pcrep-sent{
    when "../..//p:role = 'pcc'" +
      "or " +
      "../..//p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCC";
    }

    type yang:counter32;
    description
      "The number of PCRep messages sent.";
  }

  leaf num-pcrep-rcvd{
    when "../..//p:role = 'pce'" +
      "or " +
      "../..//p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }

    type yang:counter32;
    description
      "The number of PCRep messages received.";
  }

  leaf num-pcerr-sent{
    type yang:counter32;
    description
      "The number of PCErr messages sent.";
  }

  leaf num-pcerr-rcvd{
    type yang:counter32;
```

```
        description
            "The number of PCErr messages received.";
    }

    leaf num-pcntf-sent{
        type yang:counter32;
        description
            "The number of PCNtf messages sent.";
    }

    leaf num-pcntf-rcvd{
        type yang:counter32;
        description
            "The number of PCNtf messages received.";
    }

    leaf num-keepalive-sent{
        type yang:counter32;
        description
            "The number of Keepalive messages sent.";
    }

    leaf num-keepalive-rcvd{
        type yang:counter32;
        description
            "The number of Keepalive messages received.";
    }

    leaf num-unknown-rcvd{
        type yang:counter32;
        description
            "The number of unknown messages received.";
    }

    leaf num-corrupt-rcvd{
        type yang:counter32;
        description
            "The number of corrupted PCEP message received.";
    }

    leaf num-req-sent{
        when "../..p:role = 'pce'" +
            "or " +
            "../..p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
    }
```



```
    type yang:counter32;
    description
      "The number of requests sent. A request corresponds
      1:1 with an RP object in a PCReq message. This might
      be greater than num-pcreq-sent because multiple
      requests can be batched into a single PCReq
      message.";
  }

  leaf num-req-sent-pend-rep{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of requests that have been sent for
      which a response is still pending.";
  }

  leaf num-req-sent-ero-rcvd{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of requests that have been sent for
      which a response with an ERO object was received.
      Such responses indicate that a path was
      successfully computed by the peer.";
  }

  leaf num-req-sent-nopath-rcvd{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
```

```
    description
      "The number of requests that have been sent for
      which a response with a NO-PATH object was
      received. Such responses indicate that the peer
      could not find a path to satisfy the
      request.";
  }

  leaf num-req-sent-cancel-rcvd{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of requests that were cancelled with
      a PCNtf message.
      This might be different than num-pcntf-rcvd because
      not all PCNtf messages are used to cancel requests,
      and a single PCNtf message can cancel multiple
      requests.";
  }

  leaf num-req-sent-error-rcvd{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of requests that were rejected with a
      PCErr message.
      This might be different than num-pcerr-rcvd because
      not all PCErr messages are used to reject requests,
      and a single PCErr message can reject multiple
      requests.";
  }

  leaf num-req-sent-timeout{
    when "../p:role = 'pce'" +
      "or " +
      "../p:role = 'pcc-and-pce'"
```

```
{
  description
    "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
  "The number of requests that have been sent to a peer
  and have been abandoned because the peer has taken too
  long to respond to them.";
}

leaf num-req-sent-cancel-sent{
  when "../..p:role = 'pce'" +
    "or " +
    "../..p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of requests that were sent to the peer and
    explicitly cancelled by the local PCEP entity sending
    a PCNtf.";
}

leaf num-req-rcvd{
  when "../..p:role = 'pcc'" +
    "or " +
    "../..p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests received. A request
    corresponds 1:1 with an RP object in a PCReq
    message.
    This might be greater than num-pcreq-rcvd because
    multiple requests can be batched into a single
    PCReq message.";
}

leaf num-req-rcvd-pend-rep{
  when "../..p:role = 'pcc'" +
    "or " +
    "../..p:role = 'pcc-and-pce'"
```

```
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of requests that have been received for
       which a response is still pending.";
  }

  leaf num-req-rcvd-ero-sent{
    when "../..p:role = 'pcc'" +
      "or " +
      "../..p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of requests that have been received for
       which a response with an ERO object was sent. Such
       responses indicate that a path was successfully
       computed by the local PCEP entity.";
  }

  leaf num-req-rcvd-nopath-sent{
    when "../..p:role = 'pcc'" +
      "or " +
      "../..p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of requests that have been received for
       which a response with a NO-PATH object was sent. Such
       responses indicate that the local PCEP entity could
       not find a path to satisfy the request.";
  }

  leaf num-req-rcvd-cancel-sent{
    when "../..p:role = 'pcc'" +
      "or " +
      "../..p:role = 'pcc-and-pce'"
    {
      description
```

```
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests received that were cancelled
        by the local PCEP entity sending a PCNtf message.
        This might be different than num-pcntf-sent because
        not all PCNtf messages are used to cancel requests,
        and a single PCNtf message can cancel multiple
        requests.";
}

leaf num-req-rcvd-error-sent{
    when "../p:role = 'pcc'" +
        "or " +
        "../p:role = 'pcc-and-pce'"
    {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests received that were cancelled
        by the local PCEP entity sending a PCErr message.
        This might be different than num-pcerr-sent because
        not all PCErr messages are used to cancel requests,
        and a single PCErr message can cancel multiple
        requests.";
}

leaf num-req-rcvd-cancel-rcvd{
    when "../p:role = 'pcc'" +
        "or " +
        "../p:role = 'pcc-and-pce'"
    {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of requests that were received from the
        peer and explicitly cancelled by the peer sending
        a PCNtf.";
}

leaf num-rep-rcvd-unknown{
    when "../p:role = 'pce'" +
        "or " +
```

```
        ".../.../p:role = 'pcc-and-pce'"
    {
        description
            "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of responses to unknown requests
        received. A response to an unknown request is a
        response whose RP object does not contain the
        request ID of any request that is currently
        outstanding on the session.";
}

leaf num-req-rcvd-unknown{
    when ".../.../p:role = 'pcc'" +
        "or " +
        ".../.../p:role = 'pcc-and-pce'"
    {
        description
            "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of unknown requests that have been
        received. An unknown request is a request
        whose RP object contains a request ID of
        zero.";
}

container svec{
    if-feature p:svec;
    description
        "If synchronized path computation is supported";
    leaf num-svec-sent{
        when ".../.../.../p:role = 'pce'" +
            "or " +
            ".../.../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of SVEC objects sent in PCReq messages.
            An SVEC object represents a set of synchronized
            requests.";
    }
}
```

```
leaf num-svec-req-sent{
  when "../.../p:role = 'pce'" +
    "or " +
    "../.../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of requests sent that appeared in one
    or more SVEC objects.";
}

leaf num-svec-rcvd{
  when "../.../p:role = 'pcc'" +
    "or " +
    "../.../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of SVEC objects received in PCReq
    messages. An SVEC object represents a set of
    synchronized requests.";
}

leaf num-svec-req-rcvd{
  when "../.../p:role = 'pcc'" +
    "or " +
    "../.../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests received that appeared
    in one or more SVEC objects.";
}
}

container stateful{
  if-feature p:stateful;
  description
    "Stateful PCE related statistics";
  leaf num-pcrpt-sent{
```

```
when "../../../p:role = 'pce'" +
  "or " +
  "../../../p:role = 'pcc-and-pce'"
{
  description
    "Valid for PCEP Peer as PCE";
}
type yang:counter32;
description
  "The number of PCRpt messages sent.";
}

leaf num-pcrpt-rcvd{
  when "../../../p:role = 'pcc'" +
    "or " +
    "../../../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCRpt messages received.";
}

leaf num-pcupd-sent{
  when "../../../p:role = 'pcc'" +
    "or " +
    "../../../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCUpd messages sent.";
}

leaf num-pcupd-rcvd{
  when "../../../p:role = 'pce'" +
    "or " +
    "../../../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
```



```
        "The number of PCUpd messages received.";
    }

    leaf num-rpt-sent{
        when "../.../p:role = 'pce'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of LSP Reports sent.  A LSP report
            corresponds 1:1 with an LSP object in a PCRpt
            message. This might be greater than
            num-pcrpt-sent because multiple reports can
            be batched into a single PCRpt message.";
    }

    leaf num-rpt-rcvd{
        when "../.../p:role = 'pcc'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of LSP Reports received.  A LSP report
            corresponds 1:1 with an LSP object in a PCRpt
            message.
            This might be greater than num-pcrpt-rcvd because
            multiple reports can be batched into a single
            PCRpt message.";
    }

    leaf num-rpt-rcvd-error-sent{
        when "../.../p:role = 'pcc'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
```

```
        "The number of reports of LSPs received that were
        responded by the local PCEP entity by sending a
        PCErr message.";
    }

    leaf num-upd-sent{
        when "../.../p:role = 'pcc'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of LSP updates sent. A LSP update
            corresponds 1:1 with an LSP object in a PCUpd
            message. This might be greater than
            num-pcupd-sent because multiple updates can
            be batched into a single PCUpd message.";
    }

    leaf num-upd-rcvd{
        when "../.../p:role = 'pce'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of LSP Updates received. A LSP update
            corresponds 1:1 with an LSP object in a PCUpd
            message.
            This might be greater than num-pcupd-rcvd because
            multiple updates can be batched into a single
            PCUpd message.";
    }

    leaf num-upd-rcvd-unknown{
        when "../.../p:role = 'pce'" +
            "or " +
            "../.../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
    }
```

```
    type yang:counter32;
    description
      "The number of updates to unknown LSPs
      received. An update to an unknown LSP is a
      update whose LSP object does not contain the
      PLSP-ID of any LSP that is currently
      present.";
  }

  leaf num-upd-rcvd-undelegated{
    when "../.../p:role = 'pce'" +
      "or " +
      "../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of updates to not delegated LSPs
      received. An update to an undelegated LSP is a
      update whose LSP object does not contain the
      PLSP-ID of any LSP that is currently
      delegated to current PCEP session.";
  }

  leaf num-upd-rcvd-error-sent{
    when "../.../p:role = 'pce'" +
      "or " +
      "../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of updates to LSPs received that were
      responded by the local PCEP entity by sending a
      PCErr message.";
  }
  container initiation {
    if-feature p:pce-initiated;
    description
      "PCE-Initiated related statistics";
    leaf num-pcinitiate-sent{
      when "../.../.../p:role = 'pcc'" +
        "or " +
        "../.../.../p:role = 'pcc-and-pce'"
    }
```

```
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of PCInitiate messages sent.";
  }

  leaf num-pcinitiate-rcvd{
    when "../.../.../p:role = 'pce'" +
      "or " +
      "../.../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of PCInitiate messages received.";
  }

  leaf num-initiate-sent{
    when "../.../.../p:role = 'pcc'" +
      "or " +
      "../.../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of LSP Initiation sent via PCE.
      A LSP initiation corresponds 1:1 with an LSP
      object in a PCInitiate message. This might be
      greater than num-pcinitiate-sent because
      multiple initiations can be batched into a
      single PCInitiate message.";
  }

  leaf num-initiate-rcvd{
    when "../.../.../p:role = 'pce'" +
      "or " +
      "../.../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
  }
```

```
    type yang:counter32;
    description
      "The number of LSP Initiation received from
      PCE. A LSP initiation corresponds 1:1 with
      an LSP object in a PCInitiate message. This
      might be greater than num-pcinitiate-rcvd
      because multiple initiations can be batched
      into a single PCInitiate message.";
  }

  leaf num-initiate-rcvd-error-sent{
    when "../.../.../p:role = 'pce'" +
      "or " +
      "../.../.../p:role = 'pcc-and-pce'"
    {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of initiations of LSPs received
      that were responded by the local PCEP entity
      by sending a PCErr message.";
  }
}

container path-key {
  when "../.../p:role = 'pcc'" +
    "or " +
    "../.../p:role = 'pcc-and-pce'"
  {
    description
      "Valid for PCEP Peer as PCC";
  }
  if-feature p:path-key;
  description
    "If Path-Key is supported";
  leaf num-unknown-path-key{
    type yang:counter32;
    description
      "The number of attempts to expand an unknown
      path-key.";
  }
  leaf num-exp-path-key{
    type yang:counter32;
    description
      "The number of attempts to expand an expired
      path-key.";
```

```
    }
    leaf num-dup-path-key{
      type yang:counter32;
      description
        "The number of duplicate attempts to expand same
        path-key.";
    }
    leaf num-path-key-no-attempt{
      type yang:counter32;
      description
        "The number of expired path-keys with no attempt to
        expand it.";
    }
  }
} //pcep-stats

/*
 * Augment modules to add statistics
 */

augment "/p:pcep-state/p:entity/p:peers/p:peer" {
  description
    "Augmenting the statistics";
  leaf num-sess-setup-ok{
    type yang:counter32;
    description
      "The number of PCEP sessions successfully
      successfully established with the peer,
      including any current session. This
      counter is incremented each time a
      session with this peer is successfully
      established.";
  }

  leaf num-sess-setup-fail{
    type yang:counter32;
    description
      "The number of PCEP sessions with the peer
      that have been attempted but failed
      before being fully established. This
      counter is incremented each time a
      session retry to this peer fails.";
  }
}
container pcep-stats {
  description
    "The container for all statistics at peer
    level.";
  uses pcep-stats{

```

```
        description
            "Since PCEP sessions can be
            ephemeral, the peer statistics tracks
            a peer even when no PCEP session
            currently exists to that peer. The
            statistics contained are an aggregate
            of the statistics for all successive
            sessions to that peer.";
    }

    leaf num-req-sent-closed{
        when "../p:role = 'pce'" +
            "or " +
            "../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests that were
            sent to the peer and implicitly
            cancelled when the session they were
            sent over was closed.";
    }

    leaf num-req-rcvd-closed{
        when "../p:role = 'pcc'" +
            "or " +
            "../p:role = 'pcc-and-pce'"
        {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of requests that were
            received from the peer and
            implicitly cancelled when the
            session they were received over
            was closed.";
    }
} //pcep-stats
} //augment

augment "/p:pcep-state/p:entity/p:peers/p:peer/" +
    "p:sessions/p:session" {
    description
```

```
        "Augmenting the statistics";
    container pcep-stats {
        description
            "The container for all statistics
             at session level.";
        uses pcep-stats{
            description
                "The statistics contained are
                 for the current sessions to
                 that peer. These are lost
                 when the session goes down.
                 ";
        }
    } //pcep-stats
} //augment
} //module
```

<CODE ENDS>

10. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/pcep/entity/ - configure local parameters, capabilities etc.

/pcep/entity/peers - configure remote peers to setup PCEP session.

Unauthorized access to above list can adversely affect the PCEP session between the local entity and the peers. This may lead to

inability to compute new paths, stateful operations on the delegated as well as PCE-initiated LSPs.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/pcep-state/lsp-db - All the LSPs in the network. Unauthorized access to this could provide the all path and network usage information.

/pcep-state/path-keys/ - The Confidential Path Segments (CPS) are hidden using path-keys. Unauthorized access to this could leak confidential path information.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

trigger-resync - trigger resynchronization with the PCE.
Unauthorized access to this could force a PCEP session into continuous state synchronization.

11. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-pcep

Registrant Contact: The PCE WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [RFC6020].

Name:	ietf-pcep
Namespace:	urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix:	pcep
Reference:	This I-D

12. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. Further this document structure is based on Routing Yang Module [I-D.ietf-netmod-routing-cfg]. We would like to thank the authors of aforementioned documents.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<http://www.rfc-editor.org/info/rfc6991>>.
- [I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-18 (work in progress), December 2016.

- [I-D.ietf-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-09 (work in progress), March 2017.
- [I-D.ietf-pce-lsp-setup-type]
Sivabalan, S., Medved, J., Minei, I., Crabbe, E., Varga, R., Tantsura, J., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-03 (work in progress), June 2015.
- [I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Raszuk, R., Lopez, V., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-08 (work in progress), October 2016.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-06 (work in progress), March 2017.
- [I-D.ietf-rtgwg-yang-key-chain]
Lindem, A., Qu, Y., Yeung, D., Chen, I., Zhang, Z., and Y. Yang, "Routing Key Chain YANG Data Model", draft-ietf-rtgwg-yang-key-chain-15 (work in progress), February 2017.

13.2. Informative References

- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <<http://www.rfc-editor.org/info/rfc7420>>.

[I-D.ietf-netmod-routing-cfg] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-25 (work in progress), November 2016.

[I-D.ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-12 (work in progress), March 2017.

Appendix A. Contributor Addresses

Rohit Pobbathi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: rohit.pobbathi@huawei.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: vinods.kumar@huawei.com

Zafar Ali
Cisco Systems
Canada

EMail: zali@cisco.com

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean VA 22102
USA

EMail: Xufeng_Liu@jabil.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
EMail: leeyoung@huawei.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: udayasree.palle@huawei.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

EMail: zhang.xian@huawei.com

Avantika
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: avantika.sushilkumar@huawei.com

Authors' Addresses

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

Vishnu Pavan Beeram
Juniper Networks
USA

EMail: vbeeram@juniper.net

Jeff Tantsura
USA

EMail: jefftant@gmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: 29 July 2022

D. Dhody, Ed.
Huawei Technologies
J. Hardwick
Metaswitch
V. Beeram
Juniper Networks
J. Tantsura
Microsoft
25 January 2022

A YANG Data Model for Path Computation Element Communications Protocol
(PCEP)
draft-ietf-pce-pcep-yang-18

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration and state data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	3
3. Terminology and Notation	3
3.1. Tree Diagrams	5
3.2. Prefixes in Data Node Names	5
3.3. References in the Model	6
4. Objectives	8
5. The Design of PCEP Data Model	9
5.1. The Overview of PCEP Data Model	9
5.2. The Entity	10
5.3. The Peer Lists	14
5.4. The Session Lists	16
5.5. Notifications	18
5.6. RPC	18
6. The Design of PCEP Statistics Data Model	19
7. Advanced PCE Features	22
7.1. Stateful PCE's LSP-DB	22
8. Other Considerations	23
8.1. PCEP over TLS (PCEPS)	23
9. PCEP YANG Modules	23
9.1. ietf-pcep module	23
9.2. ietf-pcep-stats module	79
10. Security Considerations	97
11. IANA Considerations	99
12. Implementation Status	100
13. Acknowledgements	100
14. References	101
14.1. Normative References	101
14.2. Informative References	104
Appendix A. The Full PCEP Data Model	105
Appendix B. Example	112
Appendix C. Contributor Addresses	115
Authors' Addresses	116

1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path Computation Client (PCC) may make requests to a PCE for paths to be computed.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE). [RFC8231] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC7950] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data.

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

The PCEP operational state is included in the same tree as the PCEP configuration consistent with Network Management Datastore Architecture (NMDA) [RFC8342]. The origin of the data is indicated as per the origin metadata annotation.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- * Path Computation Request message (PCReq).
- * Path Computation Reply message (PCRep).
- * Notification message (PCNtf).

- * Error message (PCErr).
- * Request Parameters object (RP).
- * Synchronization Vector object (SVEC).
- * Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- * PCEP entity: a local PCEP speaker.
- * PCEP peer: to refer to a remote PCEP speaker.
- * PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [RFC8231] :

- * Stateful PCE, Passive Stateful PCE, Active Stateful PCE
- * Delegation, Revocation, Redelegation
- * LSP State Report, Path Computation Report message (PCRpt).
- * LSP State Update, Path Computation Update message (PCUpd).
- * PLSP-ID: a PCEP-specific identifier for the LSP.
- * SRP: Stateful PCE Request Parameters

[RFC8281] :

- * PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[RFC8408] :

- * Path Setup Type (PST).

[RFC8664] :

- * Segment Routing (SR).

[RFC5541] :

- * Objective Function (OF).

[RFC8697] :

* Association.

[RFC6241] :

* Configuration data.

* State data.

3.1. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	[RFC8776]
key-chain	ietf-key-chain	[RFC8177]
nacm	ietf-netconf-acm	[RFC8341]
tlss	ietf-tls-server	[I-D.ietf-netconf-tls-client-server]
tlsc	ietf-tls-client	[I-D.ietf-netconf-tls-client-server]
ospf	ietf-ospf	[I-D.ietf-ospf-yang]
isis	ietf-isis	[I-D.ietf-isis-yang-isis-cfg]

Table 1: Prefixes and corresponding YANG modules

3.3. References in the Model

Following documents are referenced in the model defined in this document -

Documents	Reference
OSPF Protocol Extensions for Path Computation Element (PCE) Discovery	[RFC5088]
IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery	[RFC5089]
Path Computation Element (PCE) Communication Protocol (PCEP)	[RFC5440]
Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism	[RFC5520]
Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)	[RFC5541]
Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization	[RFC5557]
Common YANG Data Types	[RFC6991]
YANG Data Model for Key Chains	[RFC8177]
Path Computation Element Communication Protocol (PCEP) Extensions for	[RFC8231]

Stateful PCE	
Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE	[RFC8232]
PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)	[RFC8253]
Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model	[RFC8281]
Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths	[RFC8306]
Network Configuration Access Control Model	[RFC8341]
Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages	[RFC8408]
Traffic Engineering Common YANG Types	[RFC8776]
A YANG Data Model for Traffic Engineering Tunnels and Interfaces	[I-D.ietf-teas-yang-te]
YANG Groupings for TLS Clients and TLS Servers	[I-D.ietf-netconf-tls-client-server]
PCEP Extensions for	[RFC8664]

Segment Routing	
PCEP Extensions for Establishing Relationships Between Sets of LSPs	[RFC8697]
YANG Data Model for OSPF Protocol	[I-D.ietf-ospf-yang]
YANG Data Model for IS-IS Protocol	[I-D.ietf-isis-yang-isis-cfg]
PCEP extensions for GMPLS	[RFC8779]
ASSOCIATION Type Field in Path Computation Element Protocol (PCEP) Numbers	[IANA-PCEP]
Path Computation Element (PCE) Capability Flags in Open Shortest Path First v2 (OSPFv2) Parameters	[IANA-OSPF]

Table 2: References in the YANG modules

4. Objectives

This section describes some of the design objectives for the model:

- * In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.
- * The data model should be suitable for new implementations to use as is.
- * Mapping to the PCEP MIB Module should be clear.
- * The data model should allow for static configurations of peers.

- * The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors. This could be in a separate model which augments the base data model.
- * It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

5.1. The Overview of PCEP Data Model

The PCEP YANG module defined in this document has all the common building blocks for the PCEP protocol.

```

module: ietf-pcep
  +--rw pcep!
    +--rw entity
      +--rw addr inet:ip-address-no-zone
      +--rw enabled? boolean
      +--rw role pcep-role
      +--rw description? string
      +--rw speaker-entity-id? string {sync-opt}?
      +--rw admin-status? boolean
      +--ro index? uint32
      +--ro oper-status? pcep-oper-status
      +--rw domains
        | +--rw domains* [domain-type domain-info]
        | +--...
      +--rw capability
        | +--...
      +--rw pce-info
        | +--rw scope
        | | +--...
        | +--rw neighbour-domains
        | | +--...
        | +--rw path-key {path-key}?
        | | +--...
        | +--...
      +--...
      +--ro lsp-db {stateful}?
        | +--ro db-ver? uint64 {sync-opt}?
        | +--ro association-list*
        | | [type id source global-source extended-id]
        | | {association}?
        | | +--...
        | +--ro lsp* [plsp-id pcc-id lsp-id]

```

```

    |      +--...
+---ro path-keys {path-key}?
    |      +---ro path-keys* [path-key]
    |      +--...
+---rw peers
    +---rw peer* [addr]
        +--...
        +---ro sessions
            +---ro session* [initiator]
            +--...

rpcs:
  +---x trigger-resync {stateful,sync-opt}?
    +---w input
    +---w pcc?    -> /pcep/entity/peers/peer/addr

notifications:
  +---n pcep-session-up
    |      +--...
  +---n pcep-session-down
    |      +--...
  +---n pcep-session-local-overload
    |      +--...
  +---n pcep-session-local-overload-clear
    |      +--...
  +---n pcep-session-peer-overload
    |      +--...
  +---n pcep-session-peer-overload-clear
    +--...

```

5.2. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using ietf-inet-types [RFC6991]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [RFC7420] uses an entity list and a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding pcePcepEntityIndex and only one entity is assumed.

The various information related to this entity such as its domain, capabilities etc. When the entity is PCE it could also have path-key and the LSP-DB information.


```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--rw addr inet:ip-address-no-zone
    +--rw enabled? boolean
    +--rw role pcep-role
    +--rw description? string
    +--rw speaker-entity-id? string {sync-opt}?
    +--rw admin-status? boolean
    +--ro index? uint32
    +--ro oper-status? pcep-oper-status
    +--rw domains
      +--rw domains* [domain-type domain-info]
        +--rw domain-type identityref
        +--rw domain-info domain
    +--rw capability
      +--rw capability? bits
      +--rw pce-initiated? boolean {pce-initiated}?
      +--rw include-db-ver? boolean {stateful,sync-opt}?
      +--rw trigger-resync? boolean {stateful,sync-opt}?
      +--rw trigger-initial-sync? boolean {stateful,sync-opt}?
      +--rw incremental-sync? boolean {stateful,sync-opt}?
      +--rw sr {sr}?
        +--rw enabled? boolean
        +--rw msd-limit? boolean
        +--rw nai? boolean
    +--rw msd? uint8 {sr}?
    +--rw pce-info
      +--rw scope
        +--rw path-scope? bits
        +--rw intra-area-pref? uint8
        +--rw inter-area-pref? uint8
        +--rw inter-as-pref? uint8
        +--rw inter-layer-pref? uint8
      +--rw neighbour-domains
        +--rw domains* [domain-type domain-info]
          +--rw domain-type identityref
          +--rw domain-info domain
      +--rw path-key {path-key}?
        +--rw enabled? boolean
        +--rw discard-timer? uint32
        +--rw reuse-time? uint32
        +--rw pce-id? inet:ip-address-no-zone
    +--rw connect-timer? uint16
    +--rw connect-max-retry? uint32
    +--rw init-back-off-timer? uint16
    +--rw max-back-off-timer? uint32
    +--ro open-wait-timer? uint16

```

```

+---ro keep-wait-timer?          uint16
+---rw keepalive-timer?         uint8
+---rw dead-timer?              uint8
+---rw allow-negotiation?       boolean
+---rw max-keepalive-timer?     uint8
+---rw max-dead-timer?          uint8
+---rw min-keepalive-timer?     uint8
+---rw min-dead-timer?          uint8
+---rw sync-timer?              uint16 {svec}?
+---rw request-timer?           uint16
+---rw max-sessions?            uint32
+---rw max-unknown-reqs?        uint32
+---rw max-unknown-msgs?        uint32
+---rw pcep-notification-max-rate uint32
+---rw stateful-parameter {stateful}?
|   +---rw state-timeout?        uint32
|   +---rw redelegation-timeout? uint32
|   +---rw rpt-non-pcep-lsp?     boolean
+---rw of-list {objective-function}?
|   +---rw objective-function* [of]
|       +---rw of identityref
+---ro lsp-db {stateful}?
|   +---ro db-ver?                uint64 {sync-opt}?
|   +---ro association-list*
|       [type id source global-source extended-id]
|       {association}?
|       +---ro type                identityref
|       +---ro id                  uint16
|       +---ro source               inet:ip-address-no-zone
|       +---ro global-source        uint32
|       +---ro extended-id          string
|       +---ro lsp* [plsp-id pcc-id lsp-id]
|           +---ro plsp-id         -> /pcep/entity/lsp-db/lsp/plsp-id
|           +---ro pcc-id          -> /pcep/entity/lsp-db/lsp/pcc-id
|           +---ro lsp-id          -> /pcep/entity/lsp-db/lsp/lsp-id
+---ro lsp* [plsp-id pcc-id lsp-id]
|   +---ro plsp-id                uint32
|   +---ro pcc-id                 inet:ip-address-no-zone
|   +---ro source?                inet:ip-address-no-zone
|   +---ro destination?           inet:ip-address-no-zone
|   +---ro tunnel-id?             uint16
|   +---ro lsp-id                 uint16
|   +---ro extended-tunnel-id?    inet:ip-address-no-zone
|   +---ro admin-state?           boolean
|   +---ro operational-state?     operational-state
|   +---ro delegated
|       +---ro enabled?           boolean
|       +---ro peer?              -> /pcep/entity/peers/peer/addr

```

```

|   +--ro srp-id?      uint32
+--ro initiation {pce-initiated}?
|   +--ro enabled?    boolean
|   +--ro peer?       -> /pcep/entity/peers/peer/addr
+--ro symbolic-path-name? string
+--ro last-error?     identityref
+--ro pst?            identityref
+--ro association-list*
|   [type id source global-source extended-id]
|   {association}?
+--ro type
|   -> /pcep/entity/lsp-db/association-list/type
+--ro id
|   -> /pcep/entity/lsp-db/association-list/id
+--ro source          leafref
+--ro global-source   leafref
+--ro extended-id     leafref
+--ro path-keys {path-key}?
|   +--ro path-keys* [path-key]
|   +--ro path-key    uint16
+--ro cps
|   +--ro explicit-route-objects* [index]
|   +--ro index        uint32
|   +--ro (type)?
|   |   +--:(numbered-node-hop)
|   |   |   +--ro numbered-node-hop
|   |   |   |   +--ro node-id      te-node-id
|   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   +--:(numbered-link-hop)
|   |   |   |   +--ro numbered-link-hop
|   |   |   |   |   +--ro link-tp-id   te-tp-id
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   |   |   +--ro direction?  te-link-direction
|   |   |   +--:(unnumbered-link-hop)
|   |   |   |   +--ro unnumbered-link-hop
|   |   |   |   |   +--ro link-tp-id   te-tp-id
|   |   |   |   |   +--ro node-id      te-node-id
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   |   |   +--ro direction?  te-link-direction
|   |   |   +--:(as-number)
|   |   |   |   +--ro as-number-hop
|   |   |   |   |   +--ro as-number   inet:as-number
|   |   |   |   |   +--ro hop-type?   te-hop-type
|   |   |   +--:(label)
|   |   |   |   +--ro label-hop
|   |   |   |   |   +--ro te-label
|   |   |   |   |   ...
+--ro pcc-original?    -> /pcep/entity/peers/peer/addr

```

```

    +--ro req-id?          uint32
    +--ro retrieved?       boolean
    +--ro pcc-retrieved?   -> /pcep/entity/peers/peer/addr
    +--ro creation-time?   yang:timestamp
    +--ro discard-time?    uint32
    +--ro reuse-time?      uint32
+--rw peers
  +--rw peer* [addr]
  +--...

```

5.3. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address.

There is a list for static peer configuration and operational state of all peers (i.e. static as well as discovered) ("/pcep/entity/peers"). The list is used to enable remote PCE configuration at PCC (or PCE) and has the operational state of these peers as well as the remote PCE peer which were discovered and PCC peers that have initiated session.

```

module: ietf-pcep
+--rw pcep!
  +--rw entity
  +--...
  +--rw peers
    +--rw peer* [addr]
      +--rw addr          inet:ip-address-no-zone
      +--rw role          pcep-role
      +--rw description?  string

```

```

+--rw domains
|   +--rw domains* [domain-type domain-info]
|       +--rw domain-type      identityref
|       +--rw domain-info      domain
+--rw capability
|   +--rw capability?           bits
|   +--rw pce-initiated?        boolean
|       | {pce-initiated}?
|   +--rw include-db-ver?       boolean
|       | {stateful, sync-opt}?
|   +--rw trigger-resync?       boolean
|       | {stateful, sync-opt}?
|   +--rw trigger-initial-sync? boolean
|       | {stateful, sync-opt}?
|   +--rw incremental-sync?     boolean
|       | {stateful, sync-opt}?
+--rw sr {sr}?
|   +--rw enabled?             boolean
|   +--rw msd-limit?          boolean
|   +--rw nai?                 boolean
+--rw msd?                      uint8 {sr}?
+--rw pce-info
|   +--rw scope
|       +--rw path-scope?       bits
|       +--rw intra-area-pref?  uint8
|       +--rw inter-area-pref?  uint8
|       +--rw inter-as-pref?    uint8
|       +--rw inter-layer-pref? uint8
|   +--rw neighbour-domains
|       +--rw domains* [domain-type domain-info]
|           +--rw domain-type      identityref
|           +--rw domain-info      domain
+--rw delegation-pref?          uint8 {stateful}?
+--rw auth
|   +--rw (auth-type-selection)?
|       +--:(auth-key-chain)
|           +--rw key-chain?
|               key-chain:key-chain-ref
|       +--:(auth-key)
|           +--rw crypto-algorithm          identityref
|           +--rw (key-string-style)?
|               +--:(keystring)
|                   +--rw keystring?         string
|               +--:(hexadecimal)
|                   {key-chain:hex-key-string}?
|                   +--rw hexadecimal-string?
|                       yang:hex-string
|       +--:(auth-tls) {tls}?

```

```

    |         +--rw (role)?
    |         |   +--:(server)
    |         |   |   +--rw tls-server
    |         |   |   ...
    |         |   +--:(client)
    |         |   |   +--rw tls-client
    |         |   |   ...
+--ro discontinuity-time?      yang:timestamp
+--ro initiate-session?       boolean
+--ro session-exists?         boolean
+--ro session-up-time?        yang:timestamp
+--ro session-fail-time?      yang:timestamp
+--ro session-fail-up-time?    yang:timestamp
+--ro sessions
    +--ro session* [initiator]
    +--...

```

5.4. The Session Lists

The session list contains PCEP sessions that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. When one of these sessions reaches the active state, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

```

module: ietf-pcep
  +--rw pcep!
    +--rw entity
      +--...
      +--rw peers
        +--rw peer* [addr]
          +--...
          +--ro sessions
            +--ro session* [initiator]
              +--ro initiator                pcep-initiator
              +--ro role?
              |   -> /pcep/entity/role
              +--ro state-last-change?      yang:timestamp
              +--ro state?
              |   pcep-sess-state
              +--ro session-creation?      yang:timestamp
              +--ro connect-retry?         yang:counter32
              +--ro local-id?              uint8
              +--ro remote-id?             uint8
              +--ro keepalive-timer?       uint8
              +--ro peer-keepalive-timer?  uint8
              +--ro dead-timer?            uint8
              +--ro peer-dead-timer?       uint8
              +--ro ka-hold-time-rem?      uint8
              +--ro overloaded?            boolean
              +--ro overloaded-timestamp?  yang:timestamp
              +--ro overload-time?         uint32
              +--ro peer-overloaded?       boolean
              +--ro peer-overloaded-timestamp? yang:timestamp
              +--ro peer-overload-time?    uint32
              +--ro lspdb-sync?            sync-state
              |   {stateful}?
              +--ro recv-db-ver?           uint64
              |   {stateful, sync-opt}?
              +--ro of-list {objective-function}?
              |   +--ro objective-function* [of]
              |   |   +--ro of identityref
              +--ro pst-list
              |   +--ro path-setup-type* [pst]
              |   |   +--ro pst identityref
              +--ro assoc-type-list {association}?
              |   +--ro assoc-type* [at]
              |   |   +--ro at identityref
              +--ro speaker-entity-id?     string
              |   {sync-opt}?

```

5.5. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB [RFC7420] notifications.

notifications:

```

+---n pcep-session-up
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?
|   |       -> /pcep/entity/peers/peer/sessions/session/initiator
|   +---ro state-last-change?  yang:timestamp
|   +---ro state?              pcep-sess-state
+---n pcep-session-down
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?  pcep-initiator
|   +---ro state-last-change?  yang:timestamp
|   +---ro state?              pcep-sess-state
+---n pcep-session-local-overload
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?
|   |       -> /pcep/entity/peers/peer/sessions/session/initiator
|   +---ro overloaded?         boolean
|   +---ro overloaded-timestamp? yang:timestamp
|   +---ro overload-time?      uint32
+---n pcep-session-local-overload-clear
|   +---ro peer-addr?
|   |       -> /pcep/entity/peers/peer/addr
|   +---ro overloaded?         boolean
|   +---ro overloaded-clear-timestamp? yang:timestamp
+---n pcep-session-peer-overload
|   +---ro peer-addr?
|   |       -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?
|   |       -> /pcep/entity/peers/peer/sessions/session/initiator
|   +---ro peer-overloaded?     boolean
|   +---ro peer-overloaded-timestamp? yang:timestamp
|   +---ro peer-overload-time?  uint32
+---n pcep-session-peer-overload-clear
|   +---ro peer-addr?
|   |       -> /pcep/entity/peers/peer/addr
|   +---ro peer-overloaded?     boolean
|   +---ro peer-overloaded-clear-timestamp? yang:timestamp

```

5.6. RPC

This YANG model defines a RPC to trigger state resynchronization to a particular PCEP peer.


```

rpcs:
  +---x trigger-resync {stateful,sync-opt}?
    +---w input
      +---w pcc?   -> /pcep/entity/peers/peer/addr

```

6. The Design of PCEP Statistics Data Model

The module, "ietf-pcep-stats", augments the ietf-pcep module to include statistics at the PCEP peer and session level. It includes a RPC to reset statistics.

```
module: ietf-pcep-stats
```

```

augment /pcep:pcep/pcep:entity/pcep:peers/pcep:peer:
  +--ro num-sess-setup-ok?      yang:counter32
  +--ro num-sess-setup-fail?    yang:counter32
  +--ro pcep-stats
    +--ro discontinuity-time?    yang:timestamp
    +--ro rsp-time-avg?          uint32
    +--ro rsp-time-lwm?          uint32
    +--ro rsp-time-hwm?          uint32
    +--ro num-pcreq-sent?         yang:counter32
    +--ro num-pcreq-rcvd?         yang:counter32
    +--ro num-pcrep-sent?         yang:counter32
    +--ro num-pcrep-rcvd?         yang:counter32
    +--ro num-pcerr-sent?         yang:counter32
    +--ro num-pcerr-rcvd?         yang:counter32
    +--ro num-pcntf-sent?         yang:counter32
    +--ro num-pcntf-rcvd?         yang:counter32
    +--ro num-keepalive-sent?     yang:counter32
    +--ro num-keepalive-rcvd?     yang:counter32
    +--ro num-unknown-rcvd?       yang:counter32
    +--ro num-corrupt-rcvd?       yang:counter32
    +--ro num-req-sent?           yang:counter32
    +--ro num-req-sent-pend-rep?   yang:counter32
    +--ro num-req-sent-ero-rcvd?   yang:counter32
    +--ro num-req-sent-nopath-rcvd? yang:counter32
    +--ro num-req-sent-cancel-rcvd? yang:counter32
    +--ro num-req-sent-error-rcvd? yang:counter32
    +--ro num-req-sent-timeout?    yang:counter32
    +--ro num-req-sent-cancel-sent? yang:counter32
    +--ro num-req-rcvd?           yang:counter32
    +--ro num-req-rcvd-pend-rep?   yang:counter32
    +--ro num-req-rcvd-ero-sent?   yang:counter32
    +--ro num-req-rcvd-nopath-sent? yang:counter32
    +--ro num-req-rcvd-cancel-sent? yang:counter32
    +--ro num-req-rcvd-error-sent? yang:counter32
    +--ro num-req-rcvd-cancel-rcvd? yang:counter32

```

```

+--ro num-rep-rcvd-unknown?          yang:counter32
+--ro num-req-rcvd-unknown?          yang:counter32
+--ro svec {pcep:svec}?
|   +--ro num-svec-sent?              yang:counter32
|   +--ro num-svec-req-sent?          yang:counter32
|   +--ro num-svec-rcvd?              yang:counter32
|   +--ro num-svec-req-rcvd?          yang:counter32
+--ro stateful {pcep:stateful}?
|   +--ro num-pcrpt-sent?              yang:counter32
|   +--ro num-pcrpt-rcvd?              yang:counter32
|   +--ro num-pcupd-sent?              yang:counter32
|   +--ro num-pcupd-rcvd?              yang:counter32
|   +--ro num-rpt-sent?                yang:counter32
|   +--ro num-rpt-rcvd?                yang:counter32
|   +--ro num-rpt-rcvd-error-sent?     yang:counter32
|   +--ro num-upd-sent?                yang:counter32
|   +--ro num-upd-rcvd?                yang:counter32
|   +--ro num-upd-rcvd-unknown?        yang:counter32
|   +--ro num-upd-rcvd-undelagated?    yang:counter32
|   +--ro num-upd-rcvd-error-sent?     yang:counter32
|   +--ro initiation {pcep:pce-initiated}?
|       +--ro num-pcinitiate-sent?      yang:counter32
|       +--ro num-pcinitiate-rcvd?      yang:counter32
|       +--ro num-initiate-sent?        yang:counter32
|       +--ro num-initiate-rcvd?        yang:counter32
|       +--ro num-initiate-rcvd-error-sent? yang:counter32
+--ro path-key {pcep:path-key}?
|   +--ro num-unknown-path-key?        yang:counter32
|   +--ro num-exp-path-key?            yang:counter32
|   +--ro num-dup-path-key?            yang:counter32
|   +--ro num-path-key-no-attempt?     yang:counter32
+--ro num-req-sent-closed?             yang:counter32
+--ro num-req-rcvd-closed?             yang:counter32
augment /pcep:pcep/pcep:entity/pcep:peers/pcep:peer/pcep:sessions
/pcep:session:
+--ro pcep-stats
|   +--ro discontinuity-time?           yang:timestamp
|   +--ro rsp-time-avg?                 uint32
|   +--ro rsp-time-lwm?                 uint32
|   +--ro rsp-time-hwm?                 uint32
|   +--ro num-pcreq-sent?                yang:counter32
|   +--ro num-pcreq-rcvd?                yang:counter32
|   +--ro num-pcrep-sent?                yang:counter32
|   +--ro num-pcrep-rcvd?                yang:counter32
|   +--ro num-pcerr-sent?                yang:counter32
|   +--ro num-pcerr-rcvd?                yang:counter32
|   +--ro num-pcntf-sent?                yang:counter32
|   +--ro num-pcntf-rcvd?                yang:counter32

```

```
+--ro num-keepalive-sent?          yang:counter32
+--ro num-keepalive-rcvd?          yang:counter32
+--ro num-unknown-rcvd?            yang:counter32
+--ro num-corrupt-rcvd?            yang:counter32
+--ro num-req-sent?                yang:counter32
+--ro num-req-sent-pend-rep?        yang:counter32
+--ro num-req-sent-ero-rcvd?        yang:counter32
+--ro num-req-sent-nopath-rcvd?     yang:counter32
+--ro num-req-sent-cancel-rcvd?     yang:counter32
+--ro num-req-sent-error-rcvd?      yang:counter32
+--ro num-req-sent-timeout?         yang:counter32
+--ro num-req-sent-cancel-sent?     yang:counter32
+--ro num-req-rcvd?                yang:counter32
+--ro num-req-rcvd-pend-rep?        yang:counter32
+--ro num-req-rcvd-ero-sent?        yang:counter32
+--ro num-req-rcvd-nopath-sent?     yang:counter32
+--ro num-req-rcvd-cancel-sent?     yang:counter32
+--ro num-req-rcvd-error-sent?      yang:counter32
+--ro num-req-rcvd-cancel-rcvd?     yang:counter32
+--ro num-rep-rcvd-unknown?         yang:counter32
+--ro num-req-rcvd-unknown?         yang:counter32
+--ro svec {pcep:svec}?
|   +--ro num-svec-sent?            yang:counter32
|   +--ro num-svec-req-sent?        yang:counter32
|   +--ro num-svec-rcvd?            yang:counter32
|   +--ro num-svec-req-rcvd?        yang:counter32
+--ro stateful {pcep:stateful}?
|   +--ro num-pcrpt-sent?            yang:counter32
|   +--ro num-pcrpt-rcvd?            yang:counter32
|   +--ro num-pcupd-sent?            yang:counter32
|   +--ro num-pcupd-rcvd?            yang:counter32
|   +--ro num-rpt-sent?              yang:counter32
|   +--ro num-rpt-rcvd?              yang:counter32
|   +--ro num-rpt-rcvd-error-sent?   yang:counter32
|   +--ro num-upd-sent?              yang:counter32
|   +--ro num-upd-rcvd?              yang:counter32
|   +--ro num-upd-rcvd-unknown?      yang:counter32
|   +--ro num-upd-rcvd-undelegated?  yang:counter32
|   +--ro num-upd-rcvd-error-sent?   yang:counter32
|   +--ro initiation {pcep:pce-initiated}?
|       +--ro num-pcinitiate-sent?    yang:counter32
|       +--ro num-pcinitiate-rcvd?    yang:counter32
|       +--ro num-initiate-sent?      yang:counter32
|       +--ro num-initiate-rcvd?      yang:counter32
|       +--ro num-initiate-rcvd-error-sent? yang:counter32
+--ro path-key {pcep:path-key}?
|   +--ro num-unknown-path-key?      yang:counter32
|   +--ro num-exp-path-key?          yang:counter32
```

```

      +---ro num-dup-path-key?          yang:counter32
      +---ro num-path-key-no-attempt?   yang:counter32

  rpcs:
    +---x statistics-reset
      +---w input
        +---w (peer-or-all)?
          +--:(peer)
            +---w peer-addr?
              -> /pcep:pcep/entity/peers/peer/addr
          +--:(all)

```

7. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

This document further handles advanced PCE features like -

- * Capability and Scope
- * Domain information (local/neighbour)
- * Path-Key
- * Objective Function (OF)
- * Global Concurrent Optimization (GCO)
- * P2MP
- * GMPLS
- * Inter-Layer
- * Stateful PCE
- * Segment Routing (SR)
- * Authentication including PCEPS (TLS)

7.1. Stateful PCE's LSP-DB

In the operational datastore of stateful PCE, the list of LSP state are maintained in the LSP-DB. The key is the PLSP-ID and the PCC's IP address.

The PCEP data model contains the operational state of LSPs (/pcep/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [I-D.ietf-teas-yang-te]. A reference to LSP state in TE model is maintained.

8. Other Considerations

8.1. PCEP over TLS (PCEPS)

[RFC8253] describe the use of TLS v1.2 [RFC5246] or later in PCEP. The peer acting as the PCEP client must act as the TLS client. The TLS client actively opens the TLS connection and the TLS server passively listens for the incoming TLS connections. The well-known TCP port number 4189 is used by PCEP servers to listen for TCP connections established by PCEP over TLS clients. The TLS client sends the TLS ClientHello message to begin the TLS handshake. The TLS server sends a CertificateRequest in order to request a certificate from the TLS client. Once the TLS handshake has finished, the client and the server begin to exchange PCEP messages. Client and server identity verification is done before the PCEP open message is sent. This means that the identity verification is completed before the PCEP session is started.

Note that, a PCEP speaker could act as both a client (PCC) and a server (PCE). The role within the context of a PCEP session is determined by the relationship it has with its peer (the same holds good for TLS as well).

9. PCEP YANG Modules

9.1. ietf-pcep module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2022-01-25.yang"
module ietf-pcep {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
  prefix pcep;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
}
```

```
import ietf-yang-types {
  prefix yang;
  reference
    "RFC 6991: Common YANG Data Types";
}
import ietf-te-types {
  prefix te-types;
  reference
    "RFC 8776: Common YANG Data Types for Traffic Engineering";
}
import ietf-key-chain {
  prefix key-chain;
  reference
    "RFC 8177: YANG Data Model for Key Chains";
}
import ietf-netconf-acm {
  prefix nacm;
  reference
    "RFC 8341: Network Configuration Protocol (NETCONF) Access
      Control Model";
}
import ietf-tls-server {
  prefix tlss;
  reference
    "I-D.ietf-netconf-tls-client-server: YANG Groupings for TLS
      Clients and TLS Servers";
}
import ietf-tls-client {
  prefix tlsc;
  reference
    "I-D.ietf-netconf-tls-client-server: YANG Groupings for TLS
      Clients and TLS Servers";
}
import ietf-ospf {
  prefix ospf;
  reference
    "I-D.ietf-ospf-yang: YANG Data Model for OSPF Protocol";
}
import ietf-isis {
  prefix isis;
  reference
    "I-D.ietf-isis-yang-isis-cfg: YANG Data Model for IS-IS
      Protocol";
}

organization
  "IETF PCE (Path Computation Element) Working Group";
contact
```

```
"WG Web:  <https://datatracker.ietf.org/wg/pce/about/>
WG List:  <mailto:pce@ietf.org>
Editor:   Dhruv Dhody
          <mailto:dhruv.ietf@gmail.com>";
description
"The YANG module defines a generic configuration and
operational model for PCEP.

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
'MAY', and 'OPTIONAL' in this document are to be interpreted as
described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
they appear in all capitals, as shown here.

Copyright (c) 2022 IETF Trust and the persons identified as
authors of the code.  All rights reserved.

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject
to the license terms contained in, the Revised BSD License
set forth in Section 4.c of the IETF Trust's Legal Provisions
Relating to IETF Documents
(https://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";
```

```
revision 2022-01-25 {
  description
    "Initial revision.";
  reference
    "RFC XXXX:  A YANG Data Model for Path Computation
    Element Communications Protocol (PCEP)";
}

/*
 * Typedefs
 */

typedef pcep-role {
  type enumeration {
    enum unknown {
      value 0;
      description
        "An unknown role";
    }
    enum pcc {
      value 1;
    }
  }
}
```

```
        description
            "The role of a Path Computation Client";
    }
    enum pce {
        value 2;
        description
            "The role of Path Computation Element";
    }
    enum pcc-and-pce {
        value 3;
        description
            "The role of both Path Computation Client and
            Path Computation Element";
    }
}
description
    "The role of a PCEP speaker.
    Takes one of the following values
    - unknown(0): the role is not known.
    - pcc(1): the role is of a Path Computation
      Client (PCC).
    - pce(2): the role is of a Path Computation
      Server (PCE).
    - pcc-and-pce(3): the role is of both a PCC and
      a PCE.";
reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

typedef pcep-oper-status {
    type enumeration {
        enum oper-status-up {
            value 1;
            description
                "The PCEP entity is active";
        }
        enum oper-status-down {
            value 2;
            description
                "The PCEP entity is inactive";
        }
        enum oper-status-going-up {
            value 3;
            description
                "The PCEP entity is activating";
        }
        enum oper-status-going-down {
```



```
        value 4;
        description
            "The PCEP entity is deactivating";
    }
    enum oper-status-failed {
        value 5;
        description
            "The PCEP entity has failed and will recover
            when possible.";
    }
    enum oper-status-failed-perm {
        value 6;
        description
            "The PCEP entity has failed and will not recover
            without operator intervention";
    }
}
description
    "The operational status of the PCEP entity.
    Takes one of the following values
    - oper-status-up(1): Active
    - oper-status-down(2): Inactive
    - oper-status-going-up(3): Activating
    - oper-status-going-down(4): Deactivating
    - oper-status-failed(5): Failed
    - oper-status-failed-perm(6): Failed Permanantly";
reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

typedef pcep-initiator {
    type enumeration {
        enum local {
            value 1;
            description
                "The local PCEP entity initiated the session";
        }
        enum remote {
            value 2;
            description
                "The remote PCEP peer initiated the session";
        }
    }
}
description
    "The initiator of the session, that is, whether the TCP
    connection was initiated by the local PCEP entity or
    the remote peer.
```

```
    Takes one of the following values
    - local(1): Initiated locally
    - remote(2): Initiated remotely";
}

typedef pcep-sess-state {
  type enumeration {
    enum tcp-pending {
      value 1;
      description
        "The TCPPending state of PCEP session.";
    }
    enum open-wait {
      value 2;
      description
        "The OpenWait state of PCEP session.";
    }
    enum keep-wait {
      value 3;
      description
        "The KeepWait state of PCEP session.";
    }
    enum session-up {
      value 4;
      description
        "The SessionUP state of PCEP session.";
    }
  }
  description
    "The current state of the session.
    The set of possible states excludes the idle state
    since entries do not exist in the idle state.
    Takes one of the following values
    - tcp-pending(1): PCEP TCPPending state
    - open-wait(2): PCEP OpenWait state
    - keep-wait(3): PCEP KeepWait state
    - session-up(4): PCEP SessionUP state";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

typedef domain-ospf-area {
  type ospf:area-id-type;
  description
    "OSPF Area ID.";
  reference
    "I-D.ietf-ospf-yang: YANG Data Model for OSPF Protocol";
}
```

```
}

typedef domain-isis-area {
  type isis:area-address;
  description
    "IS-IS Area ID.";
  reference
    "I-D.ietf-isis-yang-isis-cfg: YANG Data Model for IS-IS
    Protocol";
}

typedef domain-as {
  type inet:as-number;
  description
    "Autonomous System number.";
}

typedef domain {
  type union {
    type domain-ospf-area;
    type domain-isis-area;
    type domain-as;
  }
  description
    "The Domain Information";
}

typedef operational-state {
  type enumeration {
    enum down {
      value 0;
      description
        "not active.";
    }
    enum up {
      value 1;
      description
        "signalled.";
    }
    enum active {
      value 2;
      description
        "up and carrying traffic.";
    }
    enum going-down {
      value 3;
      description
        "LSP is being torn down, resources are
```

```
        being released.";
    }
    enum going-up {
        value 4;
        description
            "LSP is being signalled.";
    }
}
description
    "The operational status of the LSP";
reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

typedef sync-state {
    type enumeration {
        enum pending {
            value 0;
            description
                "The state synchronization
                has not started.";
        }
        enum ongoing {
            value 1;
            description
                "The state synchronization
                is ongoing.";
        }
        enum finished {
            value 2;
            description
                "The state synchronization
                is finished.";
        }
    }
}
description
    "The LSP-DB state synchronization operational
    status.";
reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

/*
 * Features
 */
```

```
feature svec {
  description
    "Support synchronized path computation.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

feature gmpls {
  description
    "Support GMPLS.";
  reference
    "RFC 8779: PCEP extensions for GMPLS";
}

feature objective-function {
  description
    "Support OF as per RFC 5541.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP)";
}

feature global-concurrent {
  description
    "Support Global Concurrent Optimization (GCO) as per RFC
    5557.";
  reference
    "RFC 5557: Path Computation Element Communication Protocol
    (PCEP) Requirements and Protocol Extensions in Support of
    Global Concurrent Optimization";
}

feature path-key {
  description
    "Support path-key as per RFC 5520.";
  reference
    "RFC 5520: Preserving Topology Confidentiality in Inter-
    Domain Path Computation Using a Path-Key-Based Mechanism";
}

feature p2mp {
  description
    "Support Point-to-Multipoint (P2MP) as per RFC 8306.";
  reference
    "RFC 8306: Extensions to the Path Computation Element
    Communication Protocol (PCEP) for Point-to-Multipoint
    Traffic Engineering Label Switched Paths";
}
```

```
}

feature stateful {
  description
    "Support Stateful PCE as per RFC 8231.";
  reference
    "RFC 8231: Path Computation Element Communication Protocol
    (PCEP) Extensions for Stateful PCE";
}

feature sync-opt {
  description
    "Support Stateful state synchronization optimization
    as per RFC 8232";
  reference
    "RFC 8232: Optimizations of Label Switched Path State
    Synchronization Procedures for a Stateful PCE";
}

feature pce-initiated {
  description
    "Support PCE-Initiated LSP as per
    RFC 8281.";
  reference
    "RFC 8281: Path Computation Element Communication Protocol
    (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful
    PCE Model";
}

feature tls {
  description
    "Support PCEP over TLS as per RFC 8253.";
  reference
    "RFC 8253: PCEPS: Usage of TLS to Provide a Secure Transport
    for the Path Computation Element Communication Protocol
    (PCEP)";
}

feature sr {
  description
    "Support Segment Routing (SR) for PCE.";
  reference
    "RFC 8664: Path Computation Element Communication Protocol
    (PCEP) Extensions for Segment Routing";
}

feature association {
  description
```

```
    "Support Association in PCEP.";
  reference
    "RFC 8697: Path Computation Element Communication Protocol
      (PCEP) Extensions for Establishing Relationships between
      Sets of Label Switched Paths (LSPs)";
}

/*
 * Identities
 */

identity domain-type {
  description
    "Base Domain Type for PCE";
}

identity ospf-area {
  base domain-type;
  description
    "The OSPF area.";
}

identity isis-area {
  base domain-type;
  description
    "The IS-IS area.";
}

identity autonomous-system {
  base domain-type;
  description
    "The Autonomous System (AS).";
}

identity lsp-error {
  if-feature "stateful";
  description
    "Base LSP error";
  reference
    "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
}

identity no-error-lsp-error {
  if-feature "stateful";
  base lsp-error;
  description
    "No error, LSP is fine.";
```

```
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity unknown-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "Unknown reason.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity limit-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "Limit reached for PCE-controlled LSPs.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity pending-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "Too many pending LSP update requests.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity unacceptable-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "Unacceptable parameters.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity internal-lsp-error {
    if-feature "stateful";
    base lsp-error;
```



```
    description
      "Internal error.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity admin-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "LSP administratively brought down.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity preempted-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "LSP preempted.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  identity rsvp-lsp-error {
    if-feature "stateful";
    base lsp-error;
    description
      "RSVP signaling error.";
    reference
      "RFC 8231: Path Computation Element Communication Protocol
      (PCEP) Extensions for Stateful PCE";
  }

  /*
   * Groupings
   */

  grouping pce-scope {
    description
      "This grouping defines PCE path computation scope
      information which maybe relevant to PCE selection.
      This information corresponds to PCE auto-discovery
      information.";
    reference
```

```
"RFC 5088: OSPF Protocol Extensions for Path
Computation Element (PCE) Discovery
RFC 5089: IS-IS Protocol Extensions for Path
Computation Element (PCE) Discovery";
leaf path-scope {
  type bits {
    bit intra-area-scope {
      description
        "PCE can compute intra-area paths (L bit).";
    }
    bit inter-area-scope {
      description
        "PCE can compute inter-area paths (R bit).";
    }
    bit inter-area-scope-default {
      description
        "PCE can act as a default PCE for inter-area
        path computation. (Rd bit)";
    }
    bit inter-as-scope {
      description
        "PCE can compute inter-AS paths (S bit).";
    }
    bit inter-as-scope-default {
      description
        "PCE can act as a default PCE for inter-AS
        path computation (Sd bit).";
    }
    bit inter-layer-scope {
      description
        "PCE can compute inter-layer paths (Y bit).";
    }
  }
  description
    "The field corresponding to the path scope bits";
}
leaf intra-area-pref {
  type uint8 {
    range "0..7";
  }
  description
    "The PCE's preference for intra-area TE LSP
    computation (PrefL field). Where 7 reflects
    the highest preference.";
}
leaf inter-area-pref {
  type uint8 {
    range "0..7";
```

```
    }
    description
      "The PCE's preference for inter-area TE LSP
       computation (PrefR field). Where 7 reflects
       the highest preference.";
  }
  leaf inter-as-pref {
    type uint8 {
      range "0..7";
    }
    description
      "The PCE's preference for inter-AS TE LSP
       computation (PrefS field). Where 7 reflects
       the highest preference.";
  }
  leaf inter-layer-pref {
    type uint8 {
      range "0..7";
    }
    description
      "The PCE's preference for inter-layer TE LSP
       computation (PrefY field). Where 7 reflects
       the highest preference.";
  }
}

//pce-scope

grouping domain {
  description
    "This grouping specifies a Domain where the
     PCEP speaker has topology visibility.";
  leaf domain-type {
    type identityref {
      base domain-type;
    }
    description
      "The domain type.";
  }
  leaf domain-info {
    type domain;
    description
      "The domain Information.";
  }
}

//domain
```

```
grouping info {
  description
    "This grouping specifies all information which
    maybe relevant to both PCC and PCE.
    This information corresponds to PCE auto-discovery
    information.";
  container domains {
    description
      "The local domain for the PCEP entity";
    list domains {
      key "domain-type domain-info";
      description
        "The local domain.";
      uses domain {
        description
          "The local domain for the PCEP entity.";
      }
    }
  }
}
container capability {
  description
    "The PCEP entity capability information of local
    PCEP entity. This maybe relevant to PCE selection
    as well. This information corresponds to PCE auto-
    discovery information.";
  reference
    "IANA OSPF: Path Computation Element (PCE) Capability
    Flags in Open Shortest Path First v2 (OSPFv2)
    Parameters
    RFC 5088: OSPF Protocol Extensions for Path
    Computation Element (PCE) Discovery
    RFC 5089: IS-IS Protocol Extensions for Path
    Computation Element (PCE) Discovery";
  leaf capability {
    type bits {
      bit gmpls {
        if-feature "gmpls";
        description
          "Path computation with GMPLS link
          constraints.";
      }
      bit bi-dir {
        description
          "Bidirectional path computation.";
      }
      bit diverse {
        description
          "Diverse path computation.";
      }
    }
  }
}
```

```
    }
    bit load-balance {
        description
            "Load-balanced path computation.";
    }
    bit synchronize {
        if-feature "svec";
        description
            "Synchronized paths computation.";
    }
    bit objective-function {
        if-feature "objective-function";
        description
            "Support for multiple objective functions.";
    }
    bit add-path-constraint {
        description
            "Support for additive path constraints (max
            hop count, etc.).";
    }
    bit prioritization {
        description
            "Support for request prioritization.";
    }
    bit multi-request {
        description
            "Support for multiple requests per message.";
    }
    bit global-concurrent {
        if-feature "global-concurrent";
        description
            "Support for Global Concurrent Optimization
            (GCO).";
    }
    bit p2mp {
        if-feature "p2mp";
        description
            "Support for P2MP path computation.";
    }
    bit active {
        if-feature "stateful";
        description
            "Support for active stateful PCE.";
    }
    bit passive {
        if-feature "stateful";
        description
            "Support for passive stateful PCE.";
```

```
    }
    bit p2mp-active {
        if-feature "stateful";
        if-feature "p2mp";
        description
            "Support for active stateful PCE for P2MP.";
    }
    bit p2mp-passive {
        if-feature "stateful";
        if-feature "p2mp";
        description
            "Support for passive stateful PCE for P2MP.";
    }
    bit p2mp-pce-initiated {
        if-feature "stateful";
        if-feature "pce-initiated";
        if-feature "p2mp";
        description
            "Support for PCE-initiated LSP for P2MP.";
    }
}
description
    "The bits string indicating the capabilities";
reference
    "IANA OSPF: Path Computation Element (PCE) Capability
    Flags in Open Shortest Path First v2 (OSPFv2)
    Parameters
    RFC 5088: OSPF Protocol Extensions for Path
    Computation Element (PCE) Discovery
    RFC 5089: IS-IS Protocol Extensions for Path
    Computation Element (PCE) Discovery";
}
leaf pce-initiated {
    if-feature "pce-initiated";
    type boolean;
    description
        "Set to true if PCE-initiated LSP capability is
        enabled.";
    reference
        "RFC 8281: Path Computation Element Communication
        Protocol (PCEP) Extensions for PCE-Initiated LSP
        Setup in a Stateful PCE Model";
}
leaf include-db-ver {
    if-feature "stateful";
    if-feature "sync-opt";
    type boolean;
    description
```

```
        "Support inclusion of LSP-DB-VERSION
        in LSP object";
    reference
        "RFC 8232: Optimizations of Label Switched Path State
        Synchronization Procedures for a Stateful PCE";
}
leaf trigger-resync {
    if-feature "stateful";
    if-feature "sync-opt";
    type boolean;
    description
        "Support PCE triggered re-synchronization";
    reference
        "RFC 8232: Optimizations of Label Switched Path State
        Synchronization Procedures for a Stateful PCE";
}
leaf trigger-initial-sync {
    if-feature "stateful";
    if-feature "sync-opt";
    type boolean;
    description
        "PCE triggered initial synchronization";
    reference
        "RFC 8232: Optimizations of Label Switched Path State
        Synchronization Procedures for a Stateful PCE";
}
leaf incremental-sync {
    if-feature "stateful";
    if-feature "sync-opt";
    type boolean;
    description
        "Support incremental (delta) sync";
    reference
        "RFC 8232: Optimizations of Label Switched Path State
        Synchronization Procedures for a Stateful PCE";
}
container sr {
    if-feature "sr";
    description
        "If segment routing is supported";
    reference
        "RFC 8664: Path Computation Element Communication Protocol
        (PCEP) Extensions for Segment Routing";
    leaf enabled {
        type boolean;
        description
            "Set to true if SR is enabled";
    }
}
```

```
    leaf msd-limit {
      type boolean;
      default "false";
      description
        "True indicates no limit on MSD, the
         leaf msd is ignored";
    }
    leaf nai {
      type boolean;
      default "false";
      description
        "True indicates capability to resolve Node or
         Adjacency Identifier (NAI) to Segment
         Identifier (SID)";
    }
  }
  //sr
}
//capability
leaf msd {
  if-feature "sr";
  type uint8;
  description
    "Maximum SID Depth for SR";
  reference
    "RFC 8664: Path Computation Element Communication Protocol
     (PCEP) Extensions for Segment Routing";
}
}

//info

grouping pce-info {
  description
    "This grouping specifies all PCE information
     which maybe relevant to the PCE selection.
     This information corresponds to PCE auto-discovery
     information.";
  container scope {
    description
      "The path computation scope";
    uses pce-scope;
  }
  container neighbour-domains {
    description
      "The list of neighbour PCE-Domain
       toward which a PCE can compute
       paths";
  }
}
```



```
    list domains {
      key "domain-type domain-info";
      description
        "The neighbour domain.";
      uses domain {
        description
          "The PCE neighbour domain.";
      }
    }
  }
}

//pce-info

grouping notification-instance-hdr {
  description
    "This group describes common instance specific data
    for notifications.";
  leaf peer-addr {
    type leafref {
      path "/pcep/entity/peers/peer/addr";
    }
    description
      "Reference to peer address";
  }
}

// notification-instance-hdr

grouping notification-session-hdr {
  description
    "This group describes common session instance specific
    data for notifications.";
  leaf session-initiator {
    type leafref {
      path "/pcep/entity/peers/peer/sessions/"
        + "session/initiator";
    }
    description
      "Reference to pcep session initiator leaf";
  }
}

// notification-session-hdr

grouping of-list {
  description
    "List of Objective Functions (OF)";
```

```
reference
  "RFC 5541: Encoding of Objective Functions in the Path
  Computation Element Communication Protocol (PCEP)";
list objective-function {
  key "of";
  description
    "The list of authorized OF";
  leaf of {
    type identityref {
      base te-types:objective-function-type;
    }
    description
      "The OF authorized";
  }
}

/*
 * Configuration data nodes
 */

container pcep {
  presence "The PCEP is enabled";
  description
    "Parameters for list of configured PCEP entities
    on the device.";
  container entity {
    description
      "The configured PCEP entity on the device.";
    leaf addr {
      type inet:ip-address-no-zone;
      mandatory true;
      description
        "The local Internet address of this PCEP entity.
        If operating as a PCE server, the PCEP entity
        listens on this address. If operating as a PCC,
        the PCEP entity binds outgoing TCP connections
        to this address. It is possible for the PCEP entity
        to operate both as a PCC and a PCE Server, in which
        case it uses this address both to listen for incoming
        TCP connections and to bind outgoing TCP connections.";
    }
    leaf enabled {
      type boolean;
      default "true";
      description
        "The administrative status of this PCEP
        Entity; set to true when UP.";
    }
  }
}
```

```
}
leaf role {
  type pcep-role;
  mandatory true;
  description
    "The role that this entity can play.
     Takes one of the following values.
     - unknown(0): this PCEP Entity role is not
       known.
     - pcc(1): this PCEP Entity is a PCC.
     - pce(2): this PCEP Entity is a PCE.
     - pcc-and-pce(3): this PCEP Entity is both
       a PCC and a PCE.";
}
leaf description {
  type string;
  description
    "Description of the PCEP entity configured
     by the user";
}
leaf speaker-entity-id {
  if-feature "sync-opt";
  type string;
  description
    "The Speaker Entity Identifier";
  reference
    "RFC 8232: Optimizations of Label Switched
     Path State Synchronization Procedures for
     a Stateful PCE";
}
leaf admin-status {
  type boolean;
  default "true";
  description
    "The administrative status of this PCEP Entity.
     The value true represents admin status as up.
     This is the desired operational status as
     currently set by an operator or by default in
     the implementation. The value of oper-status
     represents the current status of an attempt to
     reach this desired status.";
}
leaf index {
  type uint32;
  config false;
  description
    "The index of the operational PCEP entity";
}
```

```
leaf oper-status {
  type pcep-oper-status;
  config false;
  description
    "The operational status of the PCEP entity.
    Takes one of the following values.
    - oper-status-up(1): the PCEP entity is active.
    - oper-status-down(2): the PCEP entity is inactive.
    - oper-status-going-up(3): the PCEP entity is
    activating.
    - oper-status-going-down(4): the PCEP entity is
    deactivating.
    - oper-status-failed(5): the PCEP entity has
    failed and will recover when possible.
    - oper-status-failed-perm(6): the PCEP entity
    has failed and will not recover without
    operator intervention.";
}
uses info {
  description
    "Local PCEP entity information";
}
container pce-info {
  when "../role = 'pce'"
    + "or "
    + "../role = 'pcc-and-pce'" {
    description
      "These fields are applicable when the role is PCE.";
  }
  description
    "The Local PCE Entity PCE information";
  uses pce-info {
    description
      "Local PCE information";
  }
  container path-key {
    if-feature "path-key";
    description
      "Path-Key Configuration";
    reference
      "RFC 5520: Preserving Topology Confidentiality in Inter-
      Domain Path Computation Using a Path-Key-Based Mechanism";
    leaf enabled {
      type boolean;
      description
        "Enabled or Disabled; set to true when Enabled";
    }
    leaf discard-timer {
```

```
        type uint32;
        units "minutes";
        default "10";
        description
            "A timer to discard unwanted path-keys";
    }
    leaf reuse-time {
        type uint32;
        units "minutes";
        default "30";
        description
            "A time after which the path-keys could be reused";
    }
    leaf pce-id {
        type inet:ip-address-no-zone;
        description
            "PCE Address to be used in each Path-Key Subobject
            (PKS)";
    }
}
}
leaf connect-timer {
    type uint16 {
        range "1..65535";
    }
    units "seconds";
    default "60";
    description
        "The time in seconds that the PCEP entity will wait
        to establish a TCP connection with a peer. If a
        TCP connection is not established within this time
        then PCEP aborts the session setup attempt.";
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}
leaf connect-max-retry {
    type uint32;
    default "5";
    description
        "The maximum number of times the system tries to
        establish a TCP connection to a peer before the
        session with the peer transitions to the idle
        state.";
    reference
        "RFC 5440: Path Computation Element (PCE) Communication
        Protocol (PCEP)";
}
```

```
leaf init-back-off-timer {
  type uint16 {
    range "1..65535";
  }
  units "seconds";
  description
    "The initial back-off time in seconds for retrying
    a failed session setup attempt to a peer.
    The back-off time increases for each failed
    session setup attempt, until a maximum back-off
    time is reached. The maximum back-off time is the
    max-back-off-timer leaf.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf max-back-off-timer {
  type uint32;
  units "seconds";
  description
    "The maximum back-off time in seconds for retrying
    a failed session setup attempt to a peer.
    The back-off time increases for each failed session
    setup attempt, until this maximum value is reached.
    Session setup attempts then repeat periodically
    without any further increase in back-off time.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf open-wait-timer {
  type uint16;
  units "seconds";
  config false;
  description
    "The time in seconds that the PCEP entity will wait
    to receive an Open message from a peer after the
    TCP connection has come up.
    If no Open message is received within this time then
    PCEP terminates the TCP connection and deletes the
    associated sessions.";
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}
leaf keep-wait-timer {
  type uint16;
  units "seconds";
```

```
    config false;
    description
      "The time in seconds that the PCEP entity will wait
       to receive a Keepalive or PCErr message from a peer
       during session initialization after receiving an
       Open message. If no Keepalive or PCErr message is
       received within this time then PCEP terminates the
       TCP connection and deletes the associated
       sessions.";
    reference
      "RFC 5440: Path Computation Element (PCE) Communication
       Protocol (PCEP)";
  }
  leaf keepalive-timer {
    type uint8;
    units "seconds";
    default "30";
    description
      "The Keepalive timer that this PCEP
       entity will propose in the initial Open message of
       each session it is involved in. This is the
       maximum time between two consecutive messages sent
       to a peer. Zero means that the PCEP entity prefers
       not to send Keepalives at all.
       Note that the actual Keepalive transmission
       intervals, in either direction of an active PCEP
       session, are determined by negotiation between the
       peers as specified by RFC 5440, and so may differ
       from this configured value.";
    reference
      "RFC 5440: Path Computation Element (PCE) Communication
       Protocol (PCEP)";
  }
  leaf dead-timer {
    type uint8;
    units "seconds";
    must '(> ../keepalive-timer)' {
      error-message "The DeadTimer must be "
        + "larger than the Keepalive timer";
    }
    default "120";
    description
      "The DeadTimer that this PCEP entity will propose
       in the initial Open message of each session it is
       involved in. This is the time after which a peer
       should declare a session down if it does not
       receive any PCEP messages. Zero suggests that the
       peer does not run a DeadTimer at all.";
```

```
    reference
      "RFC 5440: Path Computation Element (PCE) Communication
       Protocol (PCEP)";
  }
  leaf allow-negotiation {
    type boolean;
    default "true";
    description
      "Whether the PCEP entity will permit negotiation of
       session parameters.";
  }
  leaf max-keepalive-timer {
    type uint8;
    units "seconds";
    description
      "The maximum value that this PCEP entity will
       accept from a peer for the interval between
       Keepalive transmissions. Zero means that the PCEP
       entity will allow no Keepalive transmission at
       all.";
  }
  leaf max-dead-timer {
    type uint8;
    units "seconds";
    description
      "The maximum value in seconds, that this PCEP
       entity will accept from a peer for the DeadTimer.
       Zero means that the PCEP entity will allow not
       running a DeadTimer.";
  }
  leaf min-keepalive-timer {
    type uint8;
    units "seconds";
    description
      "The minimum value in seconds, that this PCEP
       entity will accept for the interval between
       Keepalive transmissions. Zero means that the
       PCEP entity insists on no Keepalive
       transmission at all.";
  }
  leaf min-dead-timer {
    type uint8;
    units "seconds";
    description
      "The minimum value in seconds, that this PCEP
       entity will accept for the DeadTimer. Zero
       means that the PCEP entity insists on not
       running a DeadTimer.";
```



```
}
leaf sync-timer {
  if-feature "svec";
  type uint16;
  units "seconds";
  default "60";
  description
    "The value of SyncTimer in seconds is used in the
     case of synchronized path computation request
     using the SVEC object. Consider the case where a
     PCReq message is received by a PCE that contains
     the SVEC object referring to M synchronized path
     computation requests. If after the expiration of
     the SyncTimer all the M path computation requests
     have not been, received a protocol error is
     triggered and the PCE must cancel the whole set
     of path computation requests.
     The aim of the SyncTimer is to avoid the storage
     of unused synchronized requests should one of
     them get lost for some reasons (for example, a
     misbehaving PCC).
     Zero means that the PCEP entity does not use the
     SyncTimer.";
  reference
    "RFC 5440: Path Computation Element (PCE)
     Communication Protocol (PCEP)";
}
leaf request-timer {
  type uint16 {
    range "1..65535";
  }
  units "seconds";
  description
    "The maximum time that the PCEP entity will wait
     for a response to a PCReq message.";
}
leaf max-sessions {
  type uint32;
  description
    "Maximum number of sessions involving this PCEP
     entity that can exist at any time.";
}
leaf max-unknown-reqs {
  type uint32;
  default "5";
  description
    "The maximum number of unrecognized requests and
     replies that any session on this PCEP entity is
```

```
    willing to accept per minute before terminating
    the session.
    A PCRep message contains an unrecognized reply
    if it contains an RP object whose request ID
    does not correspond to any in-progress request
    sent by this PCEP entity.
    A PCReq message contains an unrecognized request
    if it contains an RP object whose request ID is
    zero.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf max-unknown-msgs {
  type uint32;
  default "5";
  description
    "The maximum number of unknown messages that any
    session on this PCEP entity is willing to accept
    per minute before terminating the session.";
  reference
    "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf pcep-notification-max-rate {
  type uint32;
  mandatory true;
  description
    "This variable indicates the maximum number of
    notifications issued per second. If events
    occur more rapidly, the implementation may
    simply fail to emit these notifications during
    that period, or may queue them until an
    appropriate time. A value of 0 means no
    notifications are emitted and all should be
    discarded (that is, not queued).";
}
container stateful-parameter {
  if-feature "stateful";
  description
    "The configured stateful PCE parameters";
  leaf state-timeout {
    type uint32;
    units "seconds";
    description
      "When a PCEP session is terminated, a PCC
      waits for this time period before flushing
      LSP state associated with that PCEP session";
  }
}
```

```
        and reverting to operator-defined default
        parameters or behaviours.";
    }
    leaf redelegation-timeout {
        when "../role = 'pcc'"
        + "or "
        + "../role = 'pcc-and-pce'" {
            description
                "This field is applicable when the role is
                PCC";
        }
        type uint32;
        units "seconds";
        description
            "When a PCEP session is terminated, a PCC
            waits for this time period before revoking
            LSP delegation to a PCE and attempting to
            redelegate LSPs associated with the
            terminated PCEP session to an alternate
            PCE.";
    }
    leaf rpt-non-pcep-lsp {
        when "../role = 'pcc'"
        + "or "
        + "../role = 'pcc-and-pce'" {
            description
                "This field is applicable when the role is
                PCC";
        }
        type boolean;
        default "true";
        description
            "If set, a PCC reports LSPs that are not
            controlled by any PCE (for example, LSPs
            that are statically configured at the
            PCC). ";
    }
    reference
        "RFC 8231: Path Computation Element Communication Protocol
        (PCEP) Extensions for Stateful PCE";
}
container of-list {
    when "../role = 'pce'"
    + "or "
    + "../role = 'pcc-and-pce'" {
        description
            "These field are applicable when the role is
            PCE";
    }
}
```

```
    }
    if-feature "objective-function";
    uses of-list;
    description
        "The authorized OF-List at PCE for all peers";
}
container lsp-db {
    if-feature "stateful";
    config false;
    description
        "The LSP-DB";
    leaf db-ver {
        when "../role = 'pcc'"
        + "or "
        + "../role = 'pcc-and-pce'" {
            description
                "This field is applicable when the role is
                PCC";
        }
        if-feature "sync-opt";
        type uint64;
        description
            "The LSP State Database Version Number";
    }
    list association-list {
        if-feature "association";
        key "type id source global-source extended-id";
        description
            "List of all PCEP associations";
        reference
            "RFC 8697: Path Computation Element Communication
            Protocol (PCEP) Extensions for Establishing
            Relationships between Sets of Label Switched
            Paths (LSPs)";
        leaf type {
            type identityref {
                base te-types:association-type;
            }
            description
                "The PCEP Association Type";
            reference
                "IANA PCEP: ASSOCIATION Type Field in Path
                Computation Element Protocol (PCEP) Numbers";
        }
        leaf id {
            type uint16;
            description
                "PCEP Association ID";
        }
    }
}
```

```
    }
    leaf source {
      type inet:ip-address-no-zone;
      description
        "PCEP Association Source.";
    }
    leaf global-source {
      type uint32;
      description
        "PCEP Global Association Source.";
    }
    leaf extended-id {
      type string;
      description
        "Additional information to support unique
         identification (Extended Association ID).";
    }
    list lsp {
      key "plsp-id pcc-id lsp-id";
      description
        "List of all LSP in this association";
      leaf plsp-id {
        type leafref {
          path "/pcep/entity/lsp-db/"
            + "lsp/plsp-id";
        }
        description
          "Reference to PLSP-ID in LSP-DB";
      }
      leaf pcc-id {
        type leafref {
          path "/pcep/entity/lsp-db/"
            + "lsp/pcc-id";
        }
        description
          "Reference to PCC-ID in LSP-DB";
      }
      leaf lsp-id {
        type leafref {
          path "/pcep/entity/lsp-db/"
            + "lsp/lsp-id";
        }
        description
          "Reference to LSP ID in LSP-DB";
      }
    }
  }
}
list lsp {
```

```
key "plsp-id pcc-id lsp-id";
description
  "List of all LSPs in LSP-DB";
leaf plsp-id {
  type uint32 {
    range "1..1048575";
  }
  description
    "A PCEP-specific identifier for the LSP. A PCC
    creates a unique PLSP-ID for each LSP that is
    constant for the lifetime of a PCEP session.
    PLSP-ID is 20 bits with 0 and 0xFFFF are
    reserved";
}
leaf pcc-id {
  type inet:ip-address-no-zone;
  description
    "The local internet address of the PCC, that
    generated the PLSP-ID.";
}
leaf source {
  type inet:ip-address-no-zone;
  description
    "Tunnel sender address extracted from
    LSP-IDENTIFIERS TLV";
  reference
    "RFC 8231: Path Computation Element
    Communication Protocol (PCEP) Extensions
    for Stateful PCE";
}
leaf destination {
  type inet:ip-address-no-zone;
  description
    "Tunnel endpoint address extracted from
    LSP-IDENTIFIERS TLV";
  reference
    "RFC 8231: Path Computation Element
    Communication Protocol (PCEP) Extensions
    for Stateful PCE";
}
leaf tunnel-id {
  type uint16;
  description
    "Tunnel identifier used in the LSP-IDENTIFIERS
    TLV that remains constant over the life
    of the tunnel.";
  reference
    "RFC 8231: Path Computation Element
```

```
        Communication Protocol (PCEP) Extensions
        for Stateful PCE";
    }
    leaf lsp-id {
        type uint16;
        description
            "Identifier used in the LSP-IDENTIFIERS TLV
            that can be changed to allow a sender to share
            resources with itself.";
        reference
            "RFC 8231: Path Computation Element
            Communication Protocol (PCEP) Extensions
            for Stateful PCE";
    }
    leaf extended-tunnel-id {
        type inet:ip-address-no-zone;
        description
            "Extended Tunnel ID of the LSP in LSP-IDENTIFIERS
            TLV.";
        reference
            "RFC 8231: Path Computation Element
            Communication Protocol (PCEP) Extensions
            for Stateful PCE";
    }
    leaf admin-state {
        type boolean;
        default "true";
        description
            "The desired operational state";
    }
    leaf operational-state {
        type operational-state;
        description
            "The operational status of the LSP";
    }
    container delegated {
        description
            "The delegation related parameters";
        leaf enabled {
            type boolean;
            default "false";
            description
                "LSP is delegated or not; set to true when
                delegated";
        }
        leaf peer {
            type leafref {
                path "/pcep/entity/peers/peer/addr";
            }
        }
    }
}
```

```
    }
    must '(!../enabled = true())' {
        error-message "The LSP must be delegated";
    }
    description
        "At the PCC, the reference to the PCEP peer to
        which LSP is delegated; At the PCE, the
        reference to the PCEP peer which delegated this
        LSP";
}
leaf srp-id {
    type uint32 {
        range "1..4294967294";
    }
    description
        "The last SRP-ID-number associated with this
        LSP.";
}
}
container initiation {
    if-feature "pce-initiated";
    description
        "The PCE initiation related parameters";
    reference
        "RFC 8281: Path Computation Element Communication
        Protocol (PCEP) Extensions for PCE-Initiated LSP
        Setup in a Stateful PCE Model";
    leaf enabled {
        type boolean;
        default "false";
        description
            "Set to true if this LSP is initiated by a PCE";
    }
    leaf peer {
        type leafref {
            path "/pcep/entity/peers/peer/addr";
        }
        must '(!../enabled = true())' {
            error-message "The LSP must be PCE-Initiated";
        }
        description
            "At the PCC, the reference to the PCEP peer
            that initiated this LSP; At the PCE, the
            reference to the PCEP peer where the LSP
            is initiated";
    }
}
leaf symbolic-path-name {
```



```
    type string;
    description
      "The symbolic path name associated with the LSP.";
    reference
      "RFC 8231: Path Computation Element Communication
        Protocol (PCEP) Extensions for Stateful PCE";
  }
  leaf last-error {
    type identityref {
      base lsp-error;
    }
    description
      "The last error for the LSP.";
  }
  leaf pst {
    type identityref {
      base te-types:path-signaling-type;
    }
    default "te-types:path-setup-rsvp";
    description
      "The Path Setup Type (PST)";
    reference
      "RFC 8408: Conveying Path Setup Type in PCE
        Communication Protocol (PCEP) Messages";
  }
  list association-list {
    if-feature "association";
    key "type id source global-source extended-id";
    description
      "List of all PCEP associations";
    leaf type {
      type leafref {
        path "/pcep/entity/lsp-db/"
          + "association-list/type";
      }
      description
        "PCEP Association Type";
    }
    leaf id {
      type leafref {
        path "/pcep/entity/lsp-db/"
          + "association-list/id";
      }
      description
        "PCEP Association ID";
    }
    leaf source {
      type leafref {
```

```
        path "/pcep/entity/lsp-db/"
          + "association-list/source";
      }
      description
        "PCEP Association Source.";
    }
    leaf global-source {
      type leafref {
        path "/pcep/entity/lsp-db/"
          + "association-list/global-source";
      }
      description
        "PCEP Association Global Source.";
    }
    leaf extended-id {
      type leafref {
        path "/pcep/entity/lsp-db/"
          + "association-list/extended-id";
      }
      description
        "Additional information to
        support unique identification.";
    }
    reference
      "RFC 8697: Path Computation Element Communication
      Protocol (PCEP) Extensions for Establishing
      Relationships between Sets of Label Switched
      Paths (LSPs)";
  }
}
}
container path-keys {
  when "../role = 'pce' or ../role = 'pcc-and-pce'" {
    description
      "These fields are applicable when the role is
      PCE";
  }
  if-feature "path-key";
  config false;
  description
    "The path-keys generated by the PCE";
  reference
    "RFC 5520: Preserving Topology Confidentiality
    in Inter-Domain Path Computation Using a Path-
    Key-Based Mechanism";
  list path-keys {
    key "path-key";
    description
```

```
    "The list of path-keys generated by the PCE";
  leaf path-key {
    type uint16;
    description
      "The identifier, or token used to represent
       the Confidential Path Segment (CPS) within
       the context of the PCE";
  }
  container cps {
    description
      "The Confidential Path Segment (CPS)";
    list explicit-route-objects {
      key "index";
      description
        "List of explicit route objects";
      leaf index {
        type uint32;
        description
          "ERO subobject index";
      }
      uses te-types:explicit-route-hop;
    }
  }
  leaf pcc-original {
    type leafref {
      path "/pcep/entity/peers/peer/addr";
    }
    description
      "Reference to PCC peer address of
       the original request";
  }
  leaf req-id {
    type uint32;
    description
      "The request ID of the original PCReq.";
  }
  leaf retrieved {
    type boolean;
    description
      "If path-key has been retrieved yet";
  }
  leaf pcc-retrieved {
    type leafref {
      path "/pcep/entity/peers/peer/addr";
    }
    must '(!../retrieved = true())' {
      error-message "The Path-key should be retrieved";
    }
  }
```

```
        description
          "Reference to PCC peer address which
           retrieved the path-key";
      }
      leaf creation-time {
        type yang:timestamp;
        description
          "The timestamp value at the time this Path-Key
           was created.";
      }
      leaf discard-time {
        type uint32;
        units "minutes";
        description
          "A time after which this path-keys will be
           discarded";
      }
      leaf reuse-time {
        type uint32;
        units "minutes";
        description
          "A time after which this path-keys could be
           reused";
      }
    }
  }
  container peers {
    description
      "The list of configured peers for the
       entity (remote PCE)";
    list peer {
      key "addr";
      description
        "The peer configured for the entity.
         (remote PCE)";
      leaf addr {
        type inet:ip-address-no-zone;
        description
          "The local Internet address of this
           PCEP peer.";
      }
      leaf role {
        type pcep-role;
        must '(!= "pcc-and-pce")' {
          error-message
            "The PCEP peer cannot be both
             PCE and PCC at the same time";
        }
      }
    }
  }
}
```

```
mandatory true;
description
  "The role of the PCEP Peer.
  Takes one of the following values.
  - unknown(0): this PCEP peer role is not
  known.
  - pcc(1): this PCEP peer is a PCC.
  - pce(2): this PCEP peer is a PCE.
  - pcc-and-pce(3): is not allowed as PCEP
  peer cannot be acting as both a PCC and a
  PCE at the sametime.";
}
leaf description {
  type string;
  description
    "Description of the PCEP peer
    configured by the user";
}
uses info {
  description
    "PCE Peer information";
}
container pce-info {
  uses pce-info {
    description
      "PCE Peer information";
  }
  description
    "The PCE Peer information";
}
leaf delegation-pref {
  if-feature "stateful";
  type uint8 {
    range "0..7";
  }
  description
    "The PCE peer delegation preference.";
}
container auth {
  description
    "The Authentication options";
  choice auth-type-selection {
    description
      "Options for expressing authentication
      setting.";
    case auth-key-chain {
      leaf key-chain {
        type key-chain:key-chain-ref;
      }
    }
  }
}
```

```
        description
            "key-chain name.";
    }
}
case auth-key {
    leaf crypto-algorithm {
        type identityref {
            base key-chain:crypto-algorithm;
        }
        mandatory true;
        description
            "Cryptographic algorithm associated
             with key.";
    }
    choice key-string-style {
        description
            "Key string styles";
        case keystack {
            leaf keystack {
                nacm:default-deny-all;
                type string;
                description
                    "Key string in ASCII format.";
            }
        }
        case hexadecimal {
            if-feature "key-chain:hex-key-string";
            leaf hexadecimal-string {
                nacm:default-deny-all;
                type yang:hex-string;
                description
                    "Key in hexadecimal string format. When
                     compared to ASCII, specification in
                     hexadecimal affords greater key entropy
                     with the same number of internal
                     key-string octets. Additionally, it
                     discourages usage of well-known words or
                     numbers.";
            }
        }
    }
}
}
case auth-tls {
    if-feature "tls";
    choice role {
        description
            "The role of the local entity";
        case server {
```

```
        container tls-server {
            uses tlss:tls-server-grouping {
                description
                    "Server TLS information.";
            }
            description
                "TLS related information";
        }
    }
    case client {
        container tls-client {
            uses tlsc:tls-client-grouping {
                description
                    "Client TLS information.";
            }
            description
                "TLS related information";
        }
    }
}

leaf discontinuity-time {
    type yang:timestamp;
    config false;
    description
        "The timestamp of the time when the information and
        statistics were last reset.";
}

leaf initiate-session {
    type boolean;
    config false;
    description
        "Indicates whether the local PCEP entity initiates
        sessions to this peer, or waits for the peer to
        initiate a session.";
}

leaf session-exists {
    type boolean;
    config false;
    description
        "Indicates whether a session with
        this peer currently exists.";
}

leaf session-up-time {
    type yang:timestamp;
    config false;
```

```
    description
      "The timestamp value of the last time a
       session with this peer was successfully
       established.";
  }
  leaf session-fail-time {
    type yang:timestamp;
    config false;
    description
      "The timestamp value of the last time a
       session with this peer failed to be
       established.";
  }
  leaf session-fail-up-time {
    type yang:timestamp;
    config false;
    description
      "The timestamp value of the last time a
       session with this peer failed from
       active.";
  }
  container sessions {
    config false;
    description
      "This entry represents a single PCEP
       session in which the local PCEP entity participates.
       This entry exists only if the corresponding PCEP
       session has been initialized by some event, such as
       manual user configuration, auto-discovery of a peer,
       or an incoming TCP connection.";
    list session {
      key "initiator";
      description
        "The list of sessions, note that for a time being
         two sessions may exist for a peer";
      leaf initiator {
        type pcep-initiator;
        description
          "The initiator of the session, that is, whether
           the TCP connection was initiated by the local
           PCEP entity or the peer.
           There is a window during session
           initialization where two sessions can exist
           between a pair of PCEP speakers, each
           initiated by one of the speakers. One of
           these sessions is always discarded before it
           leaves OpenWait state. However, before it is
           discarded, two sessions to the given peer
```



```
        appear transiently in this YANG module. The
        sessions are distinguished by who initiated
        them, and so this field is the key.";
    }
    leaf role {
        type leafref {
            path "/pcep/entity/role";
        }
        description
            "The reference to peer role .";
    }
    leaf state-last-change {
        type yang:timestamp;
        description
            "The timestamp value at the time this
            session entered its current state as
            denoted by the state leaf.";
    }
    leaf state {
        type pcep-sess-state;
        description
            "The current state of the session.
            The set of possible states excludes the
            idle state since entries do not exist
            in the idle state.";
    }
    leaf session-creation {
        type yang:timestamp;
        description
            "The timestamp value at the time this
            session was created.";
    }
    leaf connect-retry {
        type yang:counter32;
        description
            "The number of times that the local PCEP
            entity has attempted to establish a TCP
            connection for this session without
            success. The PCEP entity gives up when
            this reaches connect-max-retry.";
    }
    leaf local-id {
        type uint8;
        description
            "The value of the PCEP session ID used by
            the local PCEP entity in the Open message
            for this session. If state is tcp-pending
            then this is the session ID that will be
```

```
        used in the Open message. Otherwise, this
        is the session ID that was sent in the
        Open message.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}
leaf remote-id {
    type uint8;
    must "((../state != 'tcp-pending' "
    + "and "
    + "../state != 'open-wait' )"
    + "or "
    + "((../state = 'tcp-pending' "
    + "or "
    + "../state = 'open-wait' )"
    + "and (. = 0)))" {
        error-message "Invalid remote-id";
    description
        "If state is TCPPending or OpenWait then
        this leaf is not used and MUST be set to
        zero.";
    }
    description
        "The value of the PCEP session ID used by the
        peer in its Open message for this session.";
    reference
        "RFC 5440: Path Computation Element (PCE)
        Communication Protocol (PCEP)";
}
leaf keepalive-timer {
    type uint8;
    units "seconds";
    must "((../state = 'session-up' "
    + "or "
    + "../state != 'session-up' "
    + "and (. = 0)))" {
        error-message "Invalid Keepalive timer";
    description
        "This field is used if and only if state is
        session-up. Otherwise, it is not used and
        MUST be set to zero.";
    }
    description
        "The agreed maximum interval at which the local
        PCEP entity transmits PCEP messages on this PCEP
        session. Zero means that the local PCEP entity
        never sends Keepalives on this session.";
```

```
reference
  "RFC 5440: Path Computation Element (PCE)
    Communication Protocol (PCEP)";
}
leaf peer-keepalive-timer {
  type uint8;
  units "seconds";
  must "(../state = 'session-up' "
    + "or "
    + "(../state != 'session-up' "
    + "and "
    + "(. = 0)))" {
    error-message "Invalid Peer Keepalive timer";
    description
      "This field is used if and only if state is
        session-up. Otherwise, it is not used and MUST
        be set to zero.";
  }
  description
    "The agreed maximum interval at which the peer
      transmits PCEP messages on this PCEP session.
      Zero means that the peer never sends Keepalives
      on this session.";
  reference
    "RFC 5440: Path Computation Element (PCE)
      Communication Protocol (PCEP)";
}
leaf dead-timer {
  type uint8;
  units "seconds";
  description
    "The DeadTimer interval for this PCEP session.";
  reference
    "RFC 5440: Path Computation Element (PCE)
      Communication Protocol (PCEP)";
}
leaf peer-dead-timer {
  type uint8;
  units "seconds";
  must "((../state != 'tcp-pending' "
    + "and "
    + "../state != 'open-wait' )"
    + "or "
    + "((../state = 'tcp-pending' "
    + "or "
    + "../state = 'open-wait' )"
    + "and "
    + "(. = 0)))" {
```

```
        error-message "Invalid Peer DeadTimer";
        description
            "If state is TCPPending or OpenWait then this
             leaf is not used and MUST be set to zero.";
    }
    description
        "The peer's DeadTimer interval
         for this PCEP session.";
    reference
        "RFC 5440: Path Computation Element (PCE)
         Communication Protocol (PCEP)";
}
leaf ka-hold-time-rem {
    type uint8;
    units "seconds";
    must "((../state != 'tcp-pending' "
        + "and "
        + "../state != 'open-wait' ) "
        + "or "
        + "((../state = 'tcp-pending' "
        + "or "
        + "../state = 'open-wait' ) "
        + "and "
        + "(. = 0)))" {
        error-message
            "Invalid Keepalive hold time remaining";
        description
            "If state is TCPPending or OpenWait then this
             field is not used and MUST be set to zero.";
    }
    description
        "The Keepalive hold time remaining for this
         session.";
}
leaf overloaded {
    type boolean;
    description
        "If the local PCEP entity has informed the peer that
         it is currently overloaded, then this is set to true.
         Otherwise, it is set to false.";
    reference
        "RFC 5440: Path Computation Element (PCE)
         Communication Protocol (PCEP)";
}
leaf overloaded-timestamp {
    when '(!../overloaded = true())' {
        description
            "Valid when overloaded";
    }
}
```

```
    }
    type yang:timestamp;
    description
      "The timestamp value of the time when the
       overloaded field was set to true.";
  }
  leaf overload-time {
    type uint32;
    units "seconds";
    must '((../overloaded = true()) '
      + 'or ((../overloaded != true()) '
      + 'and (. = 0)))' {
      error-message "Invalid overload-time";
      description
        "This field is only used if overloaded is set to
         true. Otherwise, it is not used and MUST be set
         to zero.";
    }
    description
      "The interval of time that is remaining until the
       local PCEP entity will cease to be overloaded on
       this session.";
    reference
      "RFC 5440: Path Computation Element (PCE)
       Communication Protocol (PCEP)";
  }
  leaf peer-overloaded {
    type boolean;
    description
      "If the peer has informed the local PCEP entity
       that it is currently overloaded, then this is
       set to true. Otherwise, it is set to false.";
    reference
      "RFC 5440: Path Computation Element (PCE)
       Communication Protocol (PCEP)";
  }
  leaf peer-overloaded-timestamp {
    when '(!../peer-overloaded = true())' {
      description
        "Valid when Peer is overloaded";
    }
    type yang:timestamp;
    description
      "The timestamp value of the time when the
       peer-overloaded field was set to true.";
  }
  leaf peer-overload-time {
    type uint32;
```

```
units "seconds";
must ' (../peer-overloaded = '
  + 'true()) or '
  + ' (../peer-overloaded !='
  + 'true())'
  + ' and '
  + ' (. = 0))' {
  error-message "Invalid peer overload time";
  description
    "This field is only used if peer-overloaded
    is set to true. Otherwise, it is not used
    and MUST be set to zero.";
}
description
  "The interval of time that is remaining until
  the peer will cease to be overloaded. If it
  is not known how long the peer will stay in
  overloaded state, this leaf is set to zero.";
reference
  "RFC 5440: Path Computation Element (PCE)
  Communication Protocol (PCEP)";
}
leaf lspdb-sync {
  if-feature "stateful";
  type sync-state;
  description
    "The LSP-DB state synchronization status.";
  reference
    "RFC 8231: Path Computation Element Communication
    Protocol (PCEP) Extensions for Stateful PCE";
}
leaf recv-db-ver {
  when "../role = 'pcc'"
  + "or "
  + "../role = 'pcc-and-pce'" {
  description
    "This field is applicable when the role is
    PCC";
  }
  if-feature "stateful";
  if-feature "sync-opt";
  type uint64;
  description
    "The last received LSP State Database Version
    Number";
  reference
    "RFC 8231: Path Computation Element Communication
    Protocol (PCEP) Extensions for Stateful PCE";
}
```

```
}
container of-list {
  when "../role = 'pce'"
    + "or "
    + "../role = 'pcc-and-pce'" {
    description
      "These fields are applicable when the role is
      PCE";
  }
  if-feature "objective-function";
  uses of-list;
  description
    "Indicate the list of supported OF on this
    session";
  reference
    "RFC 5541: Encoding of Objective Functions in
    the Path Computation Element Communication
    Protocol (PCEP)";
}
container pst-list {
  when "../role = 'pce'"
    + "or "
    + "../role = 'pcc-and-pce'" {
  description
    "These fields are applicable when the role is
    PCE";
  }
  description
    "Indicate the list of supported
    PST on this session";
  reference
    "RFC 8408: Conveying Path Setup Type in PCE
    Communication Protocol (PCEP) Messages";
  list path-setup-type {
    key "pst";
    description
      "The list of PST";
    leaf pst {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "The PST supported";
    }
  }
}
}
container assoc-type-list {
  if-feature "association";
```

```
    description
      "Indicate the list of supported association types
       on this session";
    reference
      "RFC 8697: Path Computation Element Communication
       Protocol (PCEP) Extensions for Establishing
       Relationships between Sets of Label Switched
       Paths (LSPs)";
    list assoc-type {
      key "at";
      description
        "The list of authorized association types";
      leaf at {
        type identityref {
          base te-types:association-type;
        }
        description
          "The association type authorized";
      }
    }
  }
  leaf speaker-entity-id {
    if-feature "sync-opt";
    type string;
    description
      "The Speaker Entity Identifier";
    reference
      "RFC 8232: Optimizations of Label Switched
       Path State Synchronization Procedures for
       a Stateful PCE";
  }
}
// session
}
// sessions
}
//peer
}
//peers
}
//entity
}

//pcep
/*
 * Notifications
 */
```



```
notification pcep-session-up {
  description
    "This notification is sent when the value of
     '/pcep/peers/peer/sessions/session/state'
     enters the 'session-up' state.";
  uses notification-instance-hdr;
  uses notification-session-hdr;
  leaf state-last-change {
    type yang:timestamp;
    description
      "The timestamp value at the time this session
       entered its current state as denoted by the state
       leaf.";
  }
  leaf state {
    type pcep-sess-state;
    description
      "The current state of the session.
       The set of possible states excludes the idle state
       since entries do not exist in the idle state.";
  }
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
     Protocol (PCEP)";
}

//notification

notification pcep-session-down {
  description
    "This notification is sent when the value of
     '/pcep/peers/peer/sessions/session/state'
     leaves the 'session-up' state.";
  uses notification-instance-hdr;
  leaf session-initiator {
    type pcep-initiator;
    description
      "The initiator of the session.";
  }
  leaf state-last-change {
    type yang:timestamp;
    description
      "The timestamp value at the time this session
       entered its current state as denoted by the state
       leaf.";
  }
  leaf state {
    type pcep-sess-state;
  }
}
```

```
    description
      "The current state of the session.
       The set of possible states excludes the idle state
       since entries do not exist in the idle state.";
  }
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
     Protocol (PCEP)";
}

//notification

notification pcep-session-local-overload {
  description
    "This notification is sent when the local PCEP entity
     enters overload state for a peer.";
  uses notification-instance-hdr;
  uses notification-session-hdr;
  leaf overloaded {
    type boolean;
    description
      "If the local PCEP entity has informed the peer
       that it is currently overloaded, then this is set
       to true. Otherwise, it is set to false.";
  }
  leaf overloaded-timestamp {
    type yang:timestamp;
    description
      "The timestamp value of the time when the
       overloaded field was set to true.";
  }
  leaf overload-time {
    type uint32;
    units "seconds";
    description
      "The interval of time that is remaining until the
       local PCEP entity will cease to be overloaded on
       this session.";
  }
  reference
    "RFC 5440: Path Computation Element (PCE) Communication
     Protocol (PCEP)";
}

//notification

notification pcep-session-local-overload-clear {
  description
```

```
    "This notification is sent when the local PCEP entity
      leaves overload state for a peer.";
uses notification-instance-hdr;
leaf overloaded {
  type boolean;
  description
    "If the local PCEP entity has informed the peer
      that it is currently overloaded, then this is set
      to true. Otherwise, it is set to false.";
}
leaf overloaded-clear-timestamp {
  type yang:timestamp;
  description
    "The timestamp value of the time when the
      overloaded field was set to false.";
}
reference
  "RFC 5440: Path Computation Element (PCE) Communication
    Protocol (PCEP)";
}

//notification

notification pcep-session-peer-overload {
  description
    "This notification is sent when a peer enters overload
      state.";
uses notification-instance-hdr;
uses notification-session-hdr;
leaf peer-overloaded {
  type boolean;
  description
    "If the peer has informed the local PCEP entity that
      it is currently overloaded, then this is set to
      true. Otherwise, it is set to false.";
}
leaf peer-overloaded-timestamp {
  type yang:timestamp;
  description
    "The timestamp value of the time when the
      peer-overloaded field was set to true.";
}
leaf peer-overload-time {
  type uint32;
  units "seconds";
  description
    "The interval of time that is remaining until the
      peer will cease to be overloaded. If it is not
```

```
        known how long the peer will stay in overloaded
        state, this leaf is set to zero.";
    }
    reference
      "RFC 5440: Path Computation Element (PCE) Communication
      Protocol (PCEP)";
  }

  //notification

  notification pcep-session-peer-overload-clear {
    description
      "This notification is sent when a peer leaves overload
      state.";
    uses notification-instance-hdr;
    leaf peer-overloaded {
      type boolean;
      description
        "If the peer has informed the local PCEP entity that
        it is currently overloaded, then this is set to
        true. Otherwise, it is set to false.";
    }
    leaf peer-overloaded-clear-timestamp {
      type yang:timestamp;
      description
        "The timestamp value of the time when the
        peer-overloaded field was set to false.";
    }
    reference
      "RFC 5440: Path Computation Element (PCE) Communication
      Protocol (PCEP)";
  }

  //notification
  /*
   * RPC
   */

  rpc trigger-resync {
    if-feature "stateful";
    if-feature "sync-opt";
    description
      "Trigger the resynchronization at the PCE";
    reference
      "RFC 8232: Optimizations of Label Switched Path State
      Synchronization Procedures for a Stateful PCE";
    input {
      leaf pcc {
```

```
        type leafref {
            path "/pcep/entity/peers/peer/addr";
        }
        description
            "The IP address to identify the PCC. The state
             synchronization is re-triggered for all LSPs from
             the PCC. The rpc on the PCC will be ignored.";
    }
}
//input
}

//rpc
}
<CODE ENDS>
```

9.2. ietf-pcep-stats module

```
<CODE BEGINS> file "ietf-pcep-stats@2022-01-25.yang"
module ietf-pcep-stats {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep-stats";
    prefix pcep-stats;

    import ietf-pcep {
        prefix pcep;
        reference
            "RFC XXXX: A YANG Data Model for Path Computation
             Element Communications Protocol (PCEP)";
    }
    import ietf-yang-types {
        prefix yang;
        reference
            "RFC 6991: Common YANG Data Types";
    }

    organization
        "IETF PCE (Path Computation Element) Working Group";
    contact
        "WG Web:  <https://datatracker.ietf.org/wg/pce/about/>
        WG List:  <mailto:pce@ietf.org>
        Editor:   Dhruv Dhody
                 <mailto:dhruv.ietf@gmail.com>";
    description
        "The YANG module augments the PCEP yang operational
         model with statistics, counters and telemetry data.

         Copyright (c) 2022 IETF Trust and the persons identified as
```

authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Revised BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2022-01-25 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for Path Computation
      Element Communications Protocol (PCEP)";
}

/*
 * Groupings
 */

grouping pcep-stats {
  description
    "This grouping defines statistics for PCEP. It is used
      for both peer and current session.";
  leaf discontinuity-time {
    type yang:timestamp;
    description
      "The timestamp value of the time when the
        statistics were last reset.";
  }
  leaf rsp-time-avg {
    when "../pcep:role = 'pce' "
      + "or "
      + "../pcep:role = 'pcc-and-pce' " {
      description
        "Valid for PCEP Peer as PCE";
    }
    type uint32;
    units "milliseconds";
    description
      "The average response time. If an average response time
        has not been calculated then this leaf has the value
        zero.";
  }
}
```

```
leaf rsp-time-lwm {
  when "../..//pcep:role = 'pce'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type uint32;
  units "milliseconds";
  description
    "The smallest (low-water mark) response time seen.
    If no responses have been received then this leaf has
    the value zero.";
}
leaf rsp-time-hwm {
  when "../..//pcep:role = 'pce'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type uint32;
  units "milliseconds";
  description
    "The greatest (high-water mark) response time seen.
    If no responses have been received then this object
    has the value zero.";
}
leaf num-pcreq-sent {
  when "../..//pcep:role = 'pce'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCReq messages sent.";
}
leaf num-pcreq-rcvd {
  when "../..//pcep:role = 'pcc'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
```

```
        "The number of PCReq messages received.";
    }
    leaf num-pcrep-sent {
        when "../../pcep:role = 'pcc'"
        + "or "
        + "../../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of PCRep messages sent.";
    }
    leaf num-pcrep-rcvd {
        when "../../pcep:role = 'pce'"
        + "or "
        + "../../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of PCRep messages received.";
    }
    leaf num-pcerr-sent {
        type yang:counter32;
        description
            "The number of PCErr messages sent.";
    }
    leaf num-pcerr-rcvd {
        type yang:counter32;
        description
            "The number of PCErr messages received.";
    }
    leaf num-pcntf-sent {
        type yang:counter32;
        description
            "The number of PCNtf messages sent.";
    }
    leaf num-pcntf-rcvd {
        type yang:counter32;
        description
            "The number of PCNtf messages received.";
    }
    leaf num-keepalive-sent {
        type yang:counter32;
        description
            "The number of Keepalive messages sent.";
```



```
    }
    leaf num-keepalive-rcvd {
      type yang:counter32;
      description
        "The number of Keepalive messages received.";
    }
    leaf num-unknown-rcvd {
      type yang:counter32;
      description
        "The number of unknown messages received.";
    }
    leaf num-corrupt-rcvd {
      type yang:counter32;
      description
        "The number of corrupted PCEP message received.";
    }
    leaf num-req-sent {
      when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
        description
          "Valid for PCEP Peer as PCE";
      }
      type yang:counter32;
      description
        "The number of requests sent. A request corresponds
        1:1 with an RP object in a PCReq message. This might
        be greater than num-pcreq-sent because multiple
        requests can be batched into a single PCReq
        message.";
    }
    leaf num-req-sent-pend-rep {
      when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
        description
          "Valid for PCEP Peer as PCE";
      }
      type yang:counter32;
      description
        "The number of requests that have been sent for
        which a response is still pending.";
    }
    leaf num-req-sent-ero-rcvd {
      when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
        description
```

```
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
        "The number of requests that have been sent for
        which a response with an ERO object was received.
        Such responses indicate that a path was
        successfully computed by the peer.";
    }
    leaf num-req-sent-nopath-rcvd {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests that have been sent for
            which a response with a NO-PATH object was
            received. Such responses indicate that the peer
            could not find a path to satisfy the
            request.";
    }
    leaf num-req-sent-cancel-rcvd {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests that were cancelled with
            a PCNtf message. This might be different than
            num-pcntf-rcvd because not all PCNtf messages are
            used to cancel requests, and a single PCNtf message
            can cancel multiple requests.";
    }
    leaf num-req-sent-error-rcvd {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
```

```
        "The number of requests that were rejected with a
        PCErr message. This might be different than
        num-pcerr-rcvd because not all PCErr messages are
        used to reject requests, and a single PCErr message
        can reject multiple requests.";
    }
    leaf num-req-sent-timeout {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests that have been sent to a peer
            and have been abandoned because the peer has taken too
            long to respond to them.";
    }
    leaf num-req-sent-cancel-sent {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of requests that were sent to the peer and
            explicitly cancelled by the local PCEP entity sending
            a PCNtf.";
    }
    leaf num-req-rcvd {
        when "../pcep:role = 'pcc'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of requests received. A request
            corresponds 1:1 with an RP object in a PCReq
            message.
            This might be greater than num-pcreq-rcvd because
            multiple requests can be batched into a single
            PCReq message.";
    }
}
```

```
leaf num-req-rcvd-pend-rep {
  when "../pcep:role = 'pcc'"
  + "or "
  + "../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests that have been received for
    which a response is still pending.";
}
leaf num-req-rcvd-ero-sent {
  when "../pcep:role = 'pcc'"
  + "or "
  + "../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests that have been received for
    which a response with an ERO object was sent. Such
    responses indicate that a path was successfully
    computed by the local PCEP entity.";
}
leaf num-req-rcvd-nopath-sent {
  when "../pcep:role = 'pcc'"
  + "or "
  + "../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests that have been received for
    which a response with a NO-PATH object was sent. Such
    responses indicate that the local PCEP entity could
    not find a path to satisfy the request.";
}
leaf num-req-rcvd-cancel-sent {
  when "../pcep:role = 'pcc'"
  + "or "
  + "../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
```

```
description
  "The number of requests received that were cancelled
  by the local PCEP entity sending a PCNtf message.
  This might be different than num-pcntf-sent because
  not all PCNtf messages are used to cancel requests,
  and a single PCNtf message can cancel multiple
  requests.";
}
leaf num-req-rcvd-error-sent {
  when "../..//pcep:role = 'pcc'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests received that were cancelled
    by the local PCEP entity sending a PCErr message.
    This might be different than num-pcerr-sent because
    not all PCErr messages are used to cancel requests,
    and a single PCErr message can cancel multiple
    requests.";
}
leaf num-req-rcvd-cancel-rcvd {
  when "../..//pcep:role = 'pcc'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of requests that were received from the
    peer and explicitly cancelled by the peer sending
    a PCNtf.";
}
leaf num-rep-rcvd-unknown {
  when "../..//pcep:role = 'pce'"
  + "or "
  + "../..//pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of responses to unknown requests
    received. A response to an unknown request is a
```

```
        response whose RP object does not contain the
        request ID of any request that is currently
        outstanding on the session.";
    }
    leaf num-req-rcvd-unknown {
        when "../..//pcep:role = 'pcc'"
        + "or "
        + "../..//pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of unknown requests that have been
            received. An unknown request is a request
            whose RP object contains a request ID of zero.";
    }
    container svec {
        if-feature "pcep:svec";
        description
            "If synchronized path computation is supported";
        leaf num-svec-sent {
            when "../..//..//pcep:role = 'pce'"
            + "or "
            + "../..//..//pcep:role = 'pcc-and-pce'" {
                description
                    "Valid for PCEP Peer as PCE";
            }
            type yang:counter32;
            description
                "The number of SVEC objects sent in PCReq messages.
                An SVEC object represents a set of synchronized
                requests.";
        }
        leaf num-svec-req-sent {
            when "../..//..//pcep:role = 'pce'"
            + "or "
            + "../..//..//pcep:role = 'pcc-and-pce'" {
                description
                    "Valid for PCEP Peer as PCE";
            }
            type yang:counter32;
            description
                "The number of requests sent that appeared in one
                or more SVEC objects.";
        }
        leaf num-svec-rcvd {
            when "../..//..//pcep:role = 'pcc'"
```

```
        + "or "
        + "../.../pcep:role = 'pcc-and-pce'" {
    description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
        "The number of SVEC objects received in PCReq
        messages. An SVEC object represents a set of
        synchronized requests.";
    }
    leaf num-svec-req-rcvd {
        when "../.../pcep:role = 'pcc'"
        + "or "
        + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
            "The number of requests received that appeared
            in one or more SVEC objects.";
    }
}
//svec
container stateful {
    if-feature "pcep:stateful";
    description
        "Stateful PCE related statistics";
    leaf num-pcrpt-sent {
        when "../.../pcep:role = 'pce'"
        + "or "
        + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of PCRpt messages sent.";
    }
    leaf num-pcrpt-rcvd {
        when "../.../pcep:role = 'pcc'"
        + "or "
        + "../.../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
    }
}
```

```
    description
      "The number of PCRpt messages received.";
  }
  leaf num-pcupd-sent {
    when "../.../pcep:role = 'pcc'"
      + "or "
      + "../.../pcep:role = 'pcc-and-pce'" {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
    description
      "The number of PCUpd messages sent.";
  }
  leaf num-pcupd-rcvd {
    when "../.../pcep:role = 'pce'"
      + "or "
      + "../.../pcep:role = 'pcc-and-pce'" {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of PCUpd messages received.";
  }
  leaf num-rpt-sent {
    when "../.../pcep:role = 'pce'"
      + "or "
      + "../.../pcep:role = 'pcc-and-pce'" {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of LSP Reports sent. A LSP report
        corresponds 1:1 with an LSP object in a PCRpt
        message. This might be greater than
        num-pcrpt-sent because multiple reports can
        be batched into a single PCRpt message.";
  }
  leaf num-rpt-rcvd {
    when "../.../pcep:role = 'pcc'"
      + "or "
      + "../.../pcep:role = 'pcc-and-pce'" {
      description
        "Valid for PCEP Peer as PCC";
    }
    type yang:counter32;
```



```
description
  "The number of LSP Reports received. A LSP report
  corresponds 1:1 with an LSP object in a PCRpt
  message.
  This might be greater than num-pcrpt-rcvd because
  multiple reports can be batched into a single
  PCRpt message.";
}
leaf num-rpt-rcvd-error-sent {
  when "../../pcep:role = 'pcc'"
  + "or "
  + "../../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of reports of LSPs received that were
    responded by the local PCEP entity by sending a
    PCErr message.";
}
leaf num-upd-sent {
  when "../../pcep:role = 'pcc'"
  + "or "
  + "../../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of LSP updates sent. A LSP update
    corresponds 1:1 with an LSP object in a PCUpd
    message. This might be greater than
    num-pcupd-sent because multiple updates can
    be batched into a single PCUpd message.";
}
leaf num-upd-rcvd {
  when "../../pcep:role = 'pce'"
  + "or "
  + "../../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of LSP Updates received. A LSP update
    corresponds 1:1 with an LSP object in a PCUpd
    message."
```

```
        This might be greater than num-pcupd-rcvd because
        multiple updates can be batched into a single
        PCUpd message.";
    }
    leaf num-upd-rcvd-unknown {
        when "../../pcep:role = 'pce'"
            + "or "
            + "../../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of updates to unknown LSPs
            received. An update to an unknown LSP is a
            update whose LSP object does not contain the
            PLSP-ID of any LSP that is currently
            present.";
    }
    leaf num-upd-rcvd-undelagated {
        when "../../pcep:role = 'pce'"
            + "or "
            + "../../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of updates to not delegated LSPs
            received. An update to an undelagated LSP is a
            update whose LSP object does not contain the
            PLSP-ID of any LSP that is currently
            delegated to current PCEP session.";
    }
    leaf num-upd-rcvd-error-sent {
        when "../../pcep:role = 'pce'"
            + "or "
            + "../../pcep:role = 'pcc-and-pce'" {
            description
                "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
            "The number of updates to LSPs received that were
            responded by the local PCEP entity by sending a
            PCErr message.";
    }
    container initiation {
```

```
if-feature "pcep:pce-initiated";
description
  "PCE-Initiated related statistics";
leaf num-pcinitiate-sent {
  when "../.../.../pcep:role = 'pcc'"
    + "or "
    + "../.../.../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of PCInitiate messages sent.";
}
leaf num-pcinitiate-rcvd {
  when "../.../.../pcep:role = 'pce'"
    + "or "
    + "../.../.../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
  type yang:counter32;
  description
    "The number of PCInitiate messages received.";
}
leaf num-initiate-sent {
  when "../.../.../pcep:role = 'pcc'"
    + "or "
    + "../.../.../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  type yang:counter32;
  description
    "The number of LSP Initiation sent via PCE.
    A LSP initiation corresponds 1:1 with an LSP
    object in a PCInitiate message. This might be
    greater than num-pcinitiate-sent because
    multiple initiations can be batched into a
    single PCInitiate message.";
}
leaf num-initiate-rcvd {
  when "../.../.../pcep:role = 'pce'"
    + "or "
    + "../.../.../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCE";
  }
}
```

```
    type yang:counter32;
    description
      "The number of LSP Initiation received from
      PCE. A LSP initiation corresponds 1:1 with
      an LSP object in a PCInitiate message. This
      might be greater than num-pcinitiate-rcvd
      because multiple initiations can be batched
      into a single PCInitiate message.";
  }
  leaf num-initiate-rcvd-error-sent {
    when "../.../.../pcep:role = 'pce'"
      + "or "
      + "../.../.../pcep:role = 'pcc-and-pce'" {
      description
        "Valid for PCEP Peer as PCE";
    }
    type yang:counter32;
    description
      "The number of initiations of LSPs received
      that were responded by the local PCEP entity
      by sending a PCErr message.";
  }
}
//initiation
}
//stateful
container path-key {
  when "../.../pcep:role = 'pcc'"
    + "or "
    + "../.../pcep:role = 'pcc-and-pce'" {
    description
      "Valid for PCEP Peer as PCC";
  }
  if-feature "pcep:path-key";
  description
    "If Path-Key is supported";
  leaf num-unknown-path-key {
    type yang:counter32;
    description
      "The number of attempts to expand an unknown
      path-key.";
  }
  leaf num-exp-path-key {
    type yang:counter32;
    description
      "The number of attempts to expand an expired
      path-key.";
  }
}
```

```
    leaf num-dup-path-key {
      type yang:counter32;
      description
        "The number of duplicate attempts to expand same
        path-key.";
    }
    leaf num-path-key-no-attempt {
      type yang:counter32;
      description
        "The number of expired path-keys with no attempt to
        expand it.";
    }
  }
  //path-key
}

//pcep-stats
/*
 * Augment modules to add statistics
 */

augment "/pcep:pcep/pcep:entity/pcep:peers/pcep:peer" {
  description
    "Augmenting the statistics";
  leaf num-sess-setup-ok {
    type yang:counter32;
    config false;
    description
      "The number of PCEP sessions successfully established with
      the peer, including any current session. This counter is
      incremented each time a session with this peer is
      successfully established.";
  }
  leaf num-sess-setup-fail {
    type yang:counter32;
    config false;
    description
      "The number of PCEP sessions with the peer
      that have been attempted but failed
      before being fully established. This
      counter is incremented each time a
      session retry to this peer fails.";
  }
  container pcep-stats {
    config false;
    description
      "The container for all statistics at peer level.";
    uses pcep-stats {

```

```
        description
        "Since PCEP sessions can be ephemeral, the peer statistics
        tracks a peer even when no PCEP session currently exists
        to that peer. The statistics contained are an aggregate of
        the statistics for all successive sessions to that peer.";
    }
    leaf num-req-sent-closed {
        when "../pcep:role = 'pce'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
            "Valid for PCEP Peer as PCE";
        }
        type yang:counter32;
        description
        "The number of requests that were sent to the peer and
        implicitly cancelled when the session they were sent
        over was closed.";
    }
    leaf num-req-rcvd-closed {
        when "../pcep:role = 'pcc'"
        + "or "
        + "../pcep:role = 'pcc-and-pce'" {
            description
            "Valid for PCEP Peer as PCC";
        }
        type yang:counter32;
        description
        "The number of requests that were received from the peer
        and implicitly cancelled when the session they were
        received over was closed.";
    }
}
//pcep-stats
}

//augment

augment "/pcep:pcep/pcep:entity/pcep:peers/pcep:peer/"
+ "pcep:sessions/pcep:session" {
    description
    "Augmenting the statistics";
    container pcep-stats {
        description
        "The container for all statistics at session level.";
        uses pcep-stats {
            description
            "The statistics contained are for the current sessions to
```

```
        that peer. These are lost when the session goes down.";
    }
}

//augment

rpc statistics-reset {
  description
    "Reset statistics collected.";
  input {
    choice peer-or-all {
      description
        "Resets statistics for a particular peer or
        all";
      case peer {
        leaf peer-addr {
          type leafref {
            path "/pcep:pcep/pcep:entity/pcep:peers"
              + "/pcep:peer/pcep:addr";
          }
          description
            "Reference to peer address";
        }
        description
          "This resets all the statistics collected for
          the peer.";
      }
      case all {
        description
          "This resets all the statistics collected.";
      }
    }
  }
}
}
<CODE ENDS>
```

10. Security Considerations

The YANG modules defined in this document is designed to be accessed via network management protocol such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a pre-configured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the ietf-pcep YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

/pcep/entity/ - configure local parameters, capabilities etc.

/pcep/entity/peers - configure remote peers to setup PCEP session.

Unauthorized access to above list can adversely affect the PCEP session between the local entity and the peers. This may lead to inability to compute new paths, stateful operations on the delegated as well as PCE-initiated LSPs.

Some of the readable data nodes in the ietf-pcep YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/pcep/lsp-db - All the LSPs in the network. Unauthorized access to this could provide the all path and network usage information.

/pcep/path-keys/ - The Confidential Path Segments (CPS) are hidden using path-keys. Unauthorized access to this could leak confidential path information.

Some of the RPC operations in the ietf-pcep YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

trigger-resync - trigger resynchronization with the PCE.
Unauthorized access to this could force a PCEP session into continuous state synchronization.

The actual authentication key data (whether locally specified or part of a key-chain) is sensitive and needs to be kept secret from unauthorized parties; compromise of the key data would allow an attacker to forge PCEP traffic that would be accepted as authentic, potentially compromising the TE domain.

The model describes several notifications, implementations must rate-limit the generation of these notifications to avoid creating significant notification load. Otherwise, this notification load may have some side effects on the system stability and may be exploited as an attack vector.

Further, this document also include another YANG module (called `ietf-pcep-stats`) for maintaining the statistics by augmenting the `ietf-pcep` YANG module. There are no data nodes defined in this module which are writable/creatable/deletable (i.e., `config true`). The readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. The statistics could provide information related to the current usage patterns of the network. It is thus important to control read access (e.g., via `get`, `get-config`, or notification).

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`statistics-reset` - The RPC is used to reset statistics.
 Unauthorized reset could impact monitoring.

11. IANA Considerations

This document request the IANA to register two URIs in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registrations are requested -

URI:	<code>urn:ietf:params:xml:ns:yang:ietf-pcep</code>
Registrant Contact:	The IESG
XML:	N/A; the requested URI is an XML namespace.
URI:	<code>urn:ietf:params:xml:ns:yang:ietf-pcep-stats</code>
Registrant Contact:	The IESG
XML:	N/A; the requested URI is an XML namespace.

This document request the IANA to register two YANG modules in the "YANG Module Names" registry [RFC6020], as follows -

Name: ietf-pcep
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix: pcep
Reference: [This.I-D]

Name: ietf-pcep-stats
Namespace: urn:ietf:params:xml:ns:yang:ietf-pcep-stats
Prefix: pcep-stats
Reference: [This.I-D]

12. Implementation Status

[Note to the RFC Editor - remove this section before publication, as well as remove the reference to RFC 7942.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

Currently, there are no known implementations of the YANG Module as specified.

13. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. We would like to thank the authors of aforementioned documents.

Thanks to Mahesh Jethmalani for the Yang Doctor review.

Thanks to Martin Bjorklund and Tom Petch for providing guidance regarding the Yang guidelines.

14. References

14.1. Normative References

- [I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for IS-IS Protocol", Work in Progress, Internet-Draft, draft-ietf-isis-yang-isis-cfg-42, 15 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-isis-yang-isis-cfg-42.txt>>.
- [I-D.ietf-netconf-tls-client-server]
Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tls-client-server-26, 14 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-netconf-tls-client-server-26.txt>>.
- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, J., Chen, I., and A. Lindem, "YANG Data Model for OSPF Protocol", Work in Progress, Internet-Draft, draft-ietf-ospf-yang-29, 17 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-ospf-yang-29.txt>>.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. D. Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-28, 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-te-28.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5088] Le Roux, J.L., Ed., Vasseur, J.P., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <<https://www.rfc-editor.org/info/rfc5088>>.

- [RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <<https://www.rfc-editor.org/info/rfc5089>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, DOI 10.17487/RFC5520, April 2009, <<https://www.rfc-editor.org/info/rfc5520>>.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<https://www.rfc-editor.org/info/rfc5541>>.
- [RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <<https://www.rfc-editor.org/info/rfc5557>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.

- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <<https://www.rfc-editor.org/info/rfc8231>>.
- [RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <<https://www.rfc-editor.org/info/rfc8232>>.
- [RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <<https://www.rfc-editor.org/info/rfc8253>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [RFC8306] Zhao, Q., Dhody, D., Ed., Palleti, R., and D. King, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 8306, DOI 10.17487/RFC8306, November 2017, <<https://www.rfc-editor.org/info/rfc8306>>.

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8408] Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages", RFC 8408, DOI 10.17487/RFC8408, July 2018, <<https://www.rfc-editor.org/info/rfc8408>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <<https://www.rfc-editor.org/info/rfc8664>>.
- [RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <<https://www.rfc-editor.org/info/rfc8697>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [RFC8779] Margaria, C., Ed., Gonzalez de Dios, O., Ed., and F. Zhang, Ed., "Path Computation Element Communication Protocol (PCEP) Extensions for GMPLS", RFC 8779, DOI 10.17487/RFC8779, July 2020, <<https://www.rfc-editor.org/info/rfc8779>>.

14.2. Informative References

[IANA-OSPF]

"Path Computation Element (PCE) Capability Flags in Open Shortest Path First v2 (OSPFv2) Parameters",
<<https://www.iana.org/assignments/ospfv2-parameters/ospfv2-parameters.xhtml#ospfv2-parameters-14>>.

[IANA-PCEP]

"ASSOCIATION Type Field in Path Computation Element Protocol (PCEP) Numbers",
<<https://www.iana.org/assignments/pcep/pcep.xhtml#association-type-field>>.

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006,
<<https://www.rfc-editor.org/info/rfc4655>>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008,
<<https://www.rfc-editor.org/info/rfc5246>>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014,
<<https://www.rfc-editor.org/info/rfc7420>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016,
<<https://www.rfc-editor.org/info/rfc7942>>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018,
<<https://www.rfc-editor.org/info/rfc8342>>.

Appendix A. The Full PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker. The tree depth in the tree is set to 10.

```

module: ietf-pcep
+--rw pcep!
  +--rw entity
    +--rw addr inet:ip-address-no-zone
    +--rw enabled? boolean
    +--rw role pcep-role
    +--rw description? string
    +--rw speaker-entity-id? string {sync-opt}?
    +--rw admin-status? boolean
    +--ro index? uint32
    +--ro oper-status? pcep-oper-status
    +--rw domains
      +--rw domains* [domain-type domain-info]
        +--rw domain-type identityref
        +--rw domain-info domain
    +--rw capability
      +--rw capability? bits
      +--rw pce-initiated? boolean {pce-initiated}?
      +--rw include-db-ver? boolean {stateful,sync-opt}?
      +--rw trigger-resync? boolean {stateful,sync-opt}?
      +--rw trigger-initial-sync? boolean {stateful,sync-opt}?
      +--rw incremental-sync? boolean {stateful,sync-opt}?
      +--rw sr {sr}?
        +--rw enabled? boolean
        +--rw msd-limit? boolean
        +--rw nai? boolean
    +--rw msd? uint8 {sr}?
    +--rw pce-info
      +--rw scope
        +--rw path-scope? bits
        +--rw intra-area-pref? uint8
        +--rw inter-area-pref? uint8
        +--rw inter-as-pref? uint8
        +--rw inter-layer-pref? uint8
      +--rw neighbour-domains
        +--rw domains* [domain-type domain-info]
          +--rw domain-type identityref
          +--rw domain-info domain
      +--rw path-key {path-key}?
        +--rw enabled? boolean
        +--rw discard-timer? uint32
        +--rw reuse-time? uint32
        +--rw pce-id? inet:ip-address-no-zone
    +--rw connect-timer? uint16
    +--rw connect-max-retry? uint32
    +--rw init-back-off-timer? uint16
    +--rw max-back-off-timer? uint32
    +--ro open-wait-timer? uint16

```



```

+---ro keep-wait-timer?          uint16
+---rw keepalive-timer?         uint8
+---rw dead-timer?              uint8
+---rw allow-negotiation?       boolean
+---rw max-keepalive-timer?     uint8
+---rw max-dead-timer?         uint8
+---rw min-keepalive-timer?     uint8
+---rw min-dead-timer?         uint8
+---rw sync-timer?              uint16 {svec}?
+---rw request-timer?           uint16
+---rw max-sessions?            uint32
+---rw max-unknown-reqs?        uint32
+---rw max-unknown-msgs?        uint32
+---rw pcep-notification-max-rate uint32
+---rw stateful-parameter {stateful}?
|   +---rw state-timeout?        uint32
|   +---rw redelegation-timeout? uint32
|   +---rw rpt-non-pcep-lsp?     boolean
+---rw of-list {objective-function}?
|   +---rw objective-function* [of]
|       +---rw of identityref
+---ro lsp-db {stateful}?
|   +---ro db-ver?                uint64 {sync-opt}?
|   +---ro association-list*
|       [type id source global-source extended-id]
|       {association}?
|       +---ro type                identityref
|       +---ro id                  uint16
|       +---ro source              inet:ip-address-no-zone
|       +---ro global-source       uint32
|       +---ro extended-id        string
|       +---ro lsp* [plsp-id pcc-id lsp-id]
|           +---ro plsp-id        -> /pcep/entity/lsp-db/lsp/plsp-id
|           +---ro pcc-id         -> /pcep/entity/lsp-db/lsp/pcc-id
|           +---ro lsp-id         -> /pcep/entity/lsp-db/lsp/lsp-id
+---ro lsp* [plsp-id pcc-id lsp-id]
|   +---ro plsp-id                uint32
|   +---ro pcc-id                 inet:ip-address-no-zone
|   +---ro source?                inet:ip-address-no-zone
|   +---ro destination?           inet:ip-address-no-zone
|   +---ro tunnel-id?             uint16
|   +---ro lsp-id                 uint16
|   +---ro extended-tunnel-id?    inet:ip-address-no-zone
|   +---ro admin-state?           boolean
|   +---ro operational-state?     operational-state
|   +---ro delegated
|       +---ro enabled?           boolean
|       +---ro peer?             -> /pcep/entity/peers/peer/addr

```

```

|   +--ro srp-id?      uint32
+--ro initiation {pce-initiated}?
|   +--ro enabled?    boolean
|   +--ro peer?       -> /pcep/entity/peers/peer/addr
+--ro symbolic-path-name? string
+--ro last-error?     identityref
+--ro pst?            identityref
+--ro association-list*
|   [type id source global-source extended-id]
|   {association}?
+--ro type
|   -> /pcep/entity/lsp-db/association-list/type
+--ro id
|   -> /pcep/entity/lsp-db/association-list/id
+--ro source          leafref
+--ro global-source   leafref
+--ro extended-id     leafref
+--ro path-keys {path-key}?
+--ro path-keys* [path-key]
+--ro path-key        uint16
+--ro cps
+--ro explicit-route-objects* [index]
+--ro index                                uint32
+--ro (type)?
+--: (numbered-node-hop)
|   +--ro numbered-node-hop
|   |   +--ro node-id      te-node-id
|   |   +--ro hop-type?    te-hop-type
+--: (numbered-link-hop)
|   +--ro numbered-link-hop
|   |   +--ro link-tp-id    te-tp-id
|   |   +--ro hop-type?    te-hop-type
|   |   +--ro direction?   te-link-direction
+--: (unnumbered-link-hop)
|   +--ro unnumbered-link-hop
|   |   +--ro link-tp-id    te-tp-id
|   |   +--ro node-id      te-node-id
|   |   +--ro hop-type?    te-hop-type
|   |   +--ro direction?   te-link-direction
+--: (as-number)
|   +--ro as-number-hop
|   |   +--ro as-number    inet:as-number
|   |   +--ro hop-type?    te-hop-type
+--: (label)
|   +--ro label-hop
|   |   +--ro te-label
|   |   ...
+--ro pcc-original?    -> /pcep/entity/peers/peer/addr

```

```

    +--ro req-id?          uint32
    +--ro retrieved?       boolean
    +--ro pcc-retrieved?   -> /pcep/entity/peers/peer/addr
    +--ro creation-time?   yang:timestamp
    +--ro discard-time?    uint32
    +--ro reuse-time?      uint32
+--rw peers
  +--rw peer* [addr]
    +--rw addr              inet:ip-address-no-zone
    +--rw role              pcep-role
    +--rw description?     string
    +--rw domains
      +--rw domains* [domain-type domain-info]
        +--rw domain-type  identityref
        +--rw domain-info  domain
    +--rw capability
      +--rw capability?    bits
      +--rw pce-initiated? boolean
      |    {pce-initiated}?
      +--rw include-db-ver? boolean
      |    {stateful, sync-opt}?
      +--rw trigger-resync? boolean
      |    {stateful, sync-opt}?
      +--rw trigger-initial-sync? boolean
      |    {stateful, sync-opt}?
      +--rw incremental-sync? boolean
      |    {stateful, sync-opt}?
      +--rw sr {sr}?
        +--rw enabled?    boolean
        +--rw msd-limit?  boolean
        +--rw nai?        boolean
    +--rw msd?              uint8 {sr}?
  +--rw pce-info
    +--rw scope
      +--rw path-scope?    bits
      +--rw intra-area-pref? uint8
      +--rw inter-area-pref? uint8
      +--rw inter-as-pref?  uint8
      +--rw inter-layer-pref? uint8
    +--rw neighbour-domains
      +--rw domains* [domain-type domain-info]
        +--rw domain-type  identityref
        +--rw domain-info  domain
    +--rw delegation-pref?  uint8 {stateful}?
  +--rw auth
    +--rw (auth-type-selection)?
    |   +--:(auth-key-chain)
    |   |   +--rw key-chain?

```

```

|           key-chain:key-chain-ref
+--:(auth-key)
|   +--rw crypto-algorithm           identityref
|   +--rw (key-string-style)?
|       +--:(keystring)
|           |   +--rw keystring?       string
|           +--:(hexadecimal)
|               {key-chain:hex-key-string}?
|               +--rw hexadecimal-string?
|                   yang:hex-string
+--:(auth-tls) {tls}?
|   +--rw (role)?
|       +--:(server)
|           |   +--rw tls-server
|           ...
|       +--:(client)
|           +--rw tls-client
|           ...
+--ro discontinuity-time?      yang:timestamp
+--ro initiate-session?       boolean
+--ro session-exists?         boolean
+--ro session-up-time?        yang:timestamp
+--ro session-fail-time?      yang:timestamp
+--ro session-fail-up-time?   yang:timestamp
+--ro sessions
|   +--ro session* [initiator]
|       +--ro initiator           pcep-initiator
|       +--ro role?
|           |   -> /pcep/entity/role
|       +--ro state-last-change?  yang:timestamp
|       +--ro state?
|           |   pcep-sess-state
|       +--ro session-creation?   yang:timestamp
|       +--ro connect-retry?      yang:counter32
|       +--ro local-id?           uint8
|       +--ro remote-id?          uint8
|       +--ro keepalive-timer?    uint8
|       +--ro peer-keepalive-timer? uint8
|       +--ro dead-timer?         uint8
|       +--ro peer-dead-timer?    uint8
|       +--ro ka-hold-time-rem?   uint8
|       +--ro overloaded?         boolean
|       +--ro overloaded-timestamp? yang:timestamp
|       +--ro overload-time?      uint32
|       +--ro peer-overloaded?    boolean
|       +--ro peer-overloaded-timestamp? yang:timestamp
|       +--ro peer-overload-time? uint32
|       +--ro lspdb-sync?         sync-state

```

```

|         {stateful}?
+---ro recv-db-ver?                               uint64
|         {stateful,sync-opt}?
+---ro of-list {objective-function}?
|   +---ro objective-function* [of]
|     +---ro of      identityref
+---ro pst-list
|   +---ro path-setup-type* [pst]
|     +---ro pst      identityref
+---ro assoc-type-list {association}?
|   +---ro assoc-type* [at]
|     +---ro at      identityref
+---ro speaker-entity-id?                         string
|         {sync-opt}?

```

rpccs:

```

+---x trigger-resync {stateful,sync-opt}?
+---w input
+---w pcc?    -> /pcep/entity/peers/peer/addr

```

notifications:

```

+---n pcep-session-up
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?
|   |       -> /pcep/entity/peers/peer/sessions/session/initiator
|   +---ro state-last-change?  yang:timestamp
|   +---ro state?              pcep-sess-state
+---n pcep-session-down
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?  pcep-initiator
|   +---ro state-last-change?  yang:timestamp
|   +---ro state?              pcep-sess-state
+---n pcep-session-local-overload
|   +---ro peer-addr?          -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?
|   |       -> /pcep/entity/peers/peer/sessions/session/initiator
|   +---ro overloaded?         boolean
|   +---ro overloaded-timestamp? yang:timestamp
|   +---ro overload-time?      uint32
+---n pcep-session-local-overload-clear
|   +---ro peer-addr?
|   |       -> /pcep/entity/peers/peer/addr
|   +---ro overloaded?         boolean
|   +---ro overloaded-clear-timestamp? yang:timestamp
+---n pcep-session-peer-overload
|   +---ro peer-addr?
|   |       -> /pcep/entity/peers/peer/addr
|   +---ro session-initiator?

```

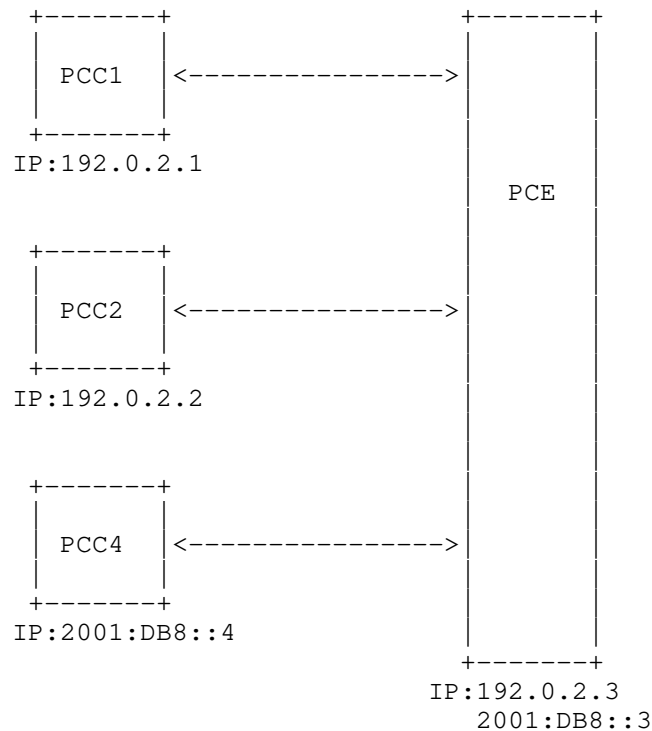
```

| |           -> /pcep/entity/peers/peer/sessions/session/initiator
| +--ro peer-overloaded?          boolean
| +--ro peer-overloaded-timestamp? yang:timestamp
| +--ro peer-overload-time?       uint32
+---n pcep-session-peer-overload-clear
  +--ro peer-addr?
  |       -> /pcep/entity/peers/peer/addr
  +--ro peer-overloaded?          boolean
  +--ro peer-overloaded-clear-timestamp? yang:timestamp

```

Appendix B. Example

The example below provide an overview of PCEP peer session informations and LSP-DB in the Yang Module.



```

at PCE:
{
  "entity": [
    {
      "addr": "192.0.2.3",

```

```
"oper-status": "oper-status-up",
"role": "pce",
"capability": {
  "stateful": {
    "enabled": true
  }
}
"lsp-db" : [
  "lsp" : {
    "plsp-id": 3,
    "pcc-id" : "192.0.2.1",
    "source": "192.0.2.1",
    "destination": "192.0.2.4"
    "tunnel-id": 16,
    "lsp-id": 3,
    "extended-tunnel-id": 0,
    "oper-status": "oper-status-up",
    "delegated": true,
    "symbolic-path-name": "iewauh",
  },
  "lsp" : {
    "plsp-id": 4,
    "pcc-id" : "192.0.2.2",
    "source": "192.0.2.2",
    "destination": "192.0.2.5"
    "tunnel-id": 17,
    "lsp-id": 4
    "extended-tunnel-id": 0,
    "oper-status": "oper-status-up",
    "delegated": true,
    "symbolic-path-name": "iewauhiewauh",
    "extended-tunnel-id": 0
  }
]
"peers": [
  {
    "peer": {
      "addr": "192.0.2.1",
      "role": "pcc",

      "capability": {
        "stateful" : {
          "enabled": true,
          "active": yes,
        }
      }
    }
    "sessions": [
      {
```

```

        "session": {
            "initiator": "remote",
            "role": "pcc",
        }
    ]
}
},
{
    "peer": {
        "addr": "192.0.2.2",
        "role": "pcc",

        "capability": {
            "stateful" : {
                "enabled": true,
                "active": true,
            }
        }
        "sessions": [
            {
                "session": {
                    "initiator": "remote",
                    "role": "pcc",
                }
            }
        ]
    }
}
],
},
{
    "addr": "2001:DB8::3",
    "oper-status": "oper-status-up",
    "role": "pce",
    "peers": [
        {
            "peer": {
                "addr": "2001:DB8::4",
                "role": "pcc",
                "sessions": [
                    {
                        "session": {
                            "initiator": "remote",
                            "role": "pcc",
                        }
                    }
                ]
            }
        }
    ]
}

```



```
    ]  
    }  
  }  
]  
}  
}
```

Appendix C. Contributor Addresses

Rohit Pobbathi
Nokia Networks
India

EMail: rohit.pobbathi_ashok@nokia.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: vinods.kumar@huawei.com

Zafar Ali
Cisco Systems
Canada

EMail: zali@cisco.com

Xufeng Liu
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Young Lee
Samsung

EMail: younglee.tx@gmail.com

Udayasree Palle

EMail: udayasreereddy@gmail.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129

P.R.China

Email: zhang.xian@huawei.com

Avantika
ECI Telecom
India

Email: avantika.srm@gmail.com

Shashikanth
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: shashikanth.vh@huawei.com

Authors' Addresses

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore 560066
Karnataka
India

Email: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield
EN2 6BQ
United Kingdom

Email: jonathan.hardwick@metaswitch.com

Vishnu Pavan Beeram
Juniper Networks
India

Email: vbeeram@juniper.net

Jeff Tantsura
Microsoft
United States of America

Email: jefftant.ietf@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2017

V. Beeram
Juniper Networks
T. Saad, Ed.
R. Gandhi
Cisco Systems, Inc.
X. Liu
Jabil
I. Bryskin
Huawei Technologies
H. Shah
Ciena
March 10, 2017

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-07

Abstract

This document defines a YANG data model for the configuration and management of RSVP Protocol. The model covers the building blocks of the RSVP protocol that can be augmented and used by other RSVP extension models such as RVSP extensions to Traffic-Engineering (RSVP-TE). The model covers the configuration, operational state, remote procedural calls, and event notifications data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 11, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Tree Diagram	3
1.3. Prefixes in Data Node Names	4
2. Design Considerations	5
2.1. Module Hierarchy	5
2.2. Data Organization	6
2.3. Configuration Inheritance	7
3. Model Organization	7
3.1. RSVP Base YANG Model	7
3.1.1. Global Data	9
3.1.2. Interface Data	9
3.1.3. Neighbor Data	9
3.1.4. Session Data	9
3.1.5. Tree Diagram	9
3.1.6. YANG Module	14
3.2. RSVP Extended YANG Model	33
3.2.1. Tree Diagram	34
3.2.2. YANG Module	40
4. IANA Considerations	52
5. Security Considerations	52
6. Acknowledgement	52
7. Contributors	53
8. References	53
8.1. Normative References	53
8.2. Informative References	54
Authors' Addresses	55

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of

implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage the RSVP protocol [RFC2205]. This model covers RSVP protocol building blocks that can be augmented and used by other RSVP extension models- such as for signaling RSVP-TE MPLS (or other technology specific) Label Switched Paths (LSP)s.

1.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Tree Diagram

A simplified graphical representation of the data model is presented in each section of the model. The following notations are used for the YANG model data tree representation.

<status> <flags> <name> <opts> <type>

<status> is one of:

- + for current
- x for deprecated
- o for obsolete

<flags> is one of:

- rw for read-write configuration data
- ro for read-only non-configuration data
- x for execution rpcs
- n for notifications

<name> is the name of the node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>

<opts> is one of:

- ? for an optional leaf or node
- ! for a presence container
- * for a leaf-list or list

Brackets [<keys>] for a list's keys

Curly braces {<condition>} for optional feature that make node conditional

Colon : for marking case nodes

Ellipses ("...") subtree contents not shown

Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

<type> is the name of the type for leafs and leaf-lists.

1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-type	ietf-routing-types	XX
key-chain	ietf-key-chain	XX

Table 1: Prefixes and corresponding YANG modules

2. Design Considerations

2.1. Module Hierarchy

The RSVP base YANG module augments the "control-plane-protocol" list in ietf-routing [RFC8022] module with specific RSVP parameters in an "rsvp" container. It also defines an extension identity "rsvp" of base "rt:routing-protocol" to identify the RSVP protocol.

During modeling discussion, some RSVP features are categorized as core to the functionality of the protocol, and hence, are supported by all vendors claiming the support for RSVP. These features' configuration and state were grouped in the RSVP base module.

Other extended RSVP features are categorized as either optional or providing additional knobs to provide better tune basic functionality of the RSVP protocol. The support for extended RSVP features by all vendors was considered optional. Such features were grouped in a separate RSVP extended module.

The augmentation of the RSVP model by other models (e.g. RSVP-TE for MPLS or other technologies) are considered outside the scope of this document and discussed in separate document(s).

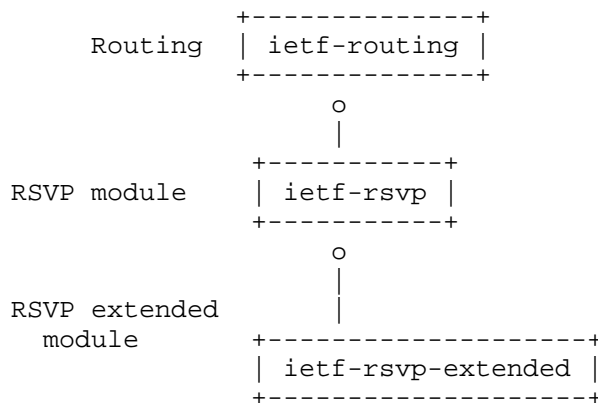


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

The RSVP base model does not aim to be feature complete. The primary intent is to cover a set of standard core features (listed below) that are commonly in use.

- o Authentication ([RFC2747])
- o Refresh Reduction ([RFC2961])
- o Hellos ([RFC3209])
- o Graceful Restart ([RFC3473], [RFC5063])

The extended RSVP YANG model covers non-basic configuration(s) for RSVP feature(s) as well as optional RSVP feature that are not a must for basic RSVP operation.

2.2. Data Organization

Throughout the model, the approach described in [I-D.openconfig-netmod-opstate] is adopted to represent data pertaining to configuration intended state, applied state and derived state data elements. Each container in the model hold a "config" and "state" sub-container.

The "config" sub-container is used to represent the intended configurable parameters, and the state sub-container is used to represent both the applied configurable parameters and any derived state, such as counters or statistical information.

The decision to use this approach was made to better align with the MPLS consolidated model in [I-D.openconfig-mpls-consolidated-model], and maximize reusability of groupings defined in this document and allow for possible convergence between the two models.

The approach described in [I-D.openconfig-netmod-opstate] allows for modeling the respective intended and applied configuration and derived state. The state data can be categorized into one of the following:

- o State corresponding to applied configuration
- o State corresponding to derived state, counters, stats, etc.

Pure state data (for example, protocol derived data) is placed inside the "state" sub-container, as shown in Figure 2.

2.3. Configuration Inheritance

The defined data model supports configuration inheritance for neighbors, and interfaces. Data elements defined in the main container (e.g. the container that encompasses the list of interfaces, or neighbors) are assumed to apply equally to all elements of the list, unless overridden explicitly for a certain element (e.g. interface). Vendors are expected to augment the above container(s) to provide the list of inheritance command for their implementations.

3. Model Organization

This document divides the RSVP model into two modules: the RSVP base and extended. Each module covers the configuration, state, notification and RPCs of data. The relationship between the different modules is depicted in Figure 1.

3.1. RSVP Base YANG Model

This section describes the RSVP base YANG data model. The container "rsvp" is the top level container in this data model. The presence of this container enables the RSVP protocol functionality.

Data for such state is contained under the respective "state" sub-container of the intended object (e.g. interface) as shown in Figure 2.

```
module: ietf-rsvp
  +--rw rsvp!
    +--rw globals
```

```

    +-- rw config
       <<intended configuration>>
    .
    +-- ro state
       <<applied configuration>>
       <<derived state associated with the tunnel>>
    .
    .
+--rw interfaces
   +-- rw config
      <<intended configuration>>
   .
   +-- ro state
      <<applied configuration>>
      <<derived state associated with the tunnel>>
   .
   .
+--rw neighbors
   +-- rw config
      <<intended configuration>>
   .
   +-- ro state
      <<applied configuration>>
      <<derived state associated with the tunnel>>
   .
   .
+--rw sessions
   +-- rw config
      <<intended configuration>>
   .
   +-- ro state
      <<applied configuration>>
      <<derived state associated with the tunnel>>
   .
rpcs:
  +--x global-rpc
  +--x interfaces-rpc
  +--x neighbors-rpc
  +--x sessions-rpc
notifications:
  +--n global-notif
  +--n interfaces-notif
  +--n neighbors-notif
  +--n sessions-notif

```

Figure 2: RSVP high-level tree model view

The following subsections provide overview of the parts of the model pertaining to configuration and state data.

Configuration and state data are organized into those applicable globally (node scope), per interfaces, per neighbors, or per session.

3.1.1. Global Data

This branch of the data model covers global configuration and states that control RSVP protocol behavior.

3.1.2. Interface Data

This branch of the data model covers configuration and state elements relevant to one or all RSVP interfaces. Any data configuration applied at the "interfaces" container level are equally applicable to all interfaces - unless overridden by explicit configuration under a specific interface.

3.1.3. Neighbor Data

This branch of the data model covers configuration of elements relevant to RSVP neighbors. This would be discussed in detail in future revisions.

3.1.4. Session Data

This branch of the data model covers configuration of elements relevant to RSVP sessions. This would be discussed in detail in future revisions.

3.1.5. Tree Diagram

```
module: ietf-rsvp
  augment
    /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
      +--rw rsvp!
        +--rw globals
          +--rw sessions
            +--ro session* [local-index]
              +--ro local-index    -> ../state/local-index
              +--ro state
                +--ro local-index?    uint64
                +--ro destination-port? inet:port-number
                +--ro source?          inet:ip-address
                +--ro destination?     inet:ip-address
                +--ro session-name?    string
                +--ro session-state?   enumeration
```

```

    +--ro session-type?      identityref
    +--ro psbs
    |   +--ro psb*
    |   |   +--ro source-port?    inet:port-number
    |   |   +--ro expires-in?     uint32
    +--ro rsbs
    |   +--ro rsb*
    |   |   +--ro source-port?    inet:port-number
    |   |   +--ro reservation-style? identityref
    |   |   +--ro expires-in?     uint32
+--rw statistics
  +--ro state
    +--ro messages
    |   +--ro ack-sent?          yang:counter64
    |   +--ro ack-received?      yang:counter64
    |   +--ro bundle-sent?       yang:counter64
    |   +--ro bundle-received?   yang:counter64
    |   +--ro hello-sent?        yang:counter64
    |   +--ro hello-received?    yang:counter64
    |   +--ro integrity-challenge-sent? yang:counter64
    |   +--ro integrity-challenge-received? yang:counter64
    |   +--ro integrity-response-sent? yang:counter64
    |   +--ro integrity-response-received? yang:counter64
    |   +--ro notify-sent?       yang:counter64
    |   +--ro notify-received?   yang:counter64
    |   +--ro path-sent?         yang:counter64
    |   +--ro path-received?     yang:counter64
    |   +--ro path-err-sent?     yang:counter64
    |   +--ro path-err-received? yang:counter64
    |   +--ro path-tear-sent?    yang:counter64
    |   +--ro path-tear-received? yang:counter64
    |   +--ro resv-sent?         yang:counter64
    |   +--ro resv-received?     yang:counter64
    |   +--ro resv-confirm-sent? yang:counter64
    |   +--ro resv-confirm-received? yang:counter64
    |   +--ro resv-err-sent?     yang:counter64
    |   +--ro resv-err-received? yang:counter64
    |   +--ro resv-tear-sent?    yang:counter64
    |   +--ro resv-tear-received? yang:counter64
    |   +--ro summary-refresh-sent? yang:counter64
    |   +--ro summary-refresh-received? yang:counter64
    |   +--ro unknown-messages-received? yang:counter64
    +--ro packets
    |   +--ro sent?              yang:counter64
    |   +--ro received?         yang:counter64
    +--ro errors
    |   +--ro authenticate?      yang:counter64
    |   +--ro checksum?          yang:counter64

```

```

|         +--ro packet-len?      yang:counter64
+--rw graceful-restart
|   +--rw config
|   |   +--rw enabled?    boolean
+--ro state
|   +--ro enabled?    boolean
+--rw interfaces
|   +--rw config
+--ro state
+--rw refresh-reduction
|   +--rw config
|   |   +--rw enabled?    boolean
+--ro state
|   +--ro enabled?    boolean
+--rw hellos
|   +--rw config
|   |   +--rw enabled?    boolean
+--ro state
|   +--ro enabled?    boolean
+--rw authentication
|   +--rw config
|   |   +--rw enabled?          boolean
|   |   +--rw authentication-key? string
|   |   +--rw crypto-algorithm  identityref
+--ro state
|   +--ro enabled?          boolean
|   +--ro authentication-key? string
|   +--ro crypto-algorithm  identityref
+--rw statistics
|   +--ro state
|   |   +--ro messages
|   |   |   +--ro ack-sent?          yang:counter64
|   |   |   +--ro ack-received?      yang:counter64
|   |   |   +--ro bundle-sent?       yang:counter64
|   |   |   +--ro bundle-received?   yang:counter64
|   |   |   +--ro hello-sent?        yang:counter64
|   |   |   +--ro hello-received?    yang:counter64
|   |   |   +--ro integrity-challenge-sent? yang:counter64
|   |   |   +--ro integrity-challenge-received? yang:counter64
|   |   |   +--ro integrity-response-sent? yang:counter64
|   |   |   +--ro integrity-response-received? yang:counter64
|   |   |   +--ro notify-sent?       yang:counter64
|   |   |   +--ro notify-received?   yang:counter64
|   |   |   +--ro path-sent?         yang:counter64
|   |   |   +--ro path-received?     yang:counter64
|   |   |   +--ro path-err-sent?     yang:counter64
|   |   |   +--ro path-err-received? yang:counter64
|   |   |   +--ro path-tear-sent?    yang:counter64

```

```

| | | | | +--ro path-tear-received?          yang:counter64
| | | | | +--ro resv-sent?                  yang:counter64
| | | | | +--ro resv-received?              yang:counter64
| | | | | +--ro resv-confirm-sent?          yang:counter64
| | | | | +--ro resv-confirm-received?      yang:counter64
| | | | | +--ro resv-err-sent?              yang:counter64
| | | | | +--ro resv-err-received?          yang:counter64
| | | | | +--ro resv-tear-sent?             yang:counter64
| | | | | +--ro resv-tear-received?         yang:counter64
| | | | | +--ro summary-refresh-sent?       yang:counter64
| | | | | +--ro summary-refresh-received?   yang:counter64
| | | | | +--ro unknown-messages-received?  yang:counter64
| | | | +--ro packets
| | | | | +--ro sent?          yang:counter64
| | | | | +--ro received?     yang:counter64
| | | | +--ro errors
| | | | | +--ro authenticate? yang:counter64
| | | | | +--ro checksum?     yang:counter64
| | | | | +--ro packet-len?   yang:counter64
+--rw interface* [interface]
| +--rw interface          if:interface-ref
| +--rw config
| +--ro state
| +--rw refresh-reduction
| | +--rw config
| | | +--rw enabled?    boolean
| | +--ro state
| | | +--ro enabled?    boolean
+--rw hellos
| +--rw config
| | +--rw enabled?    boolean
| +--ro state
| | +--ro enabled?    boolean
+--rw authentication
| +--rw config
| | +--rw enabled?          boolean
| | +--rw authentication-key? string
| | +--rw crypto-algorithm  identityref
| +--ro state
| | +--ro enabled?          boolean
| | +--ro authentication-key? string
| | +--ro crypto-algorithm  identityref
+--rw statistics
| +--ro state
| | +--ro messages
| | | +--ro ack-sent?          yang:counter64
| | | +--ro ack-received?     yang:counter64
| | | +--ro bundle-sent?      yang:counter64

```

```

+---ro bundle-received?          yang:counter64
+---ro hello-sent?               yang:counter64
+---ro hello-received?           yang:counter64
+---ro integrity-challenge-sent?  yang:counter64
+---ro integrity-challenge-received? yang:counter64
+---ro integrity-response-sent?   yang:counter64
+---ro integrity-response-received? yang:counter64
+---ro notify-sent?              yang:counter64
+---ro notify-received?          yang:counter64
+---ro path-sent?                yang:counter64
+---ro path-received?            yang:counter64
+---ro path-err-sent?            yang:counter64
+---ro path-err-received?        yang:counter64
+---ro path-tear-sent?           yang:counter64
+---ro path-tear-received?       yang:counter64
+---ro resv-sent?                yang:counter64
+---ro resv-received?            yang:counter64
+---ro resv-confirm-sent?        yang:counter64
+---ro resv-confirm-received?    yang:counter64
+---ro resv-err-sent?            yang:counter64
+---ro resv-err-received?        yang:counter64
+---ro resv-tear-sent?           yang:counter64
+---ro resv-tear-received?       yang:counter64
+---ro summary-refresh-sent?     yang:counter64
+---ro summary-refresh-received? yang:counter64
+---ro unknown-messages-received? yang:counter64
+---ro packets
|   +---ro sent?                  yang:counter64
|   +---ro received?              yang:counter64
+---ro errors
|   +---ro authenticate?          yang:counter64
|   +---ro checksum?              yang:counter64
|   +---ro packet-len?            yang:counter64
+---rw neighbors
+---rw neighbor* [address]
+---rw address    -> ../config/address
+---rw config
|   +---rw address?    inet:ip-address
+---ro state
+---ro address?          inet:ip-address
+---ro epoch?            uint32
+---ro expiry-time?      uint32
+---ro graceful-restart
|   +---ro enabled?        boolean
|   +---ro local-restart-time? uint32
|   +---ro local-recovery-time? uint32
|   +---ro neighbor-restart-time? uint32
|   +---ro neighbor-recovery-time? uint32

```



```

|   +--ro helper-mode
|   |   +--ro enabled?                               boolean
|   |   +--ro max-helper-restart-time?                uint32
|   |   +--ro max-helper-recovery-time?              uint32
|   |   +--ro neighbor-restart-time-remaining?       uint32
|   |   +--ro neighbor-recovery-time-remaining?      uint32
|   +--ro hello-status?                             enumeration
|   +--ro interface?                                if:interface-ref
|   +--ro neighbor-state?                           enumeration
|   +--ro refresh-reduction-capable?                 boolean
|   +--ro restart-count?                             yang:counter32
|   +--ro restart-time?                              yang:date-and-time

```

Figure 3: RSVP model tree diagram

3.1.6. YANG Module

```

<CODE BEGINS> file "ietf-rsvp@2017-03-10.yang"
module ietf-rsvp {

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp";

  /* Replace with IANA when assigned */
  prefix "rsvp";

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-routing {
    prefix "rt";
  }

  import ietf-key-chain {
    prefix "key-chain";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";

```

```
contact
  "WG Web:    <http://tools.ietf.org/wg/teas/>
  WG List:    <mailto:teas@ietf.org>

  WG Chair:   Lou Berger
              <mailto:lberger@labn.net>

  WG Chair:   Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>

  Editor:     Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>

  Editor:     Tarek Saad
              <mailto:tsaad@cisco.com>

  Editor:     Rakesh Gandhi
              <mailto:rgandhi@cisco.com>

  Editor:     Himanshu Shah
              <mailto:hshah@ciena.com>

  Editor:     Xufeng Liu
              <mailto:Xufeng_Liu@jabil.com>

  Editor:     Xia Chen
              <mailto:jescia.chenxia@huawei.com>

  Editor:     Raqib Jones
              <mailto:raqib@Brocade.com>

  Editor:     Bin Wen
              <mailto:Bin_Wen@cable.comcast.com>";

description
  "This module contains the RSVP YANG data model.";

revision "2017-03-10" {
  description "Latest revision of RSVP yang module.";
  reference "RFC2205";
}

identity rsvp {
  base "rt:routing-protocol";
  description "RSVP protocol";
}

identity rsvp-session-type {
```

```
    description "Base RSVP session type";
  }

  identity rsvp-session-ipv4 {
    base rsvp-session-type;
    description "RSVP IPv4 session type";
  }

  identity rsvp-session-ipv6 {
    base rsvp-session-type;
    description "RSVP IPv4 session type";
  }

  identity reservation-style {
    description "Base identity for reservation style";
  }

  identity reservation-wildcard-filter {
    base reservation-style;
    description "Wildcard-Filter (WF) Style";
    reference "RFC2205";
  }

  identity reservation-fixed-filter {
    base reservation-style;
    description "Fixed-Filter (FF) Style";
    reference "RFC2205";
  }

  identity reservation-shared-explicit {
    base reservation-style;
    description "Shared Explicit (SE) Style";
    reference "RFC2205";
  }

  grouping graceful-restart_config {
    description
      "Base configuration parameters relating to RSVP
      Graceful-Restart";
    leaf enabled {
      type boolean;
      description
        "'true' if RSVP Graceful Restart is enabled.
        'false' if RSVP Graceful Restart is disabled.";
    }
  }

  grouping graceful-restart {
```

```
description
  "RSVP graceful restart parameters grouping";
container graceful-restart {
  description
    "RSVP graceful restart parameters container";
  container config {
    description
      "Configuration parameters for graceful restart
      properties";
    uses graceful-restart_config;
  }
  container state {
    config false;
    description
      "State parameters for graceful restart
      properties";
    uses graceful-restart_config;
  }
}

grouping refresh-reduction_config {
  description
    "Configuration parameters relating to RSVP
    refresh reduction";

  leaf enabled {
    type boolean;
    description
      "'true' if RSVP Refresh Reduction is enabled.
      'false' if RSVP Refresh Reduction is disabled.";
  }
}

grouping refresh-reduction {
  description
    "Top level grouping for RSVP refresh reduction
    parameters";
  container refresh-reduction {
    description
      "Top level container for RSVP refresh reduction
      parameters";
    container config {
      description
        "Configuration parameters relating to
        RSVP refresh reduction parameters";
      uses refresh-reduction_config;
    }
  }
}
```

```
        container state {
            config false;
            description
                "State information associated with RSVP
                refresh reduction parameters";
            uses refresh-reduction_config;
        }
    }
}

grouping authentication_config {
    description
        "Configuration parameters relating to RSVP
        authentication";
    leaf enabled {
        type boolean;
        description
            "'true' if RSVP Authentication is enabled.
            'false' if RSVP Authentication is disabled.";
    }
    leaf authentication-key {
        type string;
        description
            "An authentication key string";
        reference
            "RFC 2747: RSVP Cryptographic Authentication";
    }
    leaf crypto-algorithm {
        type identityref {
            base key-chain:crypto-algorithm;
        }
        mandatory true;
        description
            "Cryptographic algorithm associated with key.";
    }
}

grouping authentication {
    description
        "Top level grouping for RSVP authentication parameters";
    container authentication {
        description
            "Top level container for RSVP authentication
            parameters";
        container config {
            description
                "Configuration parameters relating to
                RSVP authentication parameters";
        }
    }
}
```

```
        uses authentication_config;
    }
    container state {
        config false;
        description
            "State information associated with RSVP
            authentication parameters";
        uses authentication_config;
    }
}

grouping hellos_config {
    description
        "Configuration parameters relating to RSVP
        hellos";
    leaf enabled {
        type boolean;
        description
            "'true' if RSVP Hello is enabled.
            'false' if RSVP Hello is disabled.";
    }
}

grouping hellos {
    description
        "Top level grouping for RSVP hellos parameters";
    container hellos {
        description
            "Top level container for RSVP hello parameters";
        container config {
            description
                "Configuration parameters relating to
                RSVP hello parameters";
            uses hellos_config;
        }
        container state {
            config false;
            description
                "State information associated with RSVP
                hello parameters";
            uses hellos_config;
        }
    }
}

grouping signaling-parameters_config {
    description
```

```
    "Configuration parameters relating to RSVP
    signaling";
}

grouping signaling-parameters {
  description
    "Top level grouping for RSVP signaling parameters";
  container config {
    description
      "Configuration parameters relating to
      RSVP signaling parameters";
    uses signaling-parameters_config;
  }
  container state {
    config false;
    description
      "State information associated with RSVP
      signaling parameters";
    uses signaling-parameters_config;
  }
}

grouping session-attributes_state {
  description
    "Top level grouping for RSVP session properties";
  leaf local-index {
    type uint64;
    description
      "The index used to identify the RSVP session
      on the local network element. This index is
      generated by the device and is unique only
      to the local network element.";
  }
  leaf destination-port {
    type inet:port-number;
    description "RSVP destination port";
    reference "RFC2205";
  }
  leaf source {
    type inet:ip-address;
    description "RSVP source address";
    reference "RFC2205";
  }
  leaf destination {
    type inet:ip-address;
    description "RSVP destination address";
    reference "RFC2205";
  }
}
```

```
leaf session-name {
  type string;
  description
    "The signaled name of this RSVP session.";
}
leaf session-state {
  type enumeration {
    enum "up" {
      description
        "RSVP session is up";
    }
    enum "down" {
      description
        "RSVP session is down";
    }
  }
  description
    "Enumeration of RSVP session states";
}
leaf session-type {
  type identityref {
    base rsvp-session-type;
  }
  description "RSVP session type";
}
container psbs {
  description "Path State Block container";
  list psb {
    description "List of path state blocks";
    leaf source-port {
      type inet:port-number;
      description "RSVP source port";
      reference "RFC2205";
    }
    leaf expires-in {
      type uint32;
      units seconds;
      description "Time to reservation expiry (in seconds)";
    }
  }
}
container rsbs {
  description "Reservation State Block container";
  list rsb {
    description "List of reservation state blocks";
    leaf source-port {
      type inet:port-number;
      description "RSVP source port";
    }
  }
}
```



```
        reference "RFC2205";
    }
    leaf reservation-style {
        type identityref {
            base reservation-style;
        }
        description "RSVP reservation style";
    }
    leaf expires-in {
        type uint32;
        units seconds;
        description "Time to reservation expiry (in seconds)";
    }
}
}

grouping neighbor-attributes {
    description
        "Top level grouping for RSVP neighbor properties";
    container config {
        description
            "Configuration for neighbor properties";
        leaf address {
            type inet:ip-address;
            description
                "Address of RSVP neighbor";
        }
    }
    container state {
        config false;
        description
            "State information associated with RSVP
            neighbor properties";
        uses neighbor-derived_state;
    }
}

grouping packets_state {
    description
        "Packet statistics grouping";
    container packets {
        description
            "Packet statistics container";
        leaf sent {
            type yang:counter64;
            description
                "Packet sent count";
        }
    }
}
```

```
    }

    leaf received {
      type yang:counter64;
      description
        "Packet received count";
    }
  }
}

grouping protocol_state {
  description
    "RSVP protocol statistics grouping";
  container messages {
    description
      "RSVP protocol statistics container";
    leaf ack-sent {
      type yang:counter64;
      description
        "Hello sent count";
    }

    leaf ack-received {
      type yang:counter64;
      description
        "Hello received count";
    }

    leaf bundle-sent {
      type yang:counter64;
      description
        "Bundle sent count";
    }

    leaf bundle-received {
      type yang:counter64;
      description
        "Bundle received count";
    }

    leaf hello-sent {
      type yang:counter64;
      description
        "Hello sent count";
    }

    leaf hello-received {
      type yang:counter64;
```

```
        description
          "Hello received count";
      }

      leaf integrity-challenge-sent {
        type yang:counter64;
        description
          "Integrity Challenge sent count";
      }

      leaf integrity-challenge-received {
        type yang:counter64;
        description
          "Integrity Challenge received count";
      }

      leaf integrity-response-sent {
        type yang:counter64;
        description
          "Integrity Response sent count";
      }

      leaf integrity-response-received {
        type yang:counter64;
        description
          "Integrity Response received count";
      }

      leaf notify-sent {
        type yang:counter64;
        description
          "Notify sent count";
      }

      leaf notify-received {
        type yang:counter64;
        description
          "Notify received count";
      }

      leaf path-sent {
        type yang:counter64;
        description
          "Path sent count";
      }

      leaf path-received {
        type yang:counter64;
```

```
        description
          "Path received count";
      }

      leaf path-err-sent {
        type yang:counter64;
        description
          "Path error sent count";
      }

      leaf path-err-received {
        type yang:counter64;
        description
          "Path error received count";
      }

      leaf path-tear-sent {
        type yang:counter64;
        description
          "Path tear sent count";
      }

      leaf path-tear-received {
        type yang:counter64;
        description
          "Path tear received count";
      }

      leaf resv-sent {
        type yang:counter64;
        description
          "Resv sent count";
      }

      leaf resv-received {
        type yang:counter64;
        description
          "Resv received count";
      }

      leaf resv-confirm-sent {
        type yang:counter64;
        description
          "Confirm sent count";
      }

      leaf resv-confirm-received {
        type yang:counter64;
```

```
        description
          "Confirm received count";
      }

      leaf resv-err-sent {
        type yang:counter64;
        description
          "Resv error sent count";
      }

      leaf resv-err-received {
        type yang:counter64;
        description
          "Resv error received count";
      }

      leaf resv-tear-sent {
        type yang:counter64;
        description
          "Resv tear sent count";
      }

      leaf resv-tear-received {
        type yang:counter64;
        description
          "Resv tear received count";
      }

      leaf summary-refresh-sent {
        type yang:counter64;
        description
          "Summary refresh sent count";
      }

      leaf summary-refresh-received {
        type yang:counter64;
        description
          "Summary refresh received count";
      }

      leaf unknown-messages-received {
        type yang:counter64;
        description
          "Unknown packet received count";
      }
    }
  }
}
```

```
grouping errors_state {
  description
    "Error statistics state grouping";
  container errors {
    description
      "Error statistics state container";
    leaf authenticate {
      type yang:counter64;
      description
        "The total number of packets received with an
        authentication failure.";
    }

    leaf checksum {
      type yang:counter64;
      description
        "The total number of packets received with an invalid
        checksum value.";
    }

    leaf packet-len {
      type yang:counter64;
      description
        "The total number of packets received with an invalid
        packet length.";
    }
  }
}

grouping statistics_state {
  description "RSVP statistic attributes.";
  container statistics {
    description
      "statistics state container";
    container state {
      config false;
      description
        "State information associated with RSVP
        hello parameters";
      uses protocol_state;
      uses packets_state;
      uses errors_state;
    }
  }
}

grouping neighbor-derived_state {
  description
```

```
    "Derived state at neighbor level.";

    leaf address {
      type inet:ip-address;
      description
        "Address of RSVP neighbor";
    }

    leaf epoch {
      type uint32;
      description
        "Neighbor epoch.";
    }

    leaf expiry-time {
      type uint32;
      units seconds;
      description
        "Neighbor expiry time after which the neighbor state
        is purged if no states associated with it";
    }

    container graceful-restart {
      description
        "Graceful restart information.";

      leaf enabled {
        type boolean;
        description
          "'true' if graceful restart is enabled for the
          neighbor.";
      }

      leaf local-restart-time {
        type uint32;
        units seconds;
        description
          "Local node restart time";
      }

      leaf local-recovery-time {
        type uint32;
        units seconds;
        description
          "Local node recover time";
      }

      leaf neighbor-restart-time {
```

```
    type uint32;
    units seconds;
    description
        "Neighbor restart time";
}

leaf neighbor-recovery-time {
    type uint32;
    units seconds;
    description
        "Neighbor recover time";
}

container helper-mode {
    description
        "Helper mode information ";

    leaf enabled {
        type boolean;
        description
            "'true' if helper mode is enabled.";
    }

    leaf max-helper-restart-time {
        type uint32;
        units seconds;
        description
            "The time the router or switch waits after it
            discovers that a neighboring router has gone down
            before it declares the neighbor down";
    }

    leaf max-helper-recovery-time {
        type uint32;
        units seconds;
        description
            "The amount of time the router retains the state of its
            RSVP neighbors while they undergo a graceful restart";
    }

    leaf neighbor-restart-time-remaining {
        type uint32;
        units seconds;
        description
            "Number of seconds remaining for neighbor to send
            Hello message after restart.";
    }
}
```



```
    leaf neighbor-recovery-time-remaining {
      type uint32;
      units seconds;
      description
        "Number of seconds remaining for neighbor to
        refresh.";
    }
  } // helper-mode
} // graceful-restart

leaf hello-status {
  type enumeration {
    enum "enabled" {
      description
        "Enabled";
    }
    enum "disabled" {
      description
        "Disabled";
    }
    enum "restarting" {
      description
        "Restarting";
    }
  }
  description
    "Hello status";
}

leaf interface {
  type if:interface-ref;
  description
    "Interface where RSVP neighbor was detected";
}

leaf neighbor-state {
  type enumeration {
    enum "up" {
      description
        "up";
    }
    enum "down" {
      description
        "down";
    }
    enum "hello-disable" {
      description
        "hello-disable";
    }
  }
}
```

```
    }
    enum "restarting" {
      description
        "restarting";
    }
  }
  description
    "Neighbor state";
}

leaf refresh-reduction-capable {
  type boolean;
  description
    "enables all RSVP refresh reduction message
    bundling, RSVP message ID, reliable message delivery
    and summary refresh";
  reference
    "RFC 2961 RSVP Refresh Overhead Reduction
    Extensions";
}

leaf restart-count {
  type yang:counter32;
  description
    "Number of times this neighbor restart";
}

leaf restart-time {
  type yang:date-and-time;
  description
    "Last restart time of the neighbor";
}
}

grouping global-attributes {
  description
    "Top level grouping for RSVP global properties";
  container sessions {
    description
      "RSVP sessions container";
    list session {
      key "local-index";
      config false;
      description
        "List of RSVP sessions";

      leaf local-index {
        type leafref {
```

```
        path "../state/local-index";
    }
    description
        "Reference to the local index for the RSVP
        session";
    }
    container state {
        config false;
        description
            "State information associated with RSVP
            session parameters";
        uses session-attributes_state;
    }
    }
    }
    uses statistics_state;
}

grouping intf-attributes {
    description
        "Top level grouping for RSVP interface properties";
    uses signaling-parameters;
    uses refresh-reduction;
    uses hellos;
    uses authentication;
    uses statistics_state;
}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol" {
    when "rt:type = 'rsvp:rsvp'" {
        description
            "This augment is only valid when routing protocol
            instance type is RSVP.";
    }
    description
        "RSVP protocol augmentation";
    container rsvp {
        presence "Enable RSVP feature";
        description "RSVP feature container";
        container globals {
            description "RSVP global properties.";
            uses global-attributes;
            uses graceful-restart;
        }

        container interfaces {
            description
```

```

    "RSVP interfaces container";
uses intf-attributes;

list interface {
    key "interface";
description
    "RSVP interfaces.";
leaf interface {
    type if:interface-ref;
description
    "RSVP interface.";
}
uses intf-attributes;
}

container neighbors {
description
    "RSVP neighbors container";
list neighbor {
    key "address";
description
    "List of RSVP neighbors";
leaf address {
    type leafref {
        path "../config/address";
    }
description
    "Address of RSVP neighbor";
}
uses neighbor-attributes;
}
}
}
}
<CODE ENDS>
```

3.2. RSVP Extended YANG Model

The RSVP extended YANG model covers optional or non-core RSVP feature(s). It also covers feature(s) that are not necessarily supported by all vendors, and hence, guarded with "if-feature" checks.

3.2.1. Tree Diagram

Figure 4 shows the YANG tree representation for configuration and state data that is augmenting the RSVP basic module:

```

module: ietf-rsvp
augment
  /rt:routing/rt:control-plane-protocols/rt:control-plane-protocol:
  +--rw rsvp!
    +--rw globals
      +--rw sessions
        +--ro session* [local-index]
          +--ro local-index    -> ../state/local-index
          +--ro state
            +--ro local-index?      uint64
            +--ro destination-port? inet:port-number
            +--ro source?           inet:ip-address
            +--ro destination?      inet:ip-address
            +--ro session-name?     string
            +--ro session-state?    enumeration
            +--ro session-type?     identityref
            +--ro psbs
              +--ro psb*
                +--ro source-port?  inet:port-number
                +--ro expires-in?    uint32
            +--ro rsbs
              +--ro rsb*
                +--ro source-port?  inet:port-number
                +--ro reservation-style? identityref
                +--ro expires-in?    uint32
      +--rw statistics
        +--ro state
          +--ro messages
            +--ro ack-sent?          yang:counter64
            +--ro ack-received?      yang:counter64
            +--ro bundle-sent?       yang:counter64
            +--ro bundle-received?   yang:counter64
            +--ro hello-sent?        yang:counter64
            +--ro hello-received?    yang:counter64
            +--ro integrity-challenge-sent? yang:counter64
            +--ro integrity-challenge-received? yang:counter64
            +--ro integrity-response-sent? yang:counter64
            +--ro integrity-response-received? yang:counter64
            +--ro notify-sent?       yang:counter64
            +--ro notify-received?   yang:counter64
            +--ro path-sent?         yang:counter64
            +--ro path-received?     yang:counter64
            +--ro path-err-sent?     yang:counter64

```

```

| | | | | +--ro path-err-received?          yang:counter64
| | | | | +--ro path-tear-sent?           yang:counter64
| | | | | +--ro path-tear-received?       yang:counter64
| | | | | +--ro resv-sent?                yang:counter64
| | | | | +--ro resv-received?            yang:counter64
| | | | | +--ro resv-confirm-sent?        yang:counter64
| | | | | +--ro resv-confirm-received?    yang:counter64
| | | | | +--ro resv-err-sent?            yang:counter64
| | | | | +--ro resv-err-received?        yang:counter64
| | | | | +--ro resv-tear-sent?           yang:counter64
| | | | | +--ro resv-tear-received?       yang:counter64
| | | | | +--ro summary-refresh-sent?     yang:counter64
| | | | | +--ro summary-refresh-received? yang:counter64
| | | | | +--ro unknown-messages-received? yang:counter64
| | | | +--ro packets
| | | | | +--ro sent?                     yang:counter64
| | | | | +--ro received?                 yang:counter64
| | | | | +--ro rsvp-ext:discontinuity-time? yang:date-and-time
| | | | | +--ro rsvp-ext:out-dropped?     yang:counter64
| | | | | +--ro rsvp-ext:in-dropped?      yang:counter64
| | | | | +--ro rsvp-ext:out-error?       yang:counter64
| | | | | +--ro rsvp-ext:in-error?        yang:counter64
| | | | +--ro errors
| | | | | +--ro authenticate?            yang:counter64
| | | | | +--ro checksum?                 yang:counter64
| | | | | +--ro packet-len?               yang:counter64
| | | +--rw graceful-restart
| | | | +--rw config
| | | | | +--rw enabled?                  boolean
| | | | | +--rw rsvp-ext:restart-time?    uint32
| | | | | +--rw rsvp-ext:recovery-time?   uint32
| | | | +--ro state
| | | | | +--ro enabled?                  boolean
| | | | | +--ro rsvp-ext:restart-time?    uint32
| | | | | +--ro rsvp-ext:recovery-time?   uint32
| | +--rw interfaces
| | | +--rw config
| | | | +--rw rsvp-ext:refresh-interval?   uint32
| | | | +--rw rsvp-ext:refresh-misses?     uint32
| | | | +--rw rsvp-ext:checksum?           boolean
| | | | +--rw rsvp-ext:patherr-state-removal? empty
| | | | +--ro state
| | | | | +--ro rsvp-ext:refresh-interval? uint32
| | | | | +--ro rsvp-ext:refresh-misses?   uint32
| | | | | +--ro rsvp-ext:checksum?         boolean
| | | | | +--ro rsvp-ext:patherr-state-removal? empty
| | +--rw refresh-reduction
| | | +--rw config

```

```

| | | +--rw enabled? boolean
| | | +--rw rsvp-ext:bundle-message-max-size? uint32
| | | +--rw rsvp-ext:reliable-ack-hold-time? uint32
| | | +--rw rsvp-ext:reliable-ack-max-size? uint32
| | | +--rw rsvp-ext:reliable-retransmit-time? uint32
| | | +--rw rsvp-ext:reliable-srefresh? empty
| | | +--rw rsvp-ext:summary-max-size? uint32
| | +--ro state
| | | +--ro enabled? boolean
| | | +--ro rsvp-ext:bundle-message-max-size? uint32
| | | +--ro rsvp-ext:reliable-ack-hold-time? uint32
| | | +--ro rsvp-ext:reliable-ack-max-size? uint32
| | | +--ro rsvp-ext:reliable-retransmit-time? uint32
| | | +--ro rsvp-ext:reliable-srefresh? empty
| | | +--ro rsvp-ext:summary-max-size? uint32
| +--rw hellos
| | +--rw config
| | | +--rw enabled? boolean
| | | +--rw rsvp-ext:interface-based? empty
| | | +--rw rsvp-ext:hello-interval? uint32
| | | +--rw rsvp-ext:hello-misses? uint32
| | +--ro state
| | | +--ro enabled? boolean
| | | +--ro rsvp-ext:interface-based? empty
| | | +--ro rsvp-ext:hello-interval? uint32
| | | +--ro rsvp-ext:hello-misses? uint32
| +--rw authentication
| | +--rw config
| | | +--rw enabled? boolean
| | | +--rw authentication-key? string
| | | +--rw crypto-algorithm identityref
| | | +--rw rsvp-ext:lifetime? uint32
| | | +--rw rsvp-ext>window-size? uint32
| | | +--rw rsvp-ext:challenge? empty
| | | +--rw rsvp-ext:retransmits? uint32
| | | +--rw rsvp-ext:key-chain? key-chain:key-chain-ref
| | +--ro state
| | | +--ro enabled? boolean
| | | +--ro authentication-key? string
| | | +--ro crypto-algorithm identityref
| | | +--ro rsvp-ext:lifetime? uint32
| | | +--ro rsvp-ext>window-size? uint32
| | | +--ro rsvp-ext:challenge? empty
| | | +--ro rsvp-ext:retransmits? uint32
| | | +--ro rsvp-ext:key-chain? key-chain:key-chain-ref
| +--rw statistics
| | +--ro state
| | | +--ro messages

```

```

|--ro ack-sent?                yang:counter64
|--ro ack-received?            yang:counter64
|--ro bundle-sent?             yang:counter64
|--ro bundle-received?         yang:counter64
|--ro hello-sent?              yang:counter64
|--ro hello-received?          yang:counter64
|--ro integrity-challenge-sent? yang:counter64
|--ro integrity-challenge-received? yang:counter64
|--ro integrity-response-sent?  yang:counter64
|--ro integrity-response-received? yang:counter64
|--ro notify-sent?             yang:counter64
|--ro notify-received?         yang:counter64
|--ro path-sent?               yang:counter64
|--ro path-received?           yang:counter64
|--ro path-err-sent?           yang:counter64
|--ro path-err-received?       yang:counter64
|--ro path-tear-sent?          yang:counter64
|--ro path-tear-received?      yang:counter64
|--ro resv-sent?               yang:counter64
|--ro resv-received?           yang:counter64
|--ro resv-confirm-sent?       yang:counter64
|--ro resv-confirm-received?   yang:counter64
|--ro resv-err-sent?           yang:counter64
|--ro resv-err-received?       yang:counter64
|--ro resv-tear-sent?          yang:counter64
|--ro resv-tear-received?      yang:counter64
|--ro summary-refresh-sent?    yang:counter64
|--ro summary-refresh-received? yang:counter64
|--ro unknown-messages-received? yang:counter64
+--ro packets
|   |--ro sent?                yang:counter64
|   |--ro received?            yang:counter64
+--ro errors
|   |--ro authenticate?        yang:counter64
|   |--ro checksum?            yang:counter64
|   |--ro packet-len?          yang:counter64
+--rw interface* [interface]
+--rw interface                if:interface-ref
+--rw config
|   +--rw rsvp-ext:refresh-interval?    uint32
|   +--rw rsvp-ext:refresh-misses?      uint32
|   +--rw rsvp-ext:checksum?             boolean
|   +--rw rsvp-ext:patherr-state-removal? empty
+--ro state
|   +--ro rsvp-ext:refresh-interval?    uint32
|   +--ro rsvp-ext:refresh-misses?      uint32
|   +--ro rsvp-ext:checksum?             boolean
|   +--ro rsvp-ext:patherr-state-removal? empty

```



```

+--rw refresh-reduction
|
|   +--rw config
|   |   +--rw enabled?                               boolean
|   |   +--rw rsvp-ext:bundle-message-max-size?      uint32
|   |   +--rw rsvp-ext:reliable-ack-hold-time?        uint32
|   |   +--rw rsvp-ext:reliable-ack-max-size?        uint32
|   |   +--rw rsvp-ext:reliable-retransmit-time?     uint32
|   |   +--rw rsvp-ext:reliable-srefresh?            empty
|   |   +--rw rsvp-ext:summary-max-size?             uint32
|   +--ro state
|   |   +--ro enabled?                               boolean
|   |   +--ro rsvp-ext:bundle-message-max-size?      uint32
|   |   +--ro rsvp-ext:reliable-ack-hold-time?        uint32
|   |   +--ro rsvp-ext:reliable-ack-max-size?        uint32
|   |   +--ro rsvp-ext:reliable-retransmit-time?     uint32
|   |   +--ro rsvp-ext:reliable-srefresh?            empty
|   |   +--ro rsvp-ext:summary-max-size?             uint32
|
+--rw hellos
|
|   +--rw config
|   |   +--rw enabled?                               boolean
|   |   +--rw rsvp-ext:interface-based?              empty
|   |   +--rw rsvp-ext:hello-interval?               uint32
|   |   +--rw rsvp-ext:hello-misses?                 uint32
|   +--ro state
|   |   +--ro enabled?                               boolean
|   |   +--ro rsvp-ext:interface-based?              empty
|   |   +--ro rsvp-ext:hello-interval?               uint32
|   |   +--ro rsvp-ext:hello-misses?                 uint32
|
+--rw authentication
|
|   +--rw config
|   |   +--rw enabled?                               boolean
|   |   +--rw authentication-key?                   string
|   |   +--rw crypto-algorithm                       identityref
|   |   +--rw rsvp-ext:lifetime?                     uint32
|   |   +--rw rsvp-ext>window-size?                 uint32
|   |   +--rw rsvp-ext:challenge?                   empty
|   |   +--rw rsvp-ext:retransmits?                 uint32
|   |   +--rw rsvp-ext:key-chain?                    key-chain:key-chain-ref
|   +--ro state
|   |   +--ro enabled?                               boolean
|   |   +--ro authentication-key?                   string
|   |   +--ro crypto-algorithm                       identityref
|   |   +--ro rsvp-ext:lifetime?                     uint32
|   |   +--ro rsvp-ext>window-size?                 uint32
|   |   +--ro rsvp-ext:challenge?                   empty
|   |   +--ro rsvp-ext:retransmits?                 uint32
|   |   +--ro rsvp-ext:key-chain?                    key-chain:key-chain-ref
|
+--rw statistics

```

```

+--ro state
  +--ro messages
    +--ro ack-sent?          yang:counter64
    +--ro ack-received?      yang:counter64
    +--ro bundle-sent?       yang:counter64
    +--ro bundle-received?   yang:counter64
    +--ro hello-sent?        yang:counter64
    +--ro hello-received?    yang:counter64
    +--ro integrity-challenge-sent? yang:counter64
    +--ro integrity-challenge-received? yang:counter64
    +--ro integrity-response-sent? yang:counter64
    +--ro integrity-response-received? yang:counter64
    +--ro notify-sent?       yang:counter64
    +--ro notify-received?   yang:counter64
    +--ro path-sent?         yang:counter64
    +--ro path-received?     yang:counter64
    +--ro path-err-sent?     yang:counter64
    +--ro path-err-received? yang:counter64
    +--ro path-tear-sent?    yang:counter64
    +--ro path-tear-received? yang:counter64
    +--ro resv-sent?         yang:counter64
    +--ro resv-received?     yang:counter64
    +--ro resv-confirm-sent? yang:counter64
    +--ro resv-confirm-received? yang:counter64
    +--ro resv-err-sent?     yang:counter64
    +--ro resv-err-received? yang:counter64
    +--ro resv-tear-sent?    yang:counter64
    +--ro resv-tear-received? yang:counter64
    +--ro summary-refresh-sent? yang:counter64
    +--ro summary-refresh-received? yang:counter64
    +--ro unknown-messages-received? yang:counter64
  +--ro packets
    +--ro sent?              yang:counter64
    +--ro received?          yang:counter64
  +--ro errors
    +--ro authenticate?      yang:counter64
    +--ro checksum?          yang:counter64
    +--ro packet-len?        yang:counter64
+--rw neighbors
  +--rw neighbor* [address]
    +--rw address -> ../config/address
    +--rw config
      | +--rw address?      inet:ip-address
    +--ro state
      +--ro address?        inet:ip-address
      +--ro epoch?          uint32
      +--ro expiry-time?    uint32
      +--ro graceful-restart

```

	+++ro enabled?	boolean
	+++ro local-restart-time?	uint32
	+++ro local-recovery-time?	uint32
	+++ro neighbor-restart-time?	uint32
	+++ro neighbor-recovery-time?	uint32
	+++ro helper-mode	
	+++ro enabled?	boolean
	+++ro max-helper-restart-time?	uint32
	+++ro max-helper-recovery-time?	uint32
	+++ro neighbor-restart-time-remaining?	uint32
	+++ro neighbor-recovery-time-remaining?	uint32
	+++ro hello-status?	enumeration
	+++ro interface?	if:interface-ref
	+++ro neighbor-state?	enumeration
	+++ro refresh-reduction-capable?	boolean
	+++ro restart-count?	yang:counter32
	+++ro restart-time?	yang:date-and-time

Figure 4: RSVP extended model tree diagram

3.2.2. YANG Module

Figure 5 shows the RSVP extended YANG module:

```
<CODE BEGINS> file "ietf-rsvp-extended@2017-03-10.yang"
module ietf-rsvp-extended {

    namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";

    prefix "rsvp-ext";

    import ietf-rsvp {
        prefix "rsvp";
    }

    import ietf-routing {
        prefix "rt";
    }

    import ietf-yang-types {
        prefix "yang";
    }

    import ietf-key-chain {
        prefix "key-chain";
    }

    organization
```

"IETF Traffic Engineering Architecture and Signaling (TEAS)
Working Group";

contact

"WG Web: <<http://tools.ietf.org/wg/teas/>>

WG List: <<mailto:teas@ietf.org>>

WG Chair: Lou Berger
<<mailto:lberger@labn.net>>

WG Chair: Vishnu Pavan Beeram
<<mailto:vbeeram@juniper.net>>

Editor: Vishnu Pavan Beeram
<<mailto:vbeeram@juniper.net>>

Editor: Tarek Saad
<<mailto:tsaad@cisco.com>>

Editor: Rakesh Gandhi
<<mailto:rgandhi@cisco.com>>

Editor: Himanshu Shah
<<mailto:hshah@ciena.com>>

Editor: Xufeng Liu
<mailto:Xufeng_Liu@jabil.com>

Editor: Xia Chen
<<mailto:jescia.chenxia@huawei.com>>

Editor: Raqib Jones
<<mailto:raqib@Brocade.com>>

Editor: Bin Wen
<mailto:Bin_Wen@cable.comcast.com>;

description

"This module contains the Extended RSVP YANG data model.";

```
revision "2017-03-10" {  
  description "Latest revision of RSVP extended yang module.";  
  reference "RFC2205";  
}
```

```
/* RSVP features */  
feature authentication {
```

```
    description
      "Indicates support for RSVP authentication";
  }

  feature error-statistics {
    description
      "Indicates support for error statistics";
  }

  feature global-statistics {
    description
      "Indicates support for global statistics";
  }

  feature graceful-restart {
    description
      "Indicates support for RSVP graceful restart";
  }

  feature hellos {
    description
      "Indicates support for RSVP hellos (RFC3209).";
  }

  feature notify {
    description
      "Indicates support for RSVP notify message (RFC3473).";
  }

  feature refresh-reduction {
    description
      "Indicates support for RSVP refresh reduction
      (RFC2961).";
  }

  feature refresh-reduction-extended {
    description
      "Indicates support for RSVP refresh reduction
      (RFC2961).";
  }

  feature per-interface-statistics {
    description
      "Indicates support for per interface statistics";
  }

  grouping graceful-restart-extended_config {
    description
```

```
    "Configuration parameters relating to RSVP
    Graceful-Restart";
  leaf restart-time {
    type uint32;
    units seconds;
    description
      "Graceful restart time (seconds).";
    reference
      "RFC 5495: Description of the Resource
      Reservation Protocol - Traffic-Engineered
      (RSVP-TE) Graceful Restart Procedures";
  }
  leaf recovery-time {
    type uint32;
    units seconds;
    description
      "RSVP state recovery time";
  }
}

grouping authentication-extended_config {
  description
    "Configuration parameters relating to RSVP
    authentication";
  leaf lifetime {
    type uint32 {
      range "30..86400";
    }
    units seconds;
    description
      "Life time for each security association";
    reference
      "RFC 2747: RSVP Cryptographic
      Authentication";
  }
  leaf window-size {
    type uint32 {
      range "1..64";
    }
    description
      "Window-size to limit number of out-of-order
      messages.";
    reference
      "RFC 2747: RSVP Cryptographic
      Authentication";
  }
  leaf challenge {
    type empty;
  }
}
```

```
    description
      "Enable challenge messages.";
    reference
      "RFC 2747: RSVP Cryptographic
      Authentication";
  }
  leaf retransmits {
    type uint32 {
      range "1..10000";
    }
    description
      "Number of retransmits when messages are
      dropped.";
    reference
      "RFC 2747: RSVP Cryptographic
      Authentication";
  }
  leaf key-chain {
    type key-chain:key-chain-ref;
    description
      "Key chain name to authenticate RSVP
      signaling messages.";
    reference
      "RFC 2747: RSVP Cryptographic
      Authentication";
  }
}

grouping hellos-extended_config {
  description
    "Configuration parameters relating to RSVP
    hellos";
  leaf interface-based {
    type empty;
    description
      "Enable interface-based Hello adjacency if present.";
  }
  leaf hello-interval {
    type uint32;
    units milliseconds;
    description
      "Configure interval between successive Hello
      messages in milliseconds.";
    reference
      "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
      RFC 5495: Description of the Resource
      Reservation Protocol - Traffic-Engineered
      (RSVP-TE) Graceful Restart Procedures";
  }
}
```

```
    }
    leaf hello-misses {
      type uint32 {
        range "1..10";
      }
      description
        "Configure max number of consecutive missed
        Hello messages.";
      reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for
        LSP Tunnels RFC 5495: Description of the
        Resource Reservation Protocol - Traffic-
        Engineered (RSVP-TE) Graceful Restart
        Procedures";
    }
  }
}

grouping signaling-parameters-extended_config {
  description
    "Configuration parameters relating to RSVP
    signaling";
  leaf refresh-interval {
    type uint32;
    description
      "Set interval between successive refreshes";
  }
  leaf refresh-misses {
    type uint32;
    description
      "Set max number of consecutive missed
      messages for state expiry";
  }
  leaf checksum {
    type boolean;
    description
      "Enable RSVP message checksum computation";
  }
  leaf patherr-state-removal {
    type empty;
    description
      "State-Removal flag in Path Error message
      if present.";
  }
}

grouping refresh-reduction-extended_config {
  description
    "Configuration parameters relating to RSVP
```



```
    refresh reduction";

    leaf bundle-message-max-size {
      type uint32 {
        range "512..65000";
      }
      description
        "Configure maximum size (bytes) of a
        single RSVP Bundle message.";
    }
    leaf reliable-ack-hold-time {
      type uint32;
      units milliseconds;
      description
        "Configure hold time in milliseconds for
        sending RSVP ACK message(s).";
    }
    leaf reliable-ack-max-size {
      type uint32;
      description
        "Configure max size of a single RSVP ACK
        message.";
    }
    leaf reliable-retransmit-time {
      type uint32;
      units milliseconds;
      description
        "Configure min delay in milliseconds to
        wait for an ACK before a retransmit.";
    }
    leaf reliable-srefresh {
      type empty;
      description
        "Configure use of reliable messaging for
        summary refresh if present.";
    }
    leaf summary-max-size {
      type uint32 {
        range "20..65000";
      }
      description
        "Configure max size (bytes) of a single
        RSVP summary refresh message.";
    }
  }

  grouping packets-extended_state {
    description
```

```
    "Packet statistics.";
  leaf discontinuity-time {
    type yang:date-and-time;
    description
      "The time on the most recent occasion at which any one
       or more of the statistic counters suffered a
       discontinuity. If no such discontinuities have occurred
       since the last re-initialization of the local
       management subsystem, then this node contains the time
       the local management subsystem re-initialized itself.";
  }
  leaf out-dropped {
    type yang:counter64;
    description
      "Out packet drop count";
  }

  leaf in-dropped {
    type yang:counter64;
    description
      "In packet drop count";
  }

  leaf out-error {
    type yang:counter64;
    description
      "Out packet error count";
  }

  leaf in-error {
    type yang:counter64;
    description
      "In packet rx error count";
  }
}

grouping protocol-extended_state {
  description
    "RSVP protocol statistics.";
}

grouping errors-extended_state {
  description
    "Error statistics.";
}

grouping extended_state {
  description "RSVP statistic attributes.";
```

```
    uses packets-extended_state;
    uses protocol-extended_state;
    uses errors-extended_state;
}

/**
 * RSVP extensions augmentations
 */

/* RSVP globals graceful restart*/

augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:graceful-restart/rsvp:config" {
    description
        "RSVP globals configuration extensions";
    uses graceful-restart-extended_config;
}

augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:graceful-restart/rsvp:state" {
    description
        "RSVP globals configuration extensions";
    uses graceful-restart-extended_config;
}

/* RSVP statistics augmentation */
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:state/rsvp:packets" {
    description
        "RSVP packet stats extensions";
    uses packets-extended_state;
}

augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:state/rsvp:messages" {
    description
        "RSVP protocol message stats extensions";
    uses protocol-extended_state;
}

augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:state/rsvp:errors" {
    description
        "RSVP errors stats extensions";
    uses errors-extended_state;
}
```

```
}

/**
 * RSVP all interfaces extensions
 */

/* RSVP interface signaling extensions */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:config" {
  description
    "RSVP signaling all interfaces configuration extensions";
  uses signaling-parameters-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:state" {
  description
    "RSVP signaling all interfaces state extensions";
  uses signaling-parameters-extended_config;
}

/* RSVP refresh reduction extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:refresh-reduction/rsvp:config" {
  description
    "RSVP refresh-reduction all interface configuration
    extensions";
  uses refresh-reduction-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:refresh-reduction/rsvp:state" {
  description
    "RSVP refresh-reduction all interfaces state extensions";
  uses refresh-reduction-extended_config;
}

/* RSVP hellos extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:hellos/rsvp:config" {
  description
    "RSVP hello all interfaces configuration extensions";
  uses hellos-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
```

```
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:hellos/rsvp:state" {
description
    "RSVP hello all interfaces state extensions";
uses hellos-extended_config;
}

/* RSVP authentication extension */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:authentication/rsvp:config" {
description
    "RSVP authentication all interfaces configuration extensions";
uses authentication-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:authentication/rsvp:state" {
description
    "RSVP authentication all interfaces state extensions";
uses authentication-extended_config;
}

/**
 * RSVP interface extensions
 */

/* RSVP interface signaling extensions */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:config" {
description
    "RSVP signaling interface configuration extensions";
uses signaling-parameters-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:state" {
description
    "RSVP signaling interface state extensions";
uses signaling-parameters-extended_config;
}

/* RSVP refresh reduction extension */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:refresh-reduction/rsvp:config" {
description
```

```
    "RSVP refresh-reduction interface configuration extensions";
    uses refresh-reduction-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:refresh-reduction/rsvp:state" {
    description
        "RSVP refresh-reduction interface state extensions";
    uses refresh-reduction-extended_config;
}

/* RSVP hellos extension */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:hellos/rsvp:config" {
    description
        "RSVP hello interface configuration extensions";
    uses hellos-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:hellos/rsvp:state" {
    description
        "RSVP hello interface state extensions";
    uses hellos-extended_config;
}

/* RSVP authentication extension */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:authentication/rsvp:config" {
    description
        "RSVP authentication interface configuration extensions";
    uses authentication-extended_config;
}
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
    "rsvp:interface/rsvp:authentication/rsvp:state" {
    description
        "RSVP authentication interface state extensions";
    uses authentication-extended_config;
}
}
<CODE ENDS>
```

Figure 5: RSVP extended YANG module

4. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-rsvp namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp
prefix: ietf-rsvp reference: RFC3209

name: ietf-rsvp-extended namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
prefix: ietf-rsvp-extended reference: RFC3209

5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF

users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

6. Acknowledgement

The authors would like to thank Lou Berger, for reviewing and providing valuable feedback on this document.

7. Contributors

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <<http://www.rfc-editor.org/info/rfc2747>>.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<http://www.rfc-editor.org/info/rfc2961>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.

- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<http://www.rfc-editor.org/info/rfc3473>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <<http://www.rfc-editor.org/info/rfc5063>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<http://www.rfc-editor.org/info/rfc6991>>.
- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <<http://www.rfc-editor.org/info/rfc8022>>.

8.2. Informative References

[I-D.openconfig-mpls-consolidated-model]

George, J., Fang, L., eric.osborne@level3.com, e., and R. Shakir, "MPLS / TE Model for Service Provider Networks", draft-openconfig-mpls-consolidated-model-02 (work in progress), October 2015.

[I-D.openconfig-netmod-opstate]

Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling of Operational State Data in YANG", draft-openconfig-netmod-opstate-01 (work in progress), July 2015.

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Tarek Saad (editor)
Cisco Systems, Inc.

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Jabil

Email: Xufeng_Liu@jabil.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 July 2022

V.P. Beeram
T. Saad
Juniper Networks
R. Gandhi
Cisco Systems, Inc.
X. Liu
Volta Networks
I. Bryskin
Individual
9 January 2022

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-17

Abstract

This document defines a YANG data model for the configuration and management of the RSVP protocol. The YANG data model covers the building blocks that may be augmented by other RSVP extension data models such as RSVP Traffic-Engineering (RSVP-TE). It is divided into two modules that cover the basic and extended RSVP features.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 13 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
2.1. Prefixes in Data Node Names	3
2.2. Model Tree Diagram	4
3. Model Overview	4
3.1. Module(s) Relationship	5
3.2. Core Features	5
3.3. Optional Features	6
3.4. Data Model Structure	6
3.5. Model Notifications	8
4. RSVP Base YANG Model	9
4.1. Tree Diagram	9
4.2. YANG Module	13
5. RSVP Extended YANG Model	34
5.1. Tree Diagram	34
5.2. YANG Module	36
6. IANA Considerations	45
7. Security Considerations	46
8. Acknowledgement	47
9. Appendix A	47
10. Contributors	53
11. References	54
11.1. Normative References	54
11.2. Informative References	56
Authors' Addresses	57

1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model for the configuration and management of the RSVP protocol [RFC2205]. The data model is divided into two modules: a base and extended RSVP YANG modules. The RSVP

base YANG 'ietf-rsvp' module covers the data that is core to the function of the RSVP protocol and MUST be supported by vendors that support RSVP protocol [RFC2205]. The RSVP extended 'ietf-rsvp-extended' module covers the data that is optional, or provides ability to tune RSVP protocol base functionality. The support for RSVP extended module features by vendors is considered optional.

The RSVP YANG model provides the building blocks needed to allow augmentation by other models that extend the RSVP protocol- such as using RSVP extensions to signal Label Switched Paths (LSPs) as defined in [RFC3209].

The YANG module(s) defined in this document are compatible with the Network Management Datastore Architecture (NMDA) [RFC7950].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology for describing YANG data models is found in [RFC7950].

2.1. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
rt-types	ietf-routing-types	[RFC8294]
inet	ietf-inet-types	[RFC6991]
yang	ietf-yang-types	[RFC6991]
key-chain	ietf-key-chain	[RFC8177]

Table 1: Prefixes and corresponding YANG modules

2.2. Model Tree Diagram

A full tree diagram of the module(s) defined in this document is given in subsequent sections as per the syntax defined in [RFC8340].

3. Model Overview

The RSVP YANG module augments the "control-plane-protocol" entry from the 'ietf-routing' module defined in [RFC8349]. It also defines the identity "rsvp" of base type "rt:routing-protocol" to identify the RSVP routing protocol.

The 'ietf-rsvp' model defines a single instance of the RSVP protocol. The top 'rsvp' container encompasses data for one such RSVP protocol instance. Multiple instances can be defined as multiple control-plane protocols instances as described in [RFC8349].

The YANG data model defined has the common building blocks for the operation of the base RSVP protocol for the session type defined in [RFC2205]. The augmentation of this model by other models (e.g. to support RSVP Traffic Engineering (TE) extensions for signaling Label Switched Paths (LSPs)) are outside the scope of this document and are discussed in separate document(s).

3.1. Module(s) Relationship

This RSVP YANG data model defined in this document is divided into two modules: a base and extended modules. The RSVP data covered in 'ietf-rsvp' module are categorized as core to the function of the protocol and MUST be supported by vendors claiming the support for RSVP protocol [RFC2205].

The RSVP extended features that are covered in 'ietf-rsvp-extended' module are categorized as either optional or providing ability to better tune the basic functionality of the RSVP protocol. The support for RSVP extended features by all vendors is considered optional.

The relationship between the base and RSVP extended YANG modules and the IETF routing YANG model is shown in Figure 1.

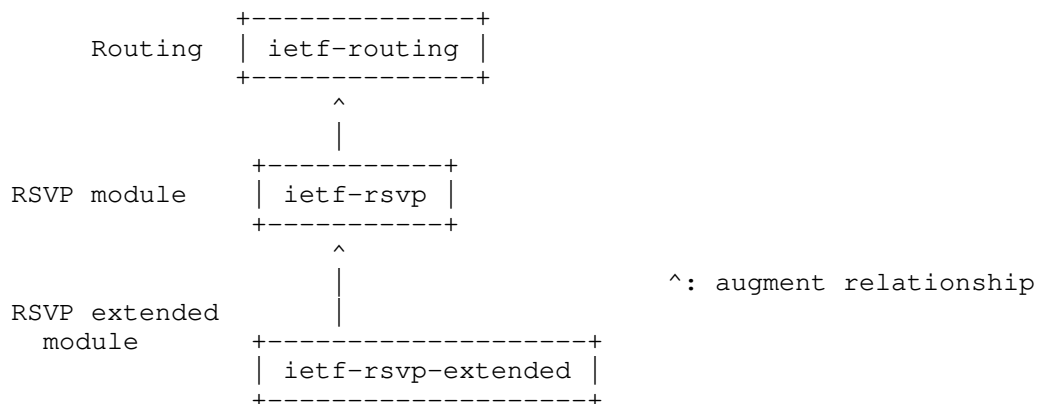


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

3.2. Core Features

The RSVP data covered in the 'ietf-rsvp' YANG module provides the common building blocks that are required to configure, operate and manage the RSVP protocol and MUST be supported by vendors that claim the support for base RSVP protocol defined in [RFC2205].

In addition, the following standard RSVP core features are modeled under the 'ietf-rsvp' module:

- * Basic operational statistics, including protocol messages, packets and errors.

- * Basic RSVP authentication feature as defined in [RFC2747]) using string based authentication key.
- * Basic RSVP Refresh Reduction feature as defined in ([RFC2961]).
- * Basic RSVP Hellos feature as defined in ([RFC3209])
- * Basic RSVP Graceful Restart feature as defined in [RFC3473], [RFC5063], and [RFC5495].

3.3. Optional Features

Optional features are beyond the basic configuration, and operation of the RSVP protocol. The decision whether to support these RSVP features on a particular device is left to the vendor that supports the RSVP core features.

The following optional features that are covered in the 'ietf-rsvp-extended' YANG module:

- * Advanced operational statistics, including protocol messages, packets and errors.
- * Advanced RSVP authentication features as defined in [RFC2747]) using various authentication key types including those defined in [RFC8177].
- * Advanced RSVP Refresh Reduction features defined in ([RFC2961]).
- * Advanced RSVP Hellos features as defined in [RFC3209], and [rfc4558].
- * Advanced RSVP Graceful Restart features as defined in [RFC3473], [RFC5063], and [RFC5495].

3.4. Data Model Structure

The RSVP YANG data model defines the 'rsvp' top-level container that contains the configuration and operational state for the RSVP protocol. The presence of this container enables the RSVP protocol functionality.

The 'rsvp' top-level container also includes data that has router level scope (i.e. applicable to all objects modeled under rsvp). It also contains configuration and state data about the following types of RSVP objects:

- * interfaces

* neighbors

* sessions

The derived state data is contained in "read-only" nodes directly under the intended object as shown in Figure 2.

```

module: ietf-rsvp
  +--rw rsvp!
    +--rw <<router-level scope data>>
      .
      .
    +--rw interfaces
      .
      +-- ro <<derived state associated with interfaces>>
      .
      .
    +--rw neighbors
      .
      +-- ro <<derived state associated with the LSP Tunnel>>
      .
      .
    +--rw sessions
      .
      +-- ro <<derived state associated with the LSP Tunnel>>
      .
  rpcs:
    +--x clear-session
    +--x clear-neighbor
    +--x clear-authentication

```

Figure 2: RSVP high-level tree model view

The following

'router-level':

The router-level scope configuration and state data are applicable to all modeled objects under the top-level 'rsvp' container, and MAY affect the RSVP protocol behavior.

'interfaces':

The 'interfaces' container includes a list of RSVP enabled interfaces. It also includes RSVP configuration and state data that is applicable to all interfaces. An entry in the interfaces list MAY carry its own configuration or state data. Any data or state under the "interfaces" container level is equally applicable to all interfaces unless it is explicitly overridden by configuration or state under a specific interface.

'neighbors' :

The 'neighbors' container includes a list of RSVP neighbors. An entry in the RSVP neighbor list MAY carry its own configuration and state relevant to the specific RSVP neighbor. The RSVP neighbors can be dynamically discovered using RSVP signaling, or can be explicitly configured.

'sessions':

The 'sessions' container includes a list RSVP sessions. An entry in the RSVP session list MAY carry its own configuration and state relevant to a specific RSVP session. RSVP sessions are usually derived state that are created as result of signaling. This model defines attributes related to IP RSVP sessions as defined in [RFC2205].

The defined YANG data model supports configuration inheritance for neighbors, and interfaces. Data nodes defined under the main container (e.g. the container that encompasses the list of interfaces, or neighbors) are assumed to apply equally to all elements of the list, unless overridden explicitly for a certain element (e.g. interface).

3.5. Model Notifications

Modeling notifications data is key in any defined YANG data model. [RFC8639] and [RFC8641] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

- * Subscribe notifications on a per client basis
- * Specify subtree filters [RFC6241] or XPath filters [RFC8639] so that only interested contents will be sent.
- * Specify either periodic or on-demand notifications.

4. RSVP Base YANG Model

The RSVP base module includes the core features and building blocks for modeling the RSVP protocol as described in Section 3.2.

4.1. Tree Diagram

Figure 3 shows the YANG tree representation for configuration, state data and RPCs that are covered in 'ietf-rsvp' YANG module:

module: ietf-rsvp

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol:
    +--rw rsvp!
      +--rw interfaces
        +--rw refresh-reduction
          | +--rw enabled?    boolean
        +--rw hellos
          | +--rw enabled?    boolean
        +--rw authentication
          | +--rw enabled?      boolean
          | +--rw authentication-key? string
          | +--rw crypto-algorithm identityref
        +--ro statistics
          +--ro messages
            +--ro ack-sent?          yang:counter64
            +--ro ack-received?      yang:counter64
            +--ro bundle-sent?       yang:counter64
            +--ro bundle-received?   yang:counter64
            +--ro hello-sent?        yang:counter64
            +--ro hello-received?    yang:counter64
            +--ro integrity-challenge-sent? yang:counter64
            +--ro integrity-challenge-received? yang:counter64
            +--ro integrity-response-sent? yang:counter64
            +--ro integrity-response-received? yang:counter64
            +--ro notify-sent?        yang:counter64
            +--ro notify-received?    yang:counter64
            +--ro path-sent?          yang:counter64
            +--ro path-received?      yang:counter64
            +--ro path-err-sent?      yang:counter64
            +--ro path-err-received?  yang:counter64
            +--ro path-tear-sent?     yang:counter64
            +--ro path-tear-received? yang:counter64
            +--ro resv-sent?          yang:counter64
            +--ro resv-received?      yang:counter64
            +--ro resv-confirm-sent?  yang:counter64
            +--ro resv-confirm-received? yang:counter64

```

```

+--ro resv-err-sent?                yang:counter64
+--ro resv-err-received?            yang:counter64
+--ro resv-tear-sent?               yang:counter64
+--ro resv-tear-received?           yang:counter64
+--ro srefresh-sent?                yang:counter64
+--ro srefresh-received?            yang:counter64
+--ro unknown-messages-received?    yang:counter64
+--ro packets
|   +--ro sent?                     yang:counter64
|   +--ro received?                 yang:counter64
+--ro errors
|   +--ro authenticate?             yang:counter64
|   +--ro checksum?                 yang:counter64
|   +--ro packet-length?            yang:counter64
+--rw interface* [name]
|   +--rw name                       if:interface-ref
|   +--rw refresh-reduction
|   |   +--rw enabled?              boolean
|   +--rw hellos
|   |   +--rw enabled?              boolean
|   +--rw authentication
|   |   +--rw enabled?              boolean
|   |   +--rw authentication-key?   string
|   |   +--rw crypto-algorithm      identityref
+--ro statistics
|   +--ro messages
|   |   +--ro ack-sent?
|   |   |   yang:counter64
|   |   +--ro ack-received?
|   |   |   yang:counter64
|   |   +--ro bundle-sent?
|   |   |   yang:counter64
|   |   +--ro bundle-received?
|   |   |   yang:counter64
|   |   +--ro hello-sent?
|   |   |   yang:counter64
|   |   +--ro hello-received?
|   |   |   yang:counter64
|   |   +--ro integrity-challenge-sent?
|   |   |   yang:counter64
|   |   +--ro integrity-challenge-received?
|   |   |   yang:counter64
|   |   +--ro integrity-response-sent?
|   |   |   yang:counter64
|   |   +--ro integrity-response-received?
|   |   |   yang:counter64
|   |   +--ro notify-sent?
|   |   |   yang:counter64

```

```

+--ro notify-received?
|   yang:counter64
+--ro path-sent?
|   yang:counter64
+--ro path-received?
|   yang:counter64
+--ro path-err-sent?
|   yang:counter64
+--ro path-err-received?
|   yang:counter64
+--ro path-tear-sent?
|   yang:counter64
+--ro path-tear-received?
|   yang:counter64
+--ro resv-sent?
|   yang:counter64
+--ro resv-received?
|   yang:counter64
+--ro resv-confirm-sent?
|   yang:counter64
+--ro resv-confirm-received?
|   yang:counter64
+--ro resv-err-sent?
|   yang:counter64
+--ro resv-err-received?
|   yang:counter64
+--ro resv-tear-sent?
|   yang:counter64
+--ro resv-tear-received?
|   yang:counter64
+--ro srefresh-sent?
|   yang:counter64
+--ro srefresh-received?
|   yang:counter64
+--ro unknown-messages-received?
|   yang:counter64
+--ro packets
|   +--ro sent?          yang:counter64
|   +--ro received?     yang:counter64
+--ro errors
|   +--ro authenticate?  yang:counter64
|   +--ro checksum?     yang:counter64
|   +--ro packet-length? yang:counter64
+--rw sessions
|   +--ro session-ip*
|       [destination protocol-id destination-port]
|   +--ro destination-port  uint16
|   +--ro protocol-id       uint8

```

```

    +--ro source?                inet:ip-address
    +--ro destination            inet:ip-address
    +--ro session-name?          string
    +--ro session-status?        enumeration
    +--ro session-type            identityref
    +--ro psbs
      +--ro psb* []
        +--ro source-port?      inet:port-number
        +--ro expires-in?       uint32
    +--ro rsbs
      +--ro rsb* []
        +--ro source-port?      inet:port-number
        +--ro reservation-style identityref
        +--ro expires-in?       uint32
+--rw neighbors
  +--rw neighbor* [address]
    +--rw address                inet:ip-address
    +--rw epoch?                 uint32
    +--rw expiry-time?           uint32
    +--rw graceful-restart
      +--ro neighbor-restart-time? uint32
      +--ro neighbor-recovery-time? uint32
      +--rw helper-mode
        +--ro neighbor-restart-time-remaining? uint32
        +--ro neighbor-recovery-time-remaining? uint32
    +--ro hello-status?          enumeration
    +--rw interface?             if:interface-ref
    +--ro neighbor-status?       enumeration
    +--rw refresh-reduction-capable? boolean
    +--ro restart-count?         yang:counter32
    +--ro restart-time?          yang:date-and-time
+--rw graceful-restart
  +--rw enabled?                 boolean
  +--rw local-restart-time?      uint32
  +--rw local-recovery-time?     uint32
  +--rw helper-mode
    +--rw enabled?               boolean
    +--rw max-helper-restart-time? uint32
    +--rw max-helper-recovery-time? uint32

rpcs:
  +---x clear-session
    +---w input
      +---w routing-protocol-instance-name leafref
      +---w (filter-type)
        +---:(match-all)
          +---w all                    empty
        +---:(match-one)

```

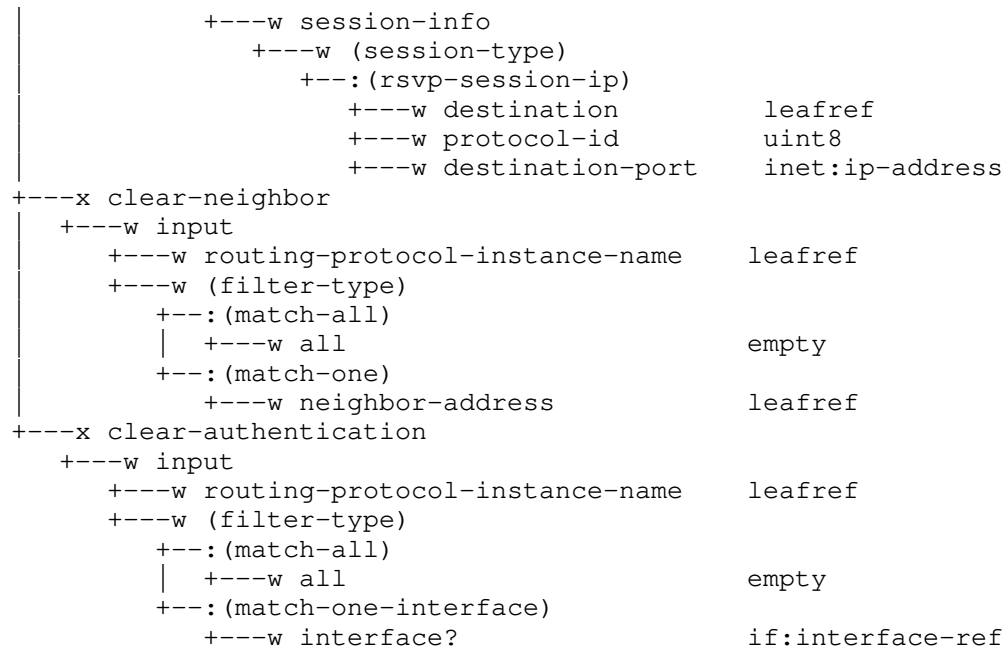


Figure 3: RSVP model tree diagram

4.2. YANG Module

The ietf-rsvp module imports from the following modules:

- * ietf-interfaces defined in [RFC8343]
- * ietf-yang-types and ietf-inet-types defined in [RFC6991]
- * ietf-routing defined in [RFC8349]
- * ietf-key-chain defined in [RFC8177]
- * ietf-netconf-acm defined in [RFC8341]

This module also references the following documents: [RFC2205], [RFC5495], [RFC3473], [RFC5063], [RFC2747], [RFC3209], and [RFC2961].

```

<CODE BEGINS> file "ietf-rsvp@2021-12-02.yang"
module ietf-rsvp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp";

  /* Replace with IANA when assigned */

```

```
prefix rsvp;

import ietf-interfaces {
  prefix if;
  reference
    "RFC8343: A YANG Data Model for Interface Management";
}
import ietf-inet-types {
  prefix inet;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC6991: Common YANG Data Types";
}
import ietf-routing {
  prefix rt;
  reference
    "RFC8349: A YANG Data Model for Routing Management
    (NMDA Version)";
}
import ietf-key-chain {
  prefix key-chain;
  reference
    "RFC8177: YANG Data Model for Key Chains";
}
import ietf-netconf-acm {
  prefix nacm;
  reference
    "RFC8341: Network Configuration Access Control Model";
}
organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:    <http://tools.ietf.org/wg/teas/>
  WG List:    <mailto:teas@ietf.org>

  Editor:     Vishnu Pavan Beeram
               <mailto:vbeeram@juniper.net>

  Editor:     Tarek Saad
               <mailto:tsaad@juniper.net>

  Editor:     Rakesh Gandhi
               <mailto:rgandhi@cisco.com>
```



```
Editor:  Xufeng Liu
        <mailto:xufeng.liu.ietf@gmail.com>

Editor:  Igor Bryskin
        <mailto:i_bryskin@yahoo.com>";
description
  "This module contains the RSVP YANG data model.
  The model fully conforms to the Network Management Datastore
  Architecture (NMDA).

  Copyright (c) 2019 IETF Trust and the persons
  identified as authors of the code.  All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2021-12-02 {
  description
    "Initial version.";
  reference
    "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
    (RSVP)";
}

identity rsvp {
  base rt:routing-protocol;
  description
    "RSVP protocol";
}

identity rsvp-session-type {
  description
    "Base RSVP session type";
}

identity rsvp-session-ip {
  base rsvp-session-type;
```

```
    description
      "RSVP IP session type";
  }

  identity reservation-style {
    description
      "Base identity for reservation style.";
  }

  identity reservation-wildcard-filter {
    base reservation-style;
    description
      "Wildcard-Filter (WF) Style.";
    reference
      "RFC2205";
  }

  identity reservation-fixed-filter {
    base reservation-style;
    description
      "Fixed-Filter (FF) Style.";
    reference
      "RFC2205";
  }

  identity reservation-shared-explicit {
    base reservation-style;
    description
      "Shared Explicit (SE) Style.";
    reference
      "RFC2205";
  }

  grouping graceful-restart {
    description
      "RSVP graceful restart local parameters grouping.";
    container graceful-restart {
      description
        "Graceful restart local information.";
      leaf enabled {
        type boolean;
        default "false";
        description
          "'true' if RSVP Graceful Restart is enabled.
          'false' if RSVP Graceful Restart is disabled.";
        reference "RFC5495";
      }
      leaf local-restart-time {
```

```
type uint32;
units "seconds";
default "120";
description
  "Time it takes the local node to restart its RSVP-TE
  component (to the point where it can exchange RSVP
  Hello with its neighbors). A value of 0xffffffff
  indicates that the restart of the neighbor's control plane
  may occur over an indeterminate interval and that the
  operation of its data plane is unaffected by control plane
  failures.";
reference "RFC3473";
}
leaf local-recovery-time {
  type uint32;
  units "seconds";
  default "120";
  description
    "The period of time, in seconds, that the local
    node requires to re-synchronize RSVP and MPLS
    forwarding state with its neighbor. A value of zero (0)
    indicates that MPLS forwarding state was not preserved
    across a particular reboot.";
  reference "RFC3473";
}
container helper-mode {
  description
    "Helper mode information. In this mode, the node
    resynchronize its stored states with a neighbor whose
    control plane has restarted. The helper mode term is
    borrowed from RFC3623 and adopted by several vendors
    vendors in their implementation of RSVP graceful restart.";
  leaf enabled {
    type boolean;
    default "true";
    description
      "'true' if helper mode is enabled.";
  }
  leaf max-helper-restart-time {
    type uint32;
    units "seconds";
    default "20";
    description
      "The maximum time the router or switch waits after it
      discovers that the neighboring router has gone down
      before it declares the neighbor down.";
    reference "RFC5063";
  }
}
```

```
    leaf max-helper-recovery-time {
      type uint32;
      units "seconds";
      default "180";
      description
        "The maximum amount of time the router retains the state
         of its RSVP neighbors while they undergo a graceful
         restart.";
      reference "RFC5063";
    }
  }
}

grouping neighbor-graceful-restart {
  description
    "RSVP graceful restart neighbor parameters grouping.";
  container graceful-restart {
    description
      "Graceful restart information.";
    leaf neighbor-restart-time {
      type uint32;
      units "seconds";
      default "120";
      config false;
      description
        "Time it takes the neighbor node to restart its RSVP-TE
         component (to the point where it can exchange RSVP
         Hello with its neighbors). A value of 0xffffffff
         indicates that the restart of the neighbor's control plane
         may occur over an indeterminate interval and that the
         operation of its data plane is unaffected by control plane
         failures.";
      reference "RFC3473";
    }
    leaf neighbor-recovery-time {
      type uint32;
      units "seconds";
      default "120";
      config false;
      description
        "The period of time, in milliseconds, that the neighbor
         node requires to re-synchronize RSVP and MPLS
         forwarding state with its neighbor. A value of zero (0)
         indicates that MPLS forwarding state was not preserved
         across a particular reboot.";
      reference "RFC3473";
    }
  }
  container helper-mode {
```

```
    description
      "Helper mode information.";
    leaf neighbor-restart-time-remaining {
      type uint32;
      units "seconds";
      config false;
      description
        "Number of seconds remaining for neighbor to send Hello
        message after restart.";
      reference "RFC5063";
    }
    leaf neighbor-recovery-time-remaining {
      type uint32;
      units "seconds";
      config false;
      description
        "Number of seconds remaining for neighbor to refresh.";
      reference "RFC5063";
    }
  }
  // helper-mode
}

grouping refresh-reduction {
  description
    "Top level grouping for RSVP refresh reduction parameters.";
  container refresh-reduction {
    description
      "Top level container for RSVP refresh reduction parameters.";
    leaf enabled {
      type boolean;
      default "true";
      description
        "'true' if RSVP Refresh Reduction is enabled.
        'false' if RSVP Refresh Reduction is disabled.";
    }
    reference
      "RFC2961 RSVP Refresh Overhead Reduction Extensions";
  }
}

grouping authentication {
  description
    "Top level grouping for RSVP authentication parameters.";
  container authentication {
    description
      "Top level container for RSVP authentication parameters.";
  }
}
```

```
    leaf enabled {
      type boolean;
      default "false";
      description
        "'true' if RSVP Authentication is enabled.
        'false' if RSVP Authentication is disabled.";
    }
    leaf authentication-key {
      type string;
      default "";
      description
        "An authentication key string.";
      reference
        "RFC2747: RSVP Cryptographic Authentication";
    }
    leaf crypto-algorithm {
      type identityref {
        base key-chain:crypto-algorithm;
      }
      mandatory true;
      description
        "Cryptographic algorithm associated with key.";
    }
  }
}

grouping hellos {
  description
    "Top level grouping for RSVP hellos parameters.";
  container hellos {
    description
      "Top level container for RSVP hello parameters.";
    leaf enabled {
      type boolean;
      default "true";
      description
        "'true' if RSVP Hello is enabled.
        'false' if RSVP Hello is disabled.";
    }
    reference
      "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
      RFC5495: Description of the Resource Reservation Protocol -
      Traffic-Engineered (RSVP-TE) Graceful Restart Procedures.";
  }
}

grouping session-attributes {
  description
```

```
    "Top level grouping for RSVP session properties.";
  leaf destination-port {
    type uint16;
    description
      "RSVP destination port.";
    reference
      "RFC2205";
  }
  leaf protocol-id {
    type uint8;
    description
      "The IP protocol ID.";
    reference
      "RFC2205, section 3.2";
  }
  leaf source {
    type inet:ip-address;
    description
      "RSVP source address.";
    reference
      "RFC2205";
  }
  leaf destination {
    type inet:ip-address;
    description
      "RSVP destination address.";
    reference
      "RFC2205";
  }
  leaf session-name {
    type string;
    default "";
    description
      "The signaled name of this RSVP session.";
  }
  leaf session-status {
    type enumeration {
      enum up {
        description
          "RSVP session is up.";
      }
      enum down {
        description
          "RSVP session is down.";
      }
    }
    default "down";
    description
```

```
        "Enumeration of RSVP session states.";
    }
    leaf session-type {
        type identityref {
            base rsvp-session-type;
        }
        mandatory "true";
        description
            "RSVP session type.";
    }
    container psbs {
        description
            "Path State Block (PSB) container.";
        list psb {
            description
                "List of Path State Blocks.";
            leaf source-port {
                type inet:port-number;
                description
                    "RSVP source port.";
                reference
                    "RFC2205";
            }
            leaf expires-in {
                type uint32;
                units "seconds";
                default "180";
                description
                    "Time to expiry (in seconds).";
            }
        }
    }
    container rsbs {
        description
            "Reservation State Block (RSB) container.";
        list rsb {
            description
                "List of Reservation State Blocks.";
            leaf source-port {
                type inet:port-number;
                description
                    "RSVP source port.";
                reference
                    "RFC2205";
            }
            leaf reservation-style {
                type identityref {
                    base reservation-style;
                }
            }
        }
    }
}
```



```
    }
    mandatory "true";
    description
      "RSVP reservation style.";
  }
  leaf expires-in {
    type uint32;
    units "seconds";
    default "180";
    description
      "Time to expiry (in seconds).";
  }
}

grouping neighbor-attributes {
  description
    "Top level grouping for RSVP neighbor properties.";
  leaf address {
    type inet:ip-address;
    description
      "Address of the RSVP neighbor.";
  }
  leaf epoch {
    type uint32;
    default "0";
    description
      "Neighbor epoch.";
    reference "RFC5063";
  }
  leaf expiry-time {
    type uint32;
    units "seconds";
    default "180";
    description
      "Neighbor expiry time after which the neighbor state is
      purged if no states associated with it.";
  }
  uses neighbor-graceful-restart {
    description
      "Allows configuration applicable to all
      neighbors";
  }
  leaf hello-status {
    type enumeration {
      enum enabled {
        description
```

```
        "RSVP Hellos enabled.";
    }
    enum disabled {
        description
            "RSVP Hellos disabled.";
    }
    enum restarting {
        description
            "RSVP restarting.";
    }
}
config false;
description
    "RSVP Hello status.";
}
leaf interface {
    type if:interface-ref;
    description
        "Interface where RSVP neighbor was detected.";
}
leaf neighbor-status {
    type enumeration {
        enum up {
            description
                "Neighbor state up.";
        }
        enum down {
            description
                "Neighbor state down.";
        }
        enum hello-disable {
            description
                "RSVP Hellos disabled.";
        }
        enum restarting {
            description
                "RSVP neighbor restarting.";
        }
    }
}
config false;
description
    "RSVP neighbor state.";
}
leaf refresh-reduction-capable {
    type boolean;
    default "true";
    description
        "Enables all RSVP refresh reduction message bundling, RSVP
```

```
        message ID, reliable message delivery and Srefresh
        messages.";
    reference
        "RFC2961 RSVP Refresh Overhead Reduction Extensions";
}
leaf restart-count {
    type yang:counter32;
    config false;
    description
        "Number of times this RSVP neighbor has restarted.";
}
leaf restart-time {
    type yang:date-and-time;
    config false;
    description
        "Last restart time of the RSVP neighbor.";
    reference "RFC3473";
}
}

grouping packet-statistics {
    description
        "Packet statistics grouping.";
    container packets {
        description
            "Packet statistics container.";
        leaf sent {
            type yang:counter64;
            description
                "RSVP packet sent count.";
        }
        leaf received {
            type yang:counter64;
            description
                "RSVP packet received count.";
        }
    }
}

grouping message-statistics {
    description
        "RSVP protocol statistics grouping.";
    container messages {
        description
            "RSVP protocol statistics container.";
        leaf ack-sent {
            type yang:counter64;
            description
```

```
        "RSVP Hello sent count.";
    }
    leaf ack-received {
        type yang:counter64;
        description
            "RSVP Hello received count.";
    }
    leaf bundle-sent {
        type yang:counter64;
        description
            "RSVP Bundle message sent count.";
    }
    leaf bundle-received {
        type yang:counter64;
        description
            "RSVP Bundle message received count.";
    }
    leaf hello-sent {
        type yang:counter64;
        description
            "RSVP Hello message sent count.";
    }
    leaf hello-received {
        type yang:counter64;
        description
            "RSVP Hello message received count.";
    }
    leaf integrity-challenge-sent {
        type yang:counter64;
        description
            "RSVP Integrity Challenge message sent count.";
    }
    leaf integrity-challenge-received {
        type yang:counter64;
        description
            "RSVP Integrity Challenge message received count.";
    }
    leaf integrity-response-sent {
        type yang:counter64;
        description
            "RSVP Integrity Response message sent count.";
    }
    leaf integrity-response-received {
        type yang:counter64;
        description
            "RSVP Integrity Response message received count.";
    }
    leaf notify-sent {
```

```
    type yang:counter64;
    description
      "RSVP Notify message sent count.";
  }
  leaf notify-received {
    type yang:counter64;
    description
      "RSVP Notify message received count.";
  }
  leaf path-sent {
    type yang:counter64;
    description
      "RSVP Path message sent count.";
  }
  leaf path-received {
    type yang:counter64;
    description
      "RSVP Path message received count.";
  }
  leaf path-err-sent {
    type yang:counter64;
    description
      "RSVP Path error message sent count.";
  }
  leaf path-err-received {
    type yang:counter64;
    description
      "RSVP Path error message received count.";
  }
  leaf path-tear-sent {
    type yang:counter64;
    description
      "RSVP Path tear message sent count.";
  }
  leaf path-tear-received {
    type yang:counter64;
    description
      "RSVP Path tear message received count.";
  }
  leaf resv-sent {
    type yang:counter64;
    description
      "RSVP Resv message sent count.";
  }
  leaf resv-received {
    type yang:counter64;
    description
      "RSVP Resv message received count.";
```

```
    }
    leaf resv-confirm-sent {
      type yang:counter64;
      description
        "RSVP Confirm message sent count.";
    }
    leaf resv-confirm-received {
      type yang:counter64;
      description
        "RSVP Confirm message received count.";
    }
    leaf resv-err-sent {
      type yang:counter64;
      description
        "RSVP Resv error message sent count.";
    }
    leaf resv-err-received {
      type yang:counter64;
      description
        "RSVP Resv error message received count.";
    }
    leaf resv-tear-sent {
      type yang:counter64;
      description
        "RSVP Resv tear message sent count.";
    }
    leaf resv-tear-received {
      type yang:counter64;
      description
        "RSVP Resv tear message received count.";
    }
    leaf srefresh-sent {
      type yang:counter64;
      description
        "RSVP Srefresh message sent count.";
    }
    leaf srefresh-received {
      type yang:counter64;
      description
        "RSVP Srefresh message received count.";
    }
    leaf unknown-messages-received {
      type yang:counter64;
      description
        "Unknown messages received count.";
    }
  }
}
```

```
grouping errors-statistics {
  description
    "Error statistics grouping.";
  container errors {
    description
      "Error statistics container.";
    leaf authenticate {
      type yang:counter64;
      description
        "The total number of RSVP packets received with an
        authentication failure.";
    }
    leaf checksum {
      type yang:counter64;
      description
        "The total number of RSVP packets received with an invalid
        checksum value.";
    }
    leaf packet-length {
      type yang:counter64;
      description
        "The total number of packets received with an invalid
        packet length.";
    }
  }
}

grouping statistics {
  description
    "RSVP statistic attributes.";
  container statistics {
    config false;
    description
      "RSVP statistics container.";
    uses message-statistics;
    uses packet-statistics;
    uses errors-statistics;
  }
}

grouping intf-attributes {
  description
    "Top level grouping for RSVP interface properties.";
  uses refresh-reduction;
  uses hellos;
  uses authentication;
  uses statistics;
}
```

```
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol" {
    when "rt:type = 'rsvp:rsvp'" {
      description
        "This augment is only valid when routing protocol instance
        type is RSVP.";
    }
    description
      "RSVP protocol augmentation.";
    container rsvp {
      presence "Enable RSVP feature";
      description
        "RSVP feature container";
      container interfaces {
        description
          "RSVP interfaces container.";
        uses intf-attributes;
        list interface {
          key "name";
          description
            "RSVP interfaces.";
          leaf name {
            type if:interface-ref;
            description
              "RSVP interface.";
          }
          uses intf-attributes;
        }
      }
    }
    container sessions {
      description
        "RSVP sessions container.";
      list session-ip {
        key "destination protocol-id destination-port";
        config false;
        description
          "List of RSVP sessions.";
        uses session-attributes;
      }
    }
    container neighbors {
      description
        "RSVP neighbors container";
      list neighbor {
        key "address";
        description
          "List of RSVP neighbors";
        uses neighbor-attributes;
      }
    }
  }
```



```
    }
  }
  uses graceful-restart;
}

grouping session-ref {
  description
    "Session reference information";
  leaf destination {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols"
        + "/rt:control-plane-protocol/rsvp:rsvp"
        + "/rsvp:sessions/rsvp:session-ip/destination";
    }
    mandatory true;
    description
      "The RSVP session destination.";
  }
  leaf protocol-id {
    type uint8;
    mandatory true;
    description
      "The RSVP session protocol ID.";
  }
  leaf destination-port {
    type inet:ip-address;
    mandatory true;
    description
      "The RSVP session destination port.";
  }
}

rpc clear-session {
  nacm:default-deny-all;
  description
    "Clears RSVP sessions RPC";
  input {
    leaf routing-protocol-instance-name {
      type leafref {
        path "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol/rt:name";
      }
      mandatory true;
      description
        "Name of the RSVP protocol instance whose session
        is being cleared."
    }
  }
}
```

```
        If the corresponding RSVP instance doesn't exist,
        then the operation will fail with an error-tag of
        'data-missing' and an error-app-tag of
        'routing-protocol-instance-not-found'.";
    }
    choice filter-type {
        mandatory true;
        description
            "Filter choice";
        case match-all {
            leaf all {
                type empty;
                mandatory true;
                description
                    "Match all RSVP sessions.";
            }
        }
        case match-one {
            container session-info {
                description
                    "Specifies the specific session to invoke the operation
                    on.";
                choice session-type {
                    mandatory true;
                    description
                        "The RSVP session type.";
                    case rsvp-session-ip {
                        uses session-ref;
                    }
                }
            }
        }
    }
}

rpc clear-neighbor {
    nacm:default-deny-all;
    description
        "RPC to clear the RSVP Hello session to a neighbor.";
    input {
        leaf routing-protocol-instance-name {
            type leafref {
                path "/rt:routing/rt:control-plane-protocols/"
                    + "rt:control-plane-protocol/rt:name";
            }
            mandatory true;
            description

```

```
        "Name of the RSVP protocol instance whose session
        is being cleared.

        If the corresponding RSVP instance doesn't exist,
        then the operation will fail with an error-tag of
        'data-missing' and an error-app-tag of
        'routing-protocol-instance-not-found'.";
    }
    choice filter-type {
        mandatory true;
        description
            "The Filter choice.";
        case match-all {
            leaf all {
                type empty;
                mandatory true;
                description
                    "Match all RSVP neighbor sessions.";
            }
        }
        case match-one {
            leaf neighbor-address {
                type leafref {
                    path "/rt:routing/rt:control-plane-protocols"
                        + "/rt:control-plane-protocol/rsvp:rsvp"
                        + "/rsvp:neighbors/rsvp:neighbor/address";
                }
                mandatory true;
                description
                    "Match the specific RSVP neighbor session.";
            }
        }
    }
}

rpc clear-authentication {
    nacm:default-deny-all;
    description
        "Clears the RSVP Security Association (SA) before the
        lifetime expires.";
    input {
        leaf routing-protocol-instance-name {
            type leafref {
                path "/rt:routing/rt:control-plane-protocols/"
                    + "rt:control-plane-protocol/rt:name";
            }
            mandatory true;
        }
    }
}
```



```

module: ietf-rsvp-extended

augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp
    /rsvp:graceful-restart:
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
    +--rw refresh-interval?          uint32
    +--rw refresh-misses?           uint32
    +--rw checksum-enable?          empty
    +--rw patherr-state-removal?     empty
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:statistics/rsvp:packets:
    +--ro discontinuity-time?        yang:date-and-time
    +--ro out-dropped?               yang:counter64
    +--ro in-dropped?                yang:counter64
    +--ro out-errors?                yang:counter64
    +--ro in-errors?                 yang:counter64
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:refresh-reduction:
    +--rw bundle-message-max-size?   uint32
    +--rw ack-hold-time?              uint32
    +--rw ack-max-size?              uint32
    +--rw ack-retransmit-time?        uint32
    +--rw srefresh-ack-desired?       empty
    +--rw srefresh-max-size?          uint32
    +--rw srefresh-relative-period?   uint8
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:hellos:
    +--rw interface-based?           empty
    +--rw hello-interval?            uint32
    +--rw hello-misses?              uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:authentication:
    +--rw lifetime?                  uint32
    +--rw window-size?               uint32
    +--rw challenge?                  empty
    +--rw retransmits?               uint32
    +--rw key-chain?                  key-chain:key-chain-ref
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface:
    +--rw refresh-interval?           uint32
    +--rw refresh-misses?             uint32

```

```

    +--rw checksum-enable?          empty
    +--rw patherr-state-removal?    empty
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:statistics/rsvp:packets:
    +--ro discontinuity-time?      yang:date-and-time
    +--ro out-dropped?             yang:counter64
    +--ro in-dropped?             yang:counter64
    +--ro out-errors?             yang:counter64
    +--ro in-errors?              yang:counter64
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:refresh-reduction:
    +--rw bundle-message-max-size? uint32
    +--rw ack-hold-time?           uint32
    +--rw ack-max-size?           uint32
    +--rw ack-retransmit-time?    uint32
    +--rw srefresh-ack-desired?   empty
    +--rw srefresh-max-size?      uint32
    +--rw srefresh-relative-period? uint8
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:hellos:
    +--rw interface-based?        empty
    +--rw hello-interval?         uint32
    +--rw hello-misses?          uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface/rsvp:authentication:
    +--rw lifetime?              uint32
    +--rw window-size?          uint32
    +--rw challenge?             empty
    +--rw retransmits?           uint32
    +--rw key-chain?             key-chain:key-chain-ref

```

Figure 4: RSVP extended module tree diagram

5.2. YANG Module

The 'ietf-rsvp-extended' module imports from the following modules:

- * ietf-rsvp defined in this document
- * ietf-routing defined in [RFC8349]
- * ietf-yang-types and ietf-inet-types defined in [RFC6991]
- * ietf-key-chain defined in [RFC8177]

Figure 5 shows the RSVP extended YANG module:

This module also references the following documents: [RFC3473], [RFC2747], [RFC3209], [RFC2205], [RFC2961], and [RFC5495].

```
<CODE BEGINS> file "ietf-rsvp-extended@2021-12-02.yang"
module ietf-rsvp-extended {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";
  prefix rsvp-extended;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
      (RSVP)";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management
      (NMDA Version)";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC6991: Common YANG Data Types";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC8177: YANG Data Model for Key Chains";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web:  <http://tools.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

    Editor:   Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>

    Editor:   Tarek Saad
              <mailto:tsaad@juniper.net>

    Editor:   Rakesh Gandhi
```

```
<mailto:rgandhi@cisco.com>

Editor:   Xufeng Liu
         <mailto:xufeng.liu.ietf@gmail.com>

Editor:   Igor Bryskin
         <mailto:i_bryskin@yahoo.com>";
description
  "This module contains the Extended RSVP YANG data model.
  The model fully conforms to the Network Management Datastore
  Architecture (NMDA).

  Copyright (c) 2019 IETF Trust and the persons
  identified as authors of the code. All rights reserved.

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject
  to the license terms contained in, the Simplified BSD License
  set forth in Section 4.c of the IETF Trust's Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info).
  This version of this YANG module is part of RFC XXXX; see
  the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2021-12-02 {
  description
    "Initial version.";
  reference
    "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
    (RSVP)";
}

grouping graceful-restart-extended {
  description
    "Configuration parameters relating to RSVP Graceful-Restart.";
}

grouping authentication-extended {
  description
    "Configuration parameters relating to RSVP authentication.";
  leaf lifetime {
    type uint32 {
      range "30..86400";
    }
  }
}
```



```
    }
    units "seconds";
    default "30";
    description
      "Life time for each security association.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
  leaf window-size {
    type uint32 {
      range "1..64";
    }
    default "2";
    description
      "Window-size to limit number of out-of-order messages.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
  leaf challenge {
    type empty;
    description
      "Enable challenge messages.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
  leaf retransmits {
    type uint32 {
      range "1..10000";
    }
    default "1";
    description
      "Number of retransmits when messages are dropped.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
  leaf key-chain {
    type key-chain:key-chain-ref;
    description
      "Key chain name to authenticate RSVP
      signaling messages.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
}

grouping hellos-extended {
  description
    "Configuration parameters relating to RSVP hellos";
```

```
leaf interface-based {
  type empty;
  description
    "Enable interface-based Hello adjacency if present.";
}
leaf hello-interval {
  type uint32;
  units "milliseconds";
  default "9000";
  description
    "Configure interval between successive Hello messages in
    milliseconds.";
  reference
    "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
    RFC5495: Description of the Resource Reservation Protocol -
    Traffic-Engineered (RSVP-TE) Graceful Restart Procedures.";
}
leaf hello-misses {
  type uint32 {
    range "1..10";
  }
  default "3";
  description
    "Configure max number of consecutive missed Hello messages.";
  reference
    "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
    RFC5495: Description of the Resource Reservation Protocol -
    Traffic- Engineered (RSVP-TE) Graceful Restart Procedures.";
}
}

grouping signaling-parameters-extended {
  description
    "Configuration parameters relating to RSVP signaling";
  leaf refresh-interval {
    type uint32;
    units "seconds";
    default "30";
    description
      "Set interval between successive refreshes";
    reference "RFC2205";
  }
  leaf refresh-misses {
    type uint32;
    default "9";
    description
      "Set max number of consecutive missed messages for state
      expiry";
  }
}
```

```
        reference "RFC2205";
    }
    leaf checksum-enable {
        type empty;
        description
            "Enable RSVP message checksum computation";
        reference "RFC2205";
    }
    leaf patherr-state-removal {
        type empty;
        description
            "State-Removal flag in Path Error message if present.";
        reference "RFC3473";
    }
}

grouping refresh-reduction-extended {
    description
        "Configuration parameters relating to RSVP refresh reduction.";
    leaf bundle-message-max-size {
        type uint32 {
            range "512..65000";
        }
        default "1500";
        description
            "Configure maximum size (bytes) of a single RSVP Bundle
            message.";
        reference "RFC2961";
    }
    leaf ack-hold-time {
        type uint32;
        units "milliseconds";
        default "9000";
        description
            "Configure hold time in milliseconds for sending RSVP ACK
            message(s).";
        reference "RFC2961";
    }
    leaf ack-max-size {
        type uint32;
        default "1500";
        description
            "Configure max size of a single RSVP ACK message.";
        reference "RFC2961";
    }
    leaf ack-retransmit-time {
        type uint32;
        units "milliseconds";
    }
}
```

```
    default "500";
    description
      "Configure min delay in milliseconds to wait for an
       acknowledgment before being retransmitted.";
    reference "RFC2961";
  }
  leaf srefresh-ack-desired {
    type empty;
    description
      "Enables the sending of MESSAGE_ID with ACK_Desired
       set with Srefresh messages.";
    reference "RFC2961";
  }
  leaf srefresh-max-size {
    type uint32 {
      range "20..65000";
    }
    default "1500";
    description
      "Configure max size (bytes) of a single RSVP Srefresh
       message.";
    reference "RFC2961";
  }
  leaf srefresh-relative-period {
    type uint8 {
      range "10..100";
    }
    description
      "Configures the period of Srefreshes relative to standard
       refresh message period in percentage.";
  }
}

grouping packets-extended-statistics {
  description
    "Packet statistics.";
  leaf discontinuity-time {
    type yang:date-and-time;
    description
      "The time on the most recent occasion at which any one or
       more of the statistic counters suffered a discontinuity.
       If no such discontinuities have occurred since the last
       re-initialization of the local management subsystem, then
       this node contains the time the local management subsystem
       re-initialized itself.";
  }
  leaf out-dropped {
    type yang:counter64;
  }
}
```

```
        description
            "Out RSVP packet drop count.";
    }
    leaf in-dropped {
        type yang:counter64;
        description
            "In RSVP packet drop count.";
    }
    leaf out-errors {
        type yang:counter64;
        description
            "Out RSVP packet errors count.";
    }
    leaf in-errors {
        type yang:counter64;
        description
            "In RSVP packet rx errors count.";
    }
}

/**
 * RSVP extensions augmentations
 */
/* RSVP graceful restart*/

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/"
    + "rsvp:graceful-restart" {
    description
        "RSVP graceful restart configuration extensions";
    uses graceful-restart-extended;
}

/**
 * RSVP all interfaces extensions
 */

/* RSVP interface signaling extensions */
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
    description
        "RSVP signaling all interfaces configuration extensions";
    uses signaling-parameters-extended;
}

/* Packet statistics extension */
augment "/rt:routing/rt:control-plane-protocols/"
```

```
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:statistics/rsvp:packets" {
  description
    "RSVP packets all interfaces configuration extensions";
  uses packets-extended-statistics;
}

/* RSVP refresh reduction extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:refresh-reduction" {
  description
    "RSVP refresh-reduction all interface configuration
    extensions";
  uses refresh-reduction-extended;
}

/* RSVP hellos extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:hellos" {
  description
    "RSVP hello all interfaces configuration extensions";
  uses hellos-extended;
}

/* RSVP authentication extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:authentication" {
  description
    "RSVP authentication all interfaces configuration extensions";
  uses authentication-extended;
}

/**
 * RSVP per interface extensions
 */
/* RSVP interface signaling extensions */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface" {
  description
    "RSVP signaling interface configuration extensions";
  uses signaling-parameters-extended;
}
```

```
/* Packet statistics extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:statistics/rsvp:packets" {
  description
    "RSVP packet stats extensions";
  uses packets-extended-statistics;
}

/* RSVP refresh reduction extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:refresh-reduction" {
  description
    "RSVP refresh-reduction interface configuration extensions";
  uses refresh-reduction-extended;
}

/* RSVP hellos extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:hellos" {
  description
    "RSVP hello interface configuration extensions";
  uses hellos-extended;
}

/* RSVP authentication extension */
augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
  + "rsvp:interface/rsvp:authentication" {
  description
    "RSVP authentication interface configuration extensions";
  uses authentication-extended;
}
}
<CODE ENDS>
```

Figure 5: RSVP extended YANG module

6. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

```
name:      ietf-rsvp
namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp
prefix:    rsvp
reference:  RFCXXXX

name:      ietf-rsvp-extended
namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
prefix:    rsvp-extended
reference:  RFCXXXX
```

7. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in the YANG module(s) defined in this document that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/
rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions
```


All of which are considered sensitive and if access to either of these is compromised, it can result in temporary network outages or be employed to mount DoS attacks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

```
/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/  
rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions
```

Additional information from these state data nodes can be inferred with respect to the network topology, and device location and subsequently be used to mount other attacks in the network.

For RSVP authentication, the configuration supported is via the specification of key-chains [RFC8177] or the direct specification of key and authentication algorithm, and hence security considerations of [RFC8177] are inherited. This includes the considerations with respect to the local storage and handling of authentication keys.

Some of the RPC operations defined in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The RSVP YANG module support the "clear-session" and "clear-neighbor" RPCs. If access to either of these is compromised, they can result in temporary network outages be employed to mount DoS attacks.

The security considerations spelled out in the YANG 1.1 specification [RFC7950] apply for this document as well.

8. Acknowledgement

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Lou Berger for reviewing and providing valuable feedback on this document.

9. Appendix A

A simple network setup is shown in {fig-example title}. R1 runs the RSVP routing protocol on both interfaces 'ge0/0/0/1', and 'ge0/0/0/2'.

State on R1:

Sessions:

=====

Destination	Protocol-ID	Dest-port
198.51.100.1	10	10

Neighbors:

=====

Neighbor Address	Interface
192.0.2.6	ge0/0/0/1

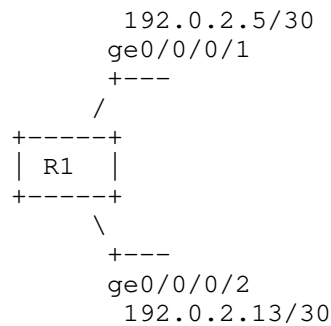


Figure 6: Example of network configuration.

The instance data tree could then be as follows:

```

{
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "rt:routing-protocol",
          "name": "rsvp:rsvp",
          "ietf-rsvp:rsvp": {
            "interfaces": {
              "refresh-reduction": {
                "enabled": true,
                "ietf-rsvp-extended:bundle-message-max-size": 2000,
                "ietf-rsvp-extended:reliable-ack-hold-time": 180,
                "ietf-rsvp-extended:reliable-ack-max-size": 2000,
                "ietf-rsvp-extended:reliable-retransmit-time": 180,
                "ietf-rsvp-extended:reliable-srefresh": [
                  null
                ],
                "ietf-rsvp-extended:summary-max-size": 2000
              }
            }
          }
        }
      ]
    }
  }
}

```

```
"hellos": {
  "enabled": true,
  "ietf-rsvp-extended:interface-based": [
    null
  ],
  "ietf-rsvp-extended:hello-interval": 27000,
  "ietf-rsvp-extended:hello-misses": 3
},
"statistics": {
  "messages": {
    "ack-sent": "777",
    "ack-received": "4840",
    "bundle-sent": "2195",
    "bundle-received": "293",
    "hello-sent": "2516",
    "hello-received": "3535",
    "integrity-challenge-sent": "2737",
    "integrity-challenge-received": "2330",
    "integrity-response-sent": "895",
    "integrity-response-received": "1029",
    "path-sent": "1197",
    "path-received": "3568",
    "path-err-sent": "4658",
    "path-err-received": "695",
    "path-tear-sent": "3706",
    "path-tear-received": "2604",
    "resv-sent": "3353",
    "resv-received": "3129",
    "resv-err-sent": "1787",
    "resv-err-received": "3205",
    "resv-tear-sent": "4465",
    "resv-tear-received": "3056",
    "summary-refresh-sent": "655",
    "summary-refresh-received": "3856"
  },
  "packets": {
    "sent": "2147",
    "received": "4374",
    "ietf-rsvp-extended:discontinuity-time":
      "2015-10-24T17:11:27+02:00",
    "ietf-rsvp-extended:out-dropped": "2696",
    "ietf-rsvp-extended:in-dropped": "941",
    "ietf-rsvp-extended:out-errors": "19",
    "ietf-rsvp-extended:in-errors": "2732"
  },
  "errors": {
    "authenticate": "2540",
    "checksum": "2566",
```

```
        "packet-length": "267"
      }
    },
    "interface": [
      {
        "interface": "ge0/0/0/1",
        "statistics": {
          "messages": {
            "ack-sent": "2747",
            "ack-received": "4934",
            "bundle-sent": "1618",
            "bundle-received": "3668",
            "hello-sent": "4288",
            "hello-received": "1194",
            "integrity-challenge-sent": "4850",
            "integrity-challenge-received": "3979",
            "integrity-response-sent": "479",
            "integrity-response-received": "1773",
            "path-sent": "2230",
            "path-received": "1793",
            "path-err-sent": "465",
            "path-err-received": "1859",
            "path-tear-sent": "923",
            "path-tear-received": "3924",
            "resv-sent": "3203",
            "resv-received": "2507",
            "resv-err-sent": "1259",
            "resv-err-received": "2445",
            "resv-tear-sent": "3045",
            "resv-tear-received": "4676",
            "summary-refresh-sent": "365",
            "summary-refresh-received": "2129"
          },
          "packets": {
            "sent": "847",
            "received": "3114",
            "ietf-rsvp-extended:discontinuity-time":
              "2015-10-24T17:11:27+02:00",
            "ietf-rsvp-extended:out-dropped": "1841",
            "ietf-rsvp-extended:in-dropped": "4832",
            "ietf-rsvp-extended:out-errors": "1334",
            "ietf-rsvp-extended:in-errors": "3900"
          },
          "errors": {
            "authenticate": "3494",
            "checksum": "4374",
            "packet-length": "2456"
          }
        }
      }
    ]
  }
}
```

```
    }  
  },  
  {  
    "interface": "ge0/0/0/2",  
    "statistics": {  
      "messages": {  
        "ack-sent": "1276",  
        "ack-received": "2427",  
        "bundle-sent": "4053",  
        "bundle-received": "3509",  
        "hello-sent": "3261",  
        "hello-received": "2863",  
        "integrity-challenge-sent": "4744",  
        "integrity-challenge-received": "3554",  
        "integrity-response-sent": "3155",  
        "integrity-response-received": "169",  
        "path-sent": "3853",  
        "path-received": "409",  
        "path-err-sent": "4227",  
        "path-err-received": "2830",  
        "path-tear-sent": "1742",  
        "path-tear-received": "3344",  
        "resv-sent": "3154",  
        "resv-received": "3492",  
        "resv-err-sent": "3112",  
        "resv-err-received": "3974",  
        "resv-tear-sent": "3657",  
        "resv-tear-received": "533",  
        "summary-refresh-sent": "4036",  
        "summary-refresh-received": "2123"  
      },  
      "packets": {  
        "sent": "473",  
        "received": "314",  
        "ietf-rsvp-extended:discontinuity-time":  
          "2015-10-24T17:11:27+02:00",  
        "ietf-rsvp-extended:out-dropped": "2042",  
        "ietf-rsvp-extended:in-dropped": "90",  
        "ietf-rsvp-extended:out-errors": "1210",  
        "ietf-rsvp-extended:in-errors": "1361"  
      },  
      "errors": {  
        "authenticate": "543",  
        "checksum": "2241",  
        "packet-length": "480"  
      }  
    }  
  }  
}
```

```
    ],
    "ietf-rsvp-extended:refresh-interval": 30,
    "ietf-rsvp-extended:refresh-misses": 5,
    "ietf-rsvp-extended:checksum_enabled": true,
    "ietf-rsvp-extended:patherr-state-removal": [
      null
    ]
  },
  "sessions": {
    "session-ip": [
      {
        "destination-port": 10,
        "protocol-id": 10,
        "destination": "198.51.100.1",
        "psbs": {
          "psb": [
            {
              "source-port": 10,
              "expires-in": 100
            }
          ]
        },
        "rsbs": {
          "rsb": [
            {
              "source-port": 10,
              "reservation-style":
                "rsvp:reservation-wildcard-filter",
              "expires-in": 100
            }
          ]
        }
      ]
    ]
  },
  "neighbors": {
    "neighbor": [
      {
        "address": "192.0.2.6",
        "epoch": 130,
        "expiry-time": 260,
        "graceful-restart": {
          "enabled": true,
          "local-restart-time": 271,
          "local-recovery-time": 138,
          "neighbor-restart-time": 341,
          "neighbor-recovery-time": 342
        }
      }
    ]
  }
}
```

```
        "hello-status": "enabled",
        "interface": "ge0/0/0/1",
        "restart-count": 2,
        "restart-time": "2015-10-24T17:11:27+02:00"
      }
    ]
  },
  "graceful-restart": {
    "enabled": true,
    "local-restart-time": 60,
    "local-recovery-time": 180,
    "neighbor-restart-time": 80,
    "neighbor-recovery-time": 200,
    "helper-mode": {
      "enabled": true
    }
  }
}
}
```

Figure 7: Example RSVP JSON encoded data instance tree.

10. Contributors

Himanshu Shah
Ciena

Email: hshah@ciena.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.

11.2. Informative References

- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <<https://www.rfc-editor.org/info/rfc2747>>.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<https://www.rfc-editor.org/info/rfc2961>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [rfc4558] Ali, Z., Rahman, R., Prairie, D., and D. Papadimitriou, "Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement", RFC 4558, DOI 10.17487/RFC4558, June 2006, <<https://www.rfc-editor.org/info/rfc4558>>.

- [RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <<https://www.rfc-editor.org/info/rfc5063>>.
- [RFC5495] Li, D., Gao, J., Satyanarayana, A., and S. Bardalai, "Description of the Resource Reservation Protocol - Traffic-Engineered (RSVP-TE) Graceful Restart Procedures", RFC 5495, DOI 10.17487/RFC5495, March 2009, <<https://www.rfc-editor.org/info/rfc5495>>.

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Igor Bryskin
Individual

Email: i_bryskin@yahoo.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 11 August 2022

T. Saad
Juniper Networks
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V.P. Beeram
Juniper Networks
I. Bryskin
Individual
O. Gonzalez de Dios
Telefonica
7 February 2022

A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths
and Interfaces
draft-ietf-teas-yang-te-29

Abstract

This document defines a YANG data model for the provisioning and management of Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model is divided into YANG modules that classify data into generic, device-specific, technology agnostic, and technology-specific elements.

This model covers data for configuration, operational state, remote procedural calls, and event notifications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 11 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	3
2.1. Prefixes in Data Node Names	4
2.2. Model Tree Diagrams	4
3. Design Considerations	5
3.1. State Data Organization	5
4. Model Overview	6
4.1. Module Relationship	6
5. TE YANG Model	7
5.1. Module Structure	7
5.1.1. TE Globals	9
5.1.2. TE Tunnels	12
5.1.3. TE LSPs	19
5.2. Tree Diagram	19
5.3. YANG Module	60
6. TE Device YANG Model	98
6.1. Module Structure	99
6.1.1. TE Interfaces	99
6.2. Tree Diagram	100
6.3. YANG Module	102
7. Notifications	116
8. TE Generic and Helper YANG Modules	117
9. IANA Considerations	117
10. Security Considerations	117
11. Acknowledgement	119
12. Contributors	119
13. Appendix A: Data Tree Examples	119
13.1. Basic Tunnel Setup	120
13.2. Global Named Path Constraints	121
13.3. Tunnel with Global Path Constraint	121
13.4. Tunnel with Per-tunnel Path Constraint	122
13.5. Tunnel State	123

14. References	124
14.1. Normative References	124
14.2. Informative References	127
Authors' Addresses	128

1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes YANG data model for Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model covers data applicable to generic or device-independent, device-specific, and Multiprotocol Label Switching (MPLS) technology specific.

The document describes a high-level relationship between the modules defined in this document, as well as other external protocol YANG modules. The TE generic YANG data model does not include any data specific to a signaling protocol. It is expected other data plane technology model(s) will augment the TE generic YANG data model.

Also, it is expected other YANG module(s) that model TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) [I-D.ietf-spring-segment-routing-policy] will augment the generic TE YANG module.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- * client
- * configuration data

- * state data

This document also makes use of the following terminology introduced in the YANG Data Modeling Language [RFC7950]:

- * augment

- * data model

- * data node

2.1. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te-types	ietf-te-types	[RFC8776]
te-packet-types	ietf-te-packet-types	[RFC8776]
te	ietf-te	this document
te-dev	ietf-te-device	this document

Table 1: Prefixes and corresponding YANG modules

2.2. Model Tree Diagrams

The tree diagrams extracted from the module(s) defined in this document are given in subsequent sections as per the syntax defined in [RFC8340].

3. Design Considerations

This document describes a generic TE YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technology models to reuse the TE generic data model and possibly augment it with technology specific data.

The elements of the generic TE YANG data model, including TE Tunnels, LSPs, and interfaces have leaf(s) that identify the technology layer where they reside. For example, the LSP encoding type can identify the technology associated with a TE Tunnel or LSP.

Also, the generic TE YANG data model does not cover signaling protocol data. The signaling protocol used to instantiate TE LSPs are outside the scope of this document and expected to be covered by augmentations defined in other document(s).

The following other design considerations are taken into account with respect data organization:

- * The generic TE YANG data model 'ietf-te' contains device independent data and can be used to model data off a device (e.g. on a TE controller). The device-specific TE data is defined in module 'ietf-te-device' as shown in Figure 1,
- * In general, minimal elements in the model are designated as "mandatory" to allow freedom to vendors to adapt the data model to their specific product implementation.
- * Suitable defaults are specified for all configurable elements.
- * The model declares a number of TE functions as features that can be optionally supported.

3.1. State Data Organization

The Network Management Datastore Architecture (NMDA) [RFC8342] addresses modeling state data for ephemeral objects. This document adopts the NMDA model for configuration and state data representation as per IETF guidelines for new IETF YANG models.

4. Model Overview

The data models defined in this document cover the core TE features that are commonly supported by different vendor implementations. The support of extended or vendor specific TE feature(s) is expected to be in either augmentations, or deviations to the model defined in this document.

4.1. Module Relationship

The generic TE YANG data model that is defined in "ietf-te.yang" covers the building blocks that are device independent and agnostic of any specific technology or control plane instances. The TE device model defined in "ietf-te-device.yang" augments the generic TE YANG data model and covers data that is specific to a device - for example, attributes of TE interfaces, or TE timers that are local to a TE node.

The TE data model for specific instances of data plane technology exist in a separate YANG module(s) that augment the generic TE YANG data model. For example, the MPLS-TE module "ietf-te-mpls.yang" is defined in another document and augments the TE generic model as shown in Figure 1.

The TE data model for specific instances of signaling protocol are outside the scope of this document and are defined in other documents. For example, the RSVP-TE YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp].

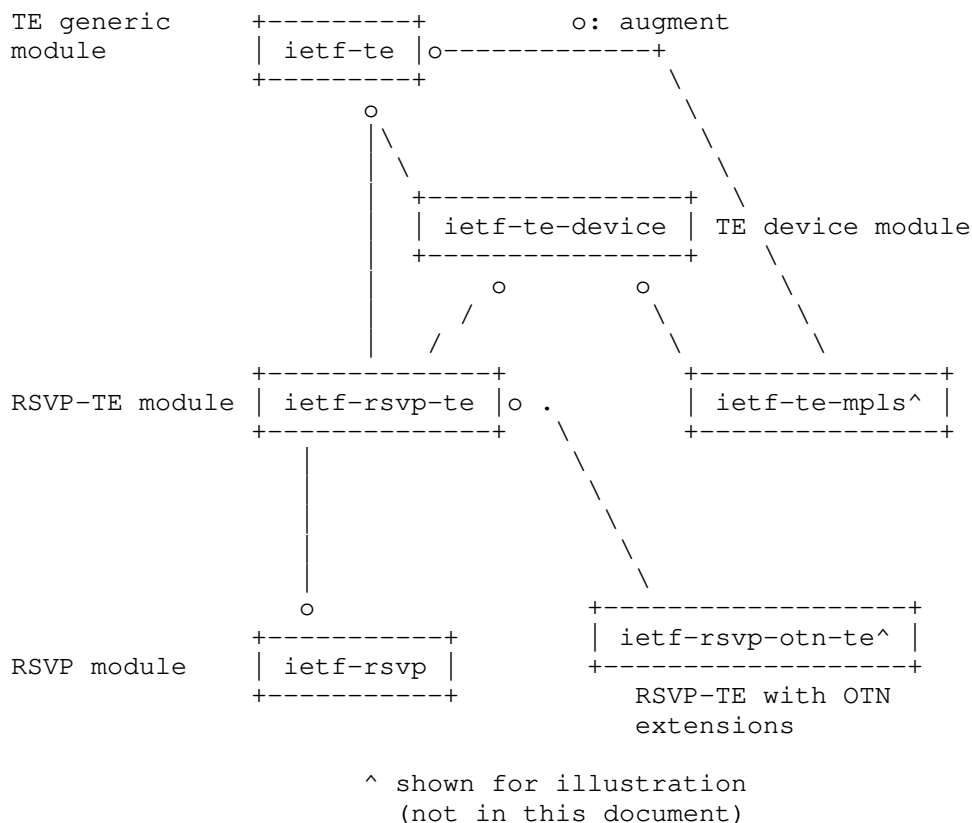


Figure 1: Relationship of TE module(s) with signaling protocol modules

5. TE YANG Model

The generic TE YANG module ('ietf-te') is meant to manage and operate a TE network. This includes creating, modifying and retrieving TE Tunnels, LSPs, and interfaces and their associated attributes (e.g. Administrative-Groups, SRLGs, etc.).

The detailed tree structure is provided in Figure 2.

5.1. Module Structure

The 'ietf-te' uses three main containers grouped under the main 'te' container (see Figure 2). The 'te' container is the top level container in the data model. The presence of the 'te' container enables TE function system wide. Below provides further descriptions of containers that exist under the 'te' top level container.

globals:

The 'globals' container maintains the set of global TE attributes that can be applicable to TE Tunnel(s) and interface(s).

tunnels:

The 'tunnels' container includes the list of TE Tunnels that are instantiated. Refer to Section 5.1.2 for further details on the properties of a TE Tunnel.

lsps:

The 'lsps' container includes the list of TE LSP(s) that are instantiated for TE Tunnels. Refer to Section 5.1.3 for further details on the properties of a TE LSP.

tunnels-path-compute:

A Remote Procedure Call (RPC) to request path computation for a specific TE Tunnel. The RPC allows requesting path computation using atomic and stateless operation. A tunnel may also be configured in 'compute-only' mode to provide stateful path updates - see Section 5.1.2 for further details.

tunnels-action:

An RPC to request a specific action (e.g. reoptimize, or tear-and-setup) to be taken on a specific tunnel or all tunnels.

```
module: ietf-te
  +--rw te!
    +--rw globals
      .
      .
    +--rw tunnels
      .
      .
    +-- lsps
```

```
rpcs:
  +---x tunnels-path-compute
  +---x tunnels-action
```

Figure 2: TE Tunnel model high-level YANG tree view

5.1.1. TE Globals

The 'globals' container covers properties that control TE features behavior system-wide, and its respective state (see Figure 3). The TE globals configuration include:

```

+--rw globals
|   +--rw named-admin-groups
|   |   +--rw named-admin-group* [name]
|   ..
|   +--rw named-srlgs
|   |   +--rw named-srlg* [name] {te-types:named-srlg-groups}?
|   ..
|   +--rw named-path-constraints
|   |   +--rw named-path-constraint* [name]
|   ..

```

Figure 3: TE globals YANG subtree high-level structure

named-admin-groups:

A YANG container for the list of named (extended) administrative groups that may be applied to TE links.

named-srlgs:

A YANG container for the list named Shared Risk Link Groups (SRLGs) that may be applied to TE links.

named-path-constraints:

A YANG container for a list of named path constraints. Each named path constraint is composed of a set of constraints that can be applied during path computation. A named path constraint can be applied to multiple TE Tunnels. Path constraints may also be specified directly under the TE Tunnel. The path constraint specified under the TE Tunnel take precedence over the path constraints derived from the referenced named path constraint. A named path constraint entry can be formed up of the following path constraints:

```

|   +--rw named-path-constraints
|       +--rw named-path-constraint* [name]
|           +--rw name                               string
|           +--rw te-bandwidth
| // ...
|       +--rw link-protection?                       identityref
|       +--rw setup-priority?                         uint8
|       +--rw hold-priority?                         uint8
|       +--rw signaling-type?                       identityref
|       +--rw path-metric-bounds
| // ...
|       +--rw path-affinities-values
| // ...
|       +--rw path-affinity-names
| // ...
|       +--rw path-srlgs-lists
| // ...
|       +--rw path-srlgs-names
| // ...
|       +--rw disjointness?
|           |
|           +--rw te-path-disjointness
| // ...
|       +--rw explicit-route-objects-always
| // ...
|       |   +--rw route-object-exclude-always* [index]
|       |   +--rw route-object-include-exclude* [index]

```

Figure 4: Named path constraints YANG subtree

- o te-bandwidth: A YANG container that holds the technology agnostic TE bandwidth constraint.
- o link-protection: A YANG leaf that holds the link protection type constraint required for the links to be included in the computed path.
- o setup/hold priority: A YANG leaf that holds the LSP setup and hold admission priority as defined in [RFC3209].
- o signaling-type: A YANG leaf that holds the LSP setup type, such as RSVP-TE or SR.
- o path-metric-bounds: A YANG container that holds the set of metric bounds applicable on the computed TE tunnel path.

- o `path-affinities-values`: A YANG container that holds the set of affinity values and mask to be used during path computation.
- o `path-affinity-names`: A YANG container that holds the set of named affinity constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `path-srlgs-lists`: A YANG container that holds the set of SRLG values and corresponding inclusion or exclusions instruction to be used during path computation.
- o `path-srlgs-names`: A YANG container that holds the set of named SRLG constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `disjointness`: The level of resource disjointness constraint that the secondary path of a TE tunnel has to adhere to.
- o `explicit-route-objects-always`: A YANG container that contains two route objects lists:
 - + `'route-object-exclude-always'`: a list of route entries to always exclude from the path computation.
 - + `'route-object-include-exclude'`: a list of route entries to include or exclude in the path computation.

The `'route-object-include-exclude'` is used to configure constraints on which route objects (e.g., nodes, links) are included or excluded in the path computation.

The interpretation of an empty `'route-object-include-exclude'` list depends on the TE Tunnel (end-to-end or Tunnel Segment) and on the specific path, according to the following rules:

1. An empty `'route-object-include-exclude'` list for the primary path of an end-to-end TE Tunnel indicates that there are no route objects to be included or excluded in the path computation.
2. An empty `'route-object-include-exclude'` list for the primary path of a TE Tunnel Segment indicates that no primary LSP is required for that TE Tunnel.

3. An empty 'route-object-include-exclude' list for a reverse path means it always follows the forward path (i.e., the TE Tunnel is co-routed). When the 'route-object-include-exclude' list is not empty, the reverse path is routed independently of the forward path.
4. An empty 'route-object-include-exclude' list for the secondary (forward) path indicates that the secondary path has the same endpoints as the primary path.

5.1.2. TE Tunnels

The 'tunnels' container holds the list of TE Tunnels that are provisioned on devices in the network (see Figure 5).

A TE Tunnel in the list is uniquely identified by a name. When the model is used to manage a specific device, the 'tunnels' list contains the TE Tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnels' list contains all TE Tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

The TE Tunnel model allows the configuration and management of the following TE tunnel related objects:

TE Tunnel:

A YANG container of one or more LSPs established between the source and destination TE Tunnel termination points. A TE Tunnel LSP is a connection-oriented service provided by the network layer for the delivery of client data between a source and the destination of the TE Tunnel termination points.

TE Tunnel Segment:

A part of a multi-domain TE Tunnel that is within a specific network domain.

```

+--rw tunnels
|   +--rw tunnel* [name]
|   |   +--rw name                               string
|   |   +--rw alias?                             string
|   |   +--rw identifier?                         uint32
|   |   +--rw color?                             uint32
|   |   +--rw description?                       string
|   |   +--ro operational-state?                 identityref
|   |   +--rw encoding?                         identityref
|   |   +--rw switching-type?                   identityref
|   |   +--rw admin-state?                     identityref
|   |   +--rw reoptimize-timer?                 uint16
|   |   +--rw source?                           te-types:te-node-id
|   |   +--rw destination?                     te-types:te-node-id
|   |   +--rw src-tunnel-tp-id?                 binary
|   |   +--rw dst-tunnel-tp-id?                 binary
|   |   +--rw controller
|   |   |   +--rw protocol-origin?               identityref
|   |   |   +--rw controller-entity-id?         string
|   |   +--rw bidirectional?                   boolean
|   |   +--rw association-objects
|   |   |   +--rw association-object* [association-key]
|   |
|   |   // ..
|   |   |
|   |   +--rw protection
|   |
|   |   // ..
|   |   |
|   |   +--rw restoration
|   |
|   |   // ..
|   |   |
|   |   +--rw te-topology-identifier
|   |
|   |   // ..
|   |   |
|   |   +--rw hierarchy
|   |
|   |   // ..

```

Figure 5: TE Tunnel list YANG subtree structure

The TE Tunnel has a number of attributes that are set directly under the tunnel (see Figure 5). The main attributes of a TE Tunnel are described below:

operational-state:

A YANG leaf that holds the operational state of the tunnel.

name:

A YANG leaf that holds the name of a TE Tunnel. The name of the TE Tunnel uniquely identifies the tunnel within the TE tunnel list. The name of the TE Tunnel can be formatted as a Uniform

Resource Indicator (URI) by including the namespace to ensure uniqueness of the name amongst all the TE Tunnels present on devices and controllers.

alias:

A YANG leaf that holds an alternate name to the TE tunnel. Unlike the TE tunnel name, the alias can be modified at any time during the lifetime of the TE tunnel.

identifier:

A YANG leaf that holds an identifier of the tunnel. This identifier is unique amongst tunnels originated from the same ingress device.

color:

A YANG leaf that holds the color associated with the TE tunnel. The color is used to map or steer services that carry matching color on to the TE tunnel as described in [RFC9012].

encoding/switching:

The 'encoding' and 'switching-type' are YANG leafs that define the specific technology in which the tunnel operates in as described in [RFC3945].

reoptimize-timer:

A YANG leaf to set the interval period for tunnel reoptimization.

source/destination:

YANG leafs that define the tunnel source and destination node endpoints.

src-tunnel-tp-id/dst-tunnel-tp-id:

YANG leafs that hold the identifiers of source and destination TE Tunnel Termination Points (TTPs) [RFC8795] residing on the source and destination nodes. The TTP identifiers are optional on nodes that have a single TTP per node. For example, TTP identifiers are optional for packet (IP/MPLS) routers.

controller:

A YANG container that holds tunnel data relevant to an optional external TE controller that may initiate or control a tunnel. This target node may be augmented by external module(s), for example, to add data for PCEP initiated and/or delegated tunnels.

bidirectional:

A YANG leaf that when present indicates the LSPs of a TE Tunnel are bidirectional and co-routed.

association-objects:

A YANG container that holds the set of associations of the TE Tunnel to other TE Tunnels. Associations at the TE Tunnel level apply to all paths of the TE Tunnel. The TE tunnel associations can be overridden by associations configured directly under the TE Tunnel path.

protection:

A YANG container that holds the TE Tunnel protection properties.

restoration:

A YANG container that holds the TE Tunnel restoration properties.

te-topology-identifier:

A YANG container that holds the topology identifier associated with the topology where paths for the TE tunnel are computed.

```

+--rw hierarchy
|   +--rw dependency-tunnels
|   |   +--rw dependency-tunnel* [name]
|   |   |   +--rw name
|   |   |   |   -> ../../../../tunnels/tunnel/name
|   |   |   +--rw encoding?          identityref
|   |   |   +--rw switching-type?    identityref
|   |   +--rw hierarchical-link
|   |   |   +--rw local-te-node-id?    te-types:te-node-id
|   |   |   +--rw local-te-link-tp-id? te-types:te-tp-id
|   |   |   +--rw remote-te-node-id?   te-types:te-node-id
|   |   +--rw te-topology-identifier
|   |   |   +--rw provider-id?    te-global-id
|   |   |   +--rw client-id?     te-global-id
|   |   |   +--rw topology-id?   te-topology-id

```

Figure 6: TE Tunnel hierarchy YANG subtree

hierarchy:

A YANG container that holds hierarchy related properties of the TE Tunnel (see Figure 6. A TE LSP can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used as a TE links to carry traffic in other (client) networks [RFC6107]. In this case, the model introduces the TE Tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the underlying TE Tunnel is associated with. The hierarchy container includes the following:

- o dependency-tunnels: A set of hierarchical TE Tunnels provisioned or to be provisioned in the immediate lower layer that this TE tunnel depends on for multi-layer path computation. A dependency TE Tunnel is provisioned if and only if it is used (selected by path computation) at least by one client layer TE Tunnel. The TE link in the client layer network topology supported by a dependent TE Tunnel is dynamically created only when the dependency TE Tunnel is actually provisioned.
- o hierarchical-link: A YANG container that holds the identity of the hierarchical link (in the client layer) that is supported by this TE Tunnel. The endpoints of the hierarchical link are defined by TE tunnel source and destination node endpoints. The hierarchical link can be identified by its source and destination link termination point identifiers.

5.1.2.1. TE Tunnel Paths

The TE Tunnel can be configured with a set of paths that define the tunnel forward and reverse paths as described in Figure 7. Moreover, a primary path can be specified a set of candidate secondary paths that can be visited to support path protection. The following describe further the list of paths associated with a TE Tunnel.

```

|      +--rw primary-paths
|      |   +--rw primary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      |   +
|      |   +--rw primary-reverse-path
|      |   |   +--rw name?
|      |   |   |   string
|  // ..
|      |   |
|      |   |   +--rw candidate-secondary-reverse-paths
|      |   |   |   +--rw candidate-secondary-reverse-path*
|      |   |   |   |   [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   +--rw candidate-secondary-paths
|      |   |   |   +--rw candidate-secondary-path* [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   |   |   +--ro active?
|      |   |   |   |   |   boolean
|
|      +--rw secondary-paths
|      |   +--rw secondary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      +--rw secondary-reverse-paths
|      |   +--rw secondary-reverse-path* [name]
|      |   |   +--rw name
|      |   |   |   string

```

Figure 7: TE Tunnel paths YANG tree structure

primary-paths:

A YANG container that holds the list of primary paths. A primary path is identified by 'name'. A primary path is selected from the list to instantiate a primary forwarding LSP for the tunnel. The list of primary paths is visited by order of preference. A primary path has the following attributes:

- primary-reverse-path: A YANG container that holds properties of the primary reverse path. The reverse path is applicable to bidirectional TE Tunnels.
- candidate-secondary-paths: A YANG container that holds a list of candidate secondary paths which may be used for the primary path to support path protection. The candidate secondary path(s) reference path(s) from the tunnel secondary paths list. The preference of the secondary paths is specified within the list and dictates the order of visiting the secondary path from the list. The attributes of a secondary path can be defined

separately from the primary path. The attributes of a secondary path will be inherited from the associated 'active' primary when not explicitly defined for the secondary path.

secondary-paths:

A YANG container that holds the set of secondary paths. A secondary path is identified by 'name'. A secondary path can be referenced from the TE Tunnel's 'candidate-secondary-path' list. A secondary path contains attributes similar to a primary path.

secondary-reverse-paths:

A YANG container that holds the set of secondary reverse paths. A secondary reverse path is identified by 'name'. A secondary reverse path can be referenced from the TE Tunnel's 'candidate-secondary-reverse-paths' list. A secondary reverse path contains attributes similar to a primary path.

The following set common path attributes are shared for primary forward and reverse primary and secondary paths:

compute-only:

A path of TE Tunnel is, by default, provisioned so that it can be instantiated in forwarding to carry traffic as soon as a valid path is computed. In some cases, a TE path may be provisioned for the only purpose of computing a path and reporting it without the need to instantiate the LSP or commit any resources. In such a case, the path is configured in 'compute-only' mode to distinguish it from the default behavior. A 'compute-only' path is configured as a usual with the associated per path constraint(s) and properties on a device or TE controller. The device or TE controller computes the feasible path(s) subject to configured constraints. A client may query the 'compute-only' computed path properties 'on-demand', or alternatively, can subscribe to be notified of computed path(s) and whenever the path properties change.

use-path-computation:

A YANG leaf that indicates whether or not path computation is to be used for a specified path.

lockdown:

A YANG leaf that when set indicates the existing path should not be reoptimized after a failure on any of its traversed links.

te-topology-identifier:

A YANG container that holds the topology identifier associated with the tunnel.

optimizations:

a YANG container that holds the optimization objectives that path computation will use to select a path.

computed-paths-properties: > A YANG container that holds properties for the list of computed paths.

computed-path-error-infos:

A YANG container that holds a list of errors related to the path.

lsps:

a YANG container that holds a list of LSPs that are instantiated for this specific path.

5.1.3. TE LSPs

The 'lsps' container includes the set of TE LSP(s) that are instantiated. A TE LSP is identified by a 3-tuple ('tunnel-name', 'node', 'lsp-id').

When the model is used to manage a specific device, the 'lsps' list contains all TE LSP(s) that traverse the device (including ingressing, transiting and egressing the device).

When the model is used to manage a TE controller, the 'lsps' list contains all TE LSP(s) that traverse all network devices (including ingressing, transiting and egressing the device) that the TE controller manages.

5.2. Tree Diagram

Figure 8 shows the tree diagram of the generic TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te
+--rw te!
  +--rw globals
    +--rw named-admin-groups
      +--rw named-admin-group* [name]
        {te-types:extended-admin-groups,te-types:named-extend
ed-admin-groups}?
        +--rw name string
        +--rw bit-position? uint32
    +--rw named-srlgs
      +--rw named-srlg* [name] {te-types:named-srlg-groups}?
        +--rw name string
        +--rw value? te-types:srlg
        +--rw cost? uint32
    +--rw named-path-constraints
      +--rw named-path-constraint* [name]
        {te-types:named-path-constraints}?
        +--rw name string
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
              +--rw generic? te-bandwidth
        +--rw link-protection? identityref
        +--rw setup-priority? uint8
        +--rw hold-priority? uint8
        +--rw signaling-type? identityref
        +--rw path-metric-bounds
          +--rw path-metric-bound* [metric-type]
            +--rw metric-type identityref
            +--rw upper-bound? uint64
        +--rw path-affinities-values
          +--rw path-affinities-value* [usage]
            +--rw usage identityref
            +--rw value? admin-groups
        +--rw path-affinity-names
          +--rw path-affinity-name* [usage]
            +--rw usage identityref
            +--rw affinity-name* [name]
              +--rw name string
        +--rw path-srlgs-lists
          +--rw path-srlgs-list* [usage]
            +--rw usage identityref
            +--rw values* srlg
        +--rw path-srlgs-names
          +--rw path-srlgs-name* [usage]
            +--rw usage identityref
            +--rw names* string
        +--rw disjointness?

```

```

    te-path-disjointness
+--rw explicit-route-objects-always
+--rw route-object-exclude-always* [index]
+--rw index                               uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw hop-type?       te-hop-type
+--rw direction?      te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-type
+--rw direction?      te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number        inet:as-number
+--rw hop-type?       te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
+--:(generic)
+--rw generic?
+--rw rt-types:generalized-label
+--rw direction?
+--rw te-label-direction
+--rw route-object-include-exclude* [index]
+--rw explicit-route-usage?          identityref
+--rw index                          uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw hop-type?       te-hop-type
+--rw direction?      te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop

```



```

    +---rw link-tp-id      te-tp-id
    +---rw node-id         te-node-id
    +---rw hop-type?       te-hop-type
    +---rw direction?      te-link-direction
+---:(as-number)
  +---rw as-number-hop
  +---rw as-number        inet:as-number
  +---rw hop-type?        te-hop-type
+---:(label)
  +---rw label-hop
  +---rw te-label
  +---rw (technology)?
  |   +---:(generic)
  |   |   +---rw generic?
  |   |   |   rt-types:generalized-label
  +---rw direction?
  |   te-label-direction
+---:(srlg)
  +---rw srlg
  +---rw srlg?      uint32
+---rw path-in-segment!
  +---rw label-restrictions
  +---rw label-restriction* [index]
  +---rw restriction?       enumeration
  +---rw index              uint32
  +---rw label-start
  |   +---rw te-label
  |   +---rw (technology)?
  |   |   +---:(generic)
  |   |   |   +---rw generic?
  |   |   |   |   rt-types:generalized-label
  +---rw direction?
  |   te-label-direction
+---rw label-end
  +---rw te-label
  +---rw (technology)?
  |   +---:(generic)
  |   |   +---rw generic?
  |   |   |   rt-types:generalized-label
  +---rw direction?
  |   te-label-direction
+---rw label-step
  +---rw (technology)?
  |   +---:(generic)
  |   |   +---rw generic?      int32
  +---rw range-bitmap?      yang:hex-string
+---rw path-out-segment!
  +---rw label-restrictions

```

```

        +---rw label-restriction* [index]
            +---rw restriction?    enumeration
            +---rw index           uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?    int32
            +---rw range-bitmap?    yang:hex-string
+---rw tunnels
    +---rw tunnel* [name]
        +---rw name                string
        +---rw alias?              string
        +---rw identifier?         uint32
        +---rw color?              uint32
        +---rw description?        string
        +---rw admin-state?        identityref
        +---ro operational-state?   identityref
        +---rw encoding?           identityref
        +---rw switching-type?     identityref
        +---rw source?             te-types:te-node-id
        +---rw destination?        te-types:te-node-id
        +---rw src-tunnel-tp-id?   binary
        +---rw dst-tunnel-tp-id?   binary
        +---rw bidirectional?      boolean
        +---rw controller
            +---rw protocol-origin? identityref
            +---rw controller-entity-id? string
        +---rw reoptimize-timer?   uint16
        +---rw association-objects
            +---rw association-object* [association-key]
                +---rw association-key    string
                +---rw type?              identityref

```

```

+--rw id?                               uint16
+--rw source
  +--rw id?      te-gen-node-id
  +--rw type?    enumeration
+--rw association-object-extended* [association-key]
  +--rw association-key    string
  +--rw type?              identityref
  +--rw id?                uint16
  +--rw source
    +--rw id?      te-gen-node-id
    +--rw type?    enumeration
  +--rw global-source?    uint32
  +--rw extended-id?      yang:hex-string
+--rw protection
  +--rw enable?                                boolean
  +--rw protection-type?                      identityref
  +--rw protection-reversion-disable?         boolean
  +--rw hold-off-time?                        uint32
  +--rw wait-to-revert?                      uint16
  +--rw aps-signal-id?                       uint8
+--rw restoration
  +--rw enable?                                boolean
  +--rw restoration-type?                    identityref
  +--rw restoration-scheme?                 identityref
  +--rw restoration-reversion-disable?       boolean
  +--rw hold-off-time?                      uint32
  +--rw wait-to-restore?                    uint16
  +--rw wait-to-revert?                    uint16
+--rw te-topology-identifier
  +--rw provider-id?    te-global-id
  +--rw client-id?      te-global-id
  +--rw topology-id?    te-topology-id
+--rw te-bandwidth
  +--rw (technology)?
    +--:(generic)
      +--rw generic?    te-bandwidth
+--rw link-protection?                identityref
+--rw setup-priority?                 uint8
+--rw hold-priority?                  uint8
+--rw signaling-type?                 identityref
+--rw hierarchy
  +--rw dependency-tunnels
    +--rw dependency-tunnel* [name]
      +--rw name
        -> /te/tunnels/tunnel/name
      +--rw encoding?              identityref
      +--rw switching-type?        identityref
+--rw hierarchical-link

```

```

+--rw local-te-node-id?          te-types:te-node-id
+--rw local-te-link-tp-id?       te-types:te-tp-id
+--rw remote-te-node-id?         te-types:te-node-id
+--rw te-topology-identifier
  +--rw provider-id?             te-global-id
  +--rw client-id?               te-global-id
  +--rw topology-id?             te-topology-id
+--rw primary-paths
  +--rw primary-path* [name]
    +--rw name                    string
    +--rw path-computation-method? identityref
    +--rw path-computation-server
      +--rw id?                   te-gen-node-id
      +--rw type?                 enumeration
    +--rw compute-only?           empty
    +--rw use-path-computation?   boolean
    +--rw lockdown?               empty
    +--rw path-scope?             identityref
    +--rw preference?             uint8
    +--rw k-requested-paths?      uint8
    +--rw association-objects
      +--rw association-object* [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
      +--rw association-object-extended*
        [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
        +--rw global-source?      uint32
        +--rw extended-id?        yang:hex-string
+--rw optimizations
  +--rw (algorithm)?
    +--:(metric) {path-optimization-metric}?
      +--rw optimization-metric* [metric-type]
        +--rw metric-type
          identityref
        +--rw weight?
          uint8
        +--rw explicit-route-exclude-objects
        +--rw route-object-exclude-object*

```

```

[index]
+---rw index
|      uint32
+---rw (type)?
+---:(numbered-node-hop)
|   +---rw numbered-node-hop
|   |   +---rw node-id
|   |   |   te-node-id
|   |   +---rw hop-type?
|   |       te-hop-type
+---:(numbered-link-hop)
|   +---rw numbered-link-hop
|   |   +---rw link-tp-id
|   |   |   te-tp-id
|   |   +---rw hop-type?
|   |   |   te-hop-type
|   |   +---rw direction?
|   |       te-link-direction
+---:(unnumbered-link-hop)
|   +---rw unnumbered-link-hop
|   |   +---rw link-tp-id
|   |   |   te-tp-id
|   |   +---rw node-id
|   |   |   te-node-id
|   |   +---rw hop-type?
|   |   |   te-hop-type
|   |   +---rw direction?
|   |       te-link-direction
+---:(as-number)
|   +---rw as-number-hop
|   |   +---rw as-number
|   |   |   inet:as-number
|   |   +---rw hop-type?
|   |       te-hop-type
+---:(label)
|   +---rw label-hop
|   |   +---rw te-label
|   |   |   +---rw (technology)?
|   |   |   |   +---:(generic)
|   |   |   |   |   +---rw generic?
|   |   |   |       rt-types:ge
|   |   |   |       +---rw direction?
|   |   |   |       |   te-label-directio
+---:(srlg)
|   +---rw srlg
|   |   +---rw srlg?   uint32

```

```

+---rw explicit-route-include-objects
+---rw route-object-include-object*
    [index]
+---rw index
    |
    |   uint32
+---rw (type)?
+---:(numbered-node-hop)
    |
    |   +---rw numbered-node-hop
    |       |
    |       |   te-node-id
    |       |
    |       +---rw hop-type?
    |           |
    |           |   te-hop-type
+---:(numbered-link-hop)
    |
    |   +---rw numbered-link-hop
    |       |
    |       |   +---rw link-tp-id
    |       |       |
    |       |       |   te-tp-id
    |       |       |
    |       |       +---rw hop-type?
    |       |           |
    |       |           |   te-hop-type
    |       |           +---rw direction?
    |       |               |
    |       |               |   te-link-direction
+---:(unnumbered-link-hop)
    |
    |   +---rw unnumbered-link-hop
    |       |
    |       |   +---rw link-tp-id
    |       |       |
    |       |       |   te-tp-id
    |       |       |
    |       |       +---rw node-id
    |       |           |
    |       |           |   te-node-id
    |       |           +---rw hop-type?
    |       |               |
    |       |               |   te-hop-type
    |       |           +---rw direction?
    |       |               |
    |       |               |   te-link-direction
+---:(as-number)
    |
    |   +---rw as-number-hop
    |       |
    |       |   +---rw as-number
    |       |       |
    |       |       |   inet:as-number
    |       |       +---rw hop-type?
    |       |           |
    |       |           |   te-hop-type
+---:(label)
    |
    |   +---rw label-hop
    |       |
    |       |   +---rw te-label
    |       |       |
    |       |       |   +---rw (technology)?
    |       |       |       |
    |       |       |       |   +---:(generic)
    |       |       |       |       |
    |       |       |       |       |   +---rw generic?
    |       |       |       |           |
    |       |       |       |           |   rt-types:ge
    |       |       |       +---rw direction?
    |       |       |           |
    |       |       |           |   te-label-directio
    |       |       +---rw tiebreakers

```

```

    +---rw tiebreaker* [tiebreaker-type]
    +---rw tiebreaker-type identityref
+---:(objective-function)
    {path-optimization-objective-function}?
    +---rw objective-function
    +---rw objective-function-type?
        identityref
+---rw named-path-constraint? leafref
    {te-types:named-path-constraints}?
+---rw te-bandwidth
    +---rw (technology)?
    +---:(generic)
        +---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
    +---rw path-metric-bound* [metric-type]
    +---rw metric-type identityref
    +---rw upper-bound? uint64
+---rw path-affinities-values
    +---rw path-affinities-value* [usage]
    +---rw usage identityref
    +---rw value? admin-groups
+---rw path-affinity-names
    +---rw path-affinity-name* [usage]
    +---rw usage identityref
    +---rw affinity-name* [name]
    +---rw name string
+---rw path-srlgs-lists
    +---rw path-srlgs-list* [usage]
    +---rw usage identityref
    +---rw values* srlg
+---rw path-srlgs-names
    +---rw path-srlgs-name* [usage]
    +---rw usage identityref
    +---rw names* string
+---rw disjointness?
    te-path-disjointness
+---rw explicit-route-objects-always
    +---rw route-object-exclude-always* [index]
    +---rw index uint32
    +---rw (type)?
    +---:(numbered-node-hop)
        +---rw numbered-node-hop
        +---rw node-id te-node-id
        +---rw hop-type? te-hop-type

```

bel

```

+---:(numbered-link-hop)
|   +---rw numbered-link-hop
|       +---rw link-tp-id      te-tp-id
|       +---rw hop-type?      te-hop-type
|       +---rw direction?     te-link-direction
+---:(unnumbered-link-hop)
|   +---rw unnumbered-link-hop
|       +---rw link-tp-id      te-tp-id
|       +---rw node-id        te-node-id
|       +---rw hop-type?      te-hop-type
|       +---rw direction?     te-link-direction
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number      inet:as-number
|       +---rw hop-type?      te-hop-type
+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?      identityref
+---rw index                      uint32
+---rw (type)?
+---:(numbered-node-hop)
|   +---rw numbered-node-hop
|       +---rw node-id      te-node-id
|       +---rw hop-type?    te-hop-type
+---:(numbered-link-hop)
|   +---rw numbered-link-hop
|       +---rw link-tp-id    te-tp-id
|       +---rw hop-type?    te-hop-type
|       +---rw direction?   te-link-direction
+---:(unnumbered-link-hop)
|   +---rw unnumbered-link-hop
|       +---rw link-tp-id    te-tp-id
|       +---rw node-id      te-node-id
|       +---rw hop-type?    te-hop-type
|       +---rw direction?   te-link-direction
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number    inet:as-number
|       +---rw hop-type?    te-hop-type

```


bel

```

+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---:(srlg)
|   +---rw srlg
|       +---rw srlg?    uint32
+---rw path-in-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?    enumeration
+---rw index            uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-end
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-step
|   +---rw (technology)?
|       +---:(generic)
|           +---rw generic?    int32
+---rw range-bitmap?    yang:hex-string
+---rw path-out-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?    enumeration
+---rw index            uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)

```

```

|         +---rw generic?
|         |         rt-types:generalized-label
|         +---rw direction?
|         |         te-label-direction
+---rw label-end
|   +---rw te-label
|   |   +---rw (technology)?
|   |   |   +---:(generic)
|   |   |   +---rw generic?
|   |   |   |         rt-types:generalized-label
|   |   +---rw direction?
|   |   |         te-label-direction
+---rw label-step
|   +---rw (technology)?
|   +---:(generic)
|   +---rw generic?   int32
+---rw range-bitmap?  yang:hex-string
+---ro computed-paths-properties
|   +---ro computed-path-properties* [k-index]
|   +---ro k-index          uint8
+---ro path-properties
|   +---ro path-metric* [metric-type]
|   |   +---ro metric-type          identityref
|   |   +---ro accumulative-value?  uint64
+---ro path-affinities-values
|   +---ro path-affinities-value* [usage]
|   |   +---ro usage          identityref
|   |   +---ro value?        admin-groups
+---ro path-affinity-names
|   +---ro path-affinity-name* [usage]
|   |   +---ro usage          identityref
|   |   +---ro affinity-name* [name]
|   |   |   +---ro name      string
+---ro path-srlgs-lists
|   +---ro path-srlgs-list* [usage]
|   |   +---ro usage          identityref
|   |   +---ro values*       srlg
+---ro path-srlgs-names
|   +---ro path-srlgs-name* [usage]
|   |   +---ro usage          identityref
|   |   +---ro names*        string
+---ro path-route-objects
|   +---ro path-route-object* [index]
|   |   +---ro index
|   |   |   uint32
|   |   +---ro (type)?
|   |   |   +---:(numbered-node-hop)
|   |   |   |   +---ro numbered-node-hop

```

```

    +--ro node-id      te-node-id
    +--ro hop-type?
        te-hop-type
    +--:(numbered-link-hop)
        +--ro numbered-link-hop
        +--ro link-tp-id      te-tp-id
        +--ro hop-type?
            |
            te-hop-type
        +--ro direction?
            te-link-direction
    +--:(unnumbered-link-hop)
        +--ro unnumbered-link-hop
        +--ro link-tp-id      te-tp-id
        +--ro node-id
            |
            te-node-id
        +--ro hop-type?
            |
            te-hop-type
        +--ro direction?
            te-link-direction
    +--:(as-number)
        +--ro as-number-hop
        +--ro as-number
            |
            inet:as-number
        +--ro hop-type?
            te-hop-type
    +--:(label)
        +--ro label-hop
        +--ro te-label
            +--ro (technology)?
                |
                +--:(generic)
                |
                +--ro generic?
                    rt-types:gener
alized-label
        +--ro direction?
            te-label-direction
    +--ro te-bandwidth
        +--ro (technology)?
            +--:(generic)
            +--ro generic?      te-bandwidth
    +--ro disjointness-type?
        te-types:te-path-disjointness
+--ro computed-path-error-infos
    +--ro computed-path-error-info* []
        +--ro error-description?      string
        +--ro error-timestamp?         yang:date-and-time
        +--ro error-reason?            identityref
+--ro lsp-provisioning-error-infos
    +--ro lsp-provisioning-error-info* []

```

```

+--ro error-description?  string
+--ro error-timestamp?    yang:date-and-time
+--ro error-node-id?      te-types:te-node-id
+--ro error-link-id?      te-types:te-tp-id
+--ro lsp-id?             uint16
+--ro lsps
  +--ro lsp* [node lsp-id]
    +--ro tunnel-name?
      |      -> /te/lsps/lsp/tunnel-name
    +--ro node      -> /te/lsps/lsp/node
    +--ro lsp-id    -> /te/lsps/lsp/lsp-id
+--rw primary-reverse-path
  +--rw name?                                string
  +--rw path-computation-method?
    |      identityref
  +--rw path-computation-server
    +--rw id?      te-gen-node-id
    +--rw type?    enumeration
  +--rw compute-only?                                empty
  +--rw use-path-computation?
    |      boolean
  +--rw lockdown?                                empty
  +--ro path-scope?
    |      identityref
  +--rw association-objects
    +--rw association-object* [association-key]
      +--rw association-key  string
      +--rw type?            identityref
      +--rw id?              uint16
      +--rw source
        +--rw id?      te-gen-node-id
        +--rw type?    enumeration
    +--rw association-object-extended*
      [association-key]
      +--rw association-key  string
      +--rw type?            identityref
      +--rw id?              uint16
      +--rw source
        +--rw id?      te-gen-node-id
        +--rw type?    enumeration
      +--rw global-source?   uint32
      +--rw extended-id?    yang:hex-string
  +--rw optimizations
    +--rw (algorithm)?
      +--:(metric) {path-optimization-metric}?
        +--rw optimization-metric* [metric-type]
          +--rw metric-type
            |      identityref

```

```

+---rw weight?
|   uint8
+---rw explicit-route-exclude-objects
|   +---rw route-object-exclude-object*
|       [index]
|       +---rw index
|           |   uint32
|       +---rw (type)?
|           +---:(numbered-node-hop)
|               +---rw numbered-node-hop
|                   +---rw node-id
|                       |   te-node-id
|                   +---rw hop-type?
|                       |   te-hop-type
|           +---:(numbered-link-hop)
|               +---rw numbered-link-hop
|                   +---rw link-tp-id
|                       |   te-tp-id
|                   +---rw hop-type?
|                       |   te-hop-type
|                   +---rw direction?
|                       |   te-link-direction
|           +---:(unnumbered-link-hop)
|               +---rw unnumbered-link-hop
|                   +---rw link-tp-id
|                       |   te-tp-id
|                   +---rw node-id
|                       |   te-node-id
|                   +---rw hop-type?
|                       |   te-hop-type
|                   +---rw direction?
|                       |   te-link-direction
|           +---:(as-number)
|               +---rw as-number-hop
|                   +---rw as-number
|                       |   inet:as-number
|                   +---rw hop-type?
|                       |   te-hop-type
|           +---:(label)
|               +---rw label-hop
|                   +---rw te-label
|                       +---rw (technology)?
|                           +---:(generic)
|                               +---rw generic?
|                                   rt-types
:generalized-label
+---rw direction?
    te-label-direc

```

tion

```

+--:(srlg)
+--rw srlg
+--rw srlg?   uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
    [index]
+--rw index
    |
    |   uint32
+--rw (type)?
+--:(numbered-node-hop)
    |
    |   +--rw numbered-node-hop
    |       |
    |       |   +--rw node-id
    |       |       |
    |       |       |   te-node-id
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-type
+--:(numbered-link-hop)
    |
    |   +--rw numbered-link-hop
    |       |
    |       |   +--rw link-tp-id
    |       |       |
    |       |       |   te-tp-id
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-type
    |       +--rw direction?
    |           |
    |           |   te-link-direction
+--:(unnumbered-link-hop)
    |
    |   +--rw unnumbered-link-hop
    |       |
    |       |   +--rw link-tp-id
    |       |       |
    |       |       |   te-tp-id
    |       |   +--rw node-id
    |       |       |
    |       |       |   te-node-id
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-type
    |       +--rw direction?
    |           |
    |           |   te-link-direction
+--:(as-number)
    |
    |   +--rw as-number-hop
    |       |
    |       |   +--rw as-number
    |       |       |
    |       |       |   inet:as-number
    |       |   +--rw hop-type?
    |       |       |
    |       |       |   te-hop-type
+--:(label)
    |
    |   +--rw label-hop
    |       |
    |       |   +--rw te-label
    |       |       |
    |       |       |   +--rw (technology)?
    |       |       |       |
    |       |       |       |   +--:(generic)
    |       |       |       |       |
    |       |       |       |       |   +--rw generic?
    |       |       |       |       |       |
    |       |       |       |       |       |   rt-types
:generalized-label

```

```

        +---rw direction?
            te-label-direct
tion
        +---rw tiebreakers
            +---rw tiebreaker* [tiebreaker-type]
                +---rw tiebreaker-type
                    identityref
        +---:(objective-function)
            {path-optimization-objective-function
}?:
        +---rw objective-function
            +---rw objective-function-type?
                identityref
        +---rw named-path-constraint?                leafref
            {te-types:named-path-constraints}?
        +---rw te-bandwidth
            +---rw (technology)?
                +---:(generic)
                    +---rw generic?    te-bandwidth
        +---rw link-protection?
            identityref
        +---rw setup-priority?                        uint8
        +---rw hold-priority?                        uint8
        +---rw signaling-type?
            identityref
        +---rw path-metric-bounds
            +---rw path-metric-bound* [metric-type]
                +---rw metric-type    identityref
                +---rw upper-bound?   uint64
        +---rw path-affinities-values
            +---rw path-affinities-value* [usage]
                +---rw usage          identityref
                +---rw value?        admin-groups
        +---rw path-affinity-names
            +---rw path-affinity-name* [usage]
                +---rw usage          identityref
                +---rw affinity-name* [name]
                    +---rw name      string
        +---rw path-srlgs-lists
            +---rw path-srlgs-list* [usage]
                +---rw usage          identityref
                +---rw values*        srlg
        +---rw path-srlgs-names
            +---rw path-srlgs-name* [usage]
                +---rw usage          identityref
                +---rw names*         string
        +---rw disjointness?
            te-path-disjointness

```

```

+---rw explicit-route-objects-always
+---rw route-object-exclude-always* [index]
+---rw index                               uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id         te-node-id
+---rw hop-type?      te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id      te-tp-id
+---rw hop-type?      te-hop-type
+---rw direction?
+---rw                    te-link-direction
+---:(unnumbered-link-hop)
+---rw unnumbered-link-hop
+---rw link-tp-id      te-tp-id
+---rw node-id         te-node-id
+---rw hop-type?      te-hop-type
+---rw direction?
+---rw                    te-link-direction
+---:(as-number)
+---rw as-number-hop
+---rw as-number       inet:as-number
+---rw hop-type?      te-hop-type
+---:(label)
+---rw label-hop
+---rw te-label
+---rw (technology)?
+---:(generic)
+---rw generic?
+---rw                    rt-types:generalized
- label
+---rw direction?
+---rw                    te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?
+---rw                    identityref
+---rw index                               uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id         te-node-id
+---rw hop-type?      te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id      te-tp-id
+---rw hop-type?      te-hop-type

```


					<pre> +---rw direction? te-link-direction +---:(unnumbered-link-hop) +---rw unnumbered-link-hop +---rw link-tp-id te-tp-id +---rw node-id te-node-id +---rw hop-type? te-hop-type +---rw direction? te-link-direction +---:(as-number) +---rw as-number-hop +---rw as-number inet:as-number +---rw hop-type? te-hop-type +---:(label) +---rw label-hop +---rw te-label +---rw (technology)? +---:(generic) +---rw generic? rt-types:generalized </pre>
-label					<pre> +---rw direction? te-label-direction +---:(srlg) +---rw srlg +---rw srlg? uint32 +---rw path-in-segment! +---rw label-restrictions +---rw label-restriction* [index] +---rw restriction? enumeration +---rw index uint32 +---rw label-start +---rw te-label +---rw (technology)? +---:(generic) +---rw generic? rt-types:generalized-la </pre>
bel					<pre> +---rw direction? te-label-direction +---rw label-end +---rw te-label +---rw (technology)? +---:(generic) +---rw generic? rt-types:generalized-la </pre>
bel					<pre> +---rw direction? </pre>

					te-label-direction
				+--rw label-step	
				+--rw (technology)?	
				+--:(generic)	
				+--rw generic?	int32
				+--rw range-bitmap?	yang:hex-string
			+--rw path-out-segment!		
			+--rw label-restrictions		
			+--rw label-restriction* [index]		
			+--rw restriction?		enumeration
			+--rw index		uint32
			+--rw label-start		
			+--rw te-label		
			+--rw (technology)?		
			+--:(generic)		
			+--rw generic?		rt-types:generalized-la
bel					
				+--rw direction?	
				te-label-direction	
			+--rw label-end		
			+--rw te-label		
			+--rw (technology)?		
			+--:(generic)		
			+--rw generic?		rt-types:generalized-la
bel					
				+--rw direction?	
				te-label-direction	
			+--rw label-step		
			+--rw (technology)?		
			+--:(generic)		
			+--rw generic?		int32
			+--rw range-bitmap?		yang:hex-string
		+--ro computed-paths-properties			
		+--ro computed-path-properties* [k-index]			
		+--ro k-index			uint8
		+--ro path-properties			
		+--ro path-metric* [metric-type]			
		+--ro metric-type			
		identityref			
		+--ro accumulative-value?			uint64
		+--ro path-affinities-values			
		+--ro path-affinities-value* [usage]			
		+--ro usage			identityref
		+--ro value?			admin-groups
		+--ro path-affinity-names			
		+--ro path-affinity-name* [usage]			

```

+---ro usage          identityref
+---ro affinity-name* [name]
    +---ro name        string
+---ro path-srlgs-lists
    +---ro path-srlgs-list* [usage]
        +---ro usage    identityref
        +---ro values*   srlg
+---ro path-srlgs-names
    +---ro path-srlgs-name* [usage]
        +---ro usage    identityref
        +---ro names*   string
+---ro path-route-objects
    +---ro path-route-object* [index]
        +---ro index
            |
            uint32
        +---ro (type)?
            +---:(numbered-node-hop)
                +---ro numbered-node-hop
                    +---ro node-id
                        |
                        te-node-id
                    +---ro hop-type?
                        |
                        te-hop-type
            +---:(numbered-link-hop)
                +---ro numbered-link-hop
                    +---ro link-tp-id
                        |
                        te-tp-id
                    +---ro hop-type?
                        |
                        te-hop-type
                    +---ro direction?
                        |
                        te-link-direction
            +---:(unnumbered-link-hop)
                +---ro unnumbered-link-hop
                    +---ro link-tp-id
                        |
                        te-tp-id
                    +---ro node-id
                        |
                        te-node-id
                    +---ro hop-type?
                        |
                        te-hop-type
                    +---ro direction?
                        |
                        te-link-direction
            +---:(as-number)
                +---ro as-number-hop
                    +---ro as-number
                        |
                        inet:as-number
                    +---ro hop-type?
                        |
                        te-hop-type
            +---:(label)
                +---ro label-hop

```

```

+--ro te-label
+--ro (technology)?
+--:(generic)
+--ro generic?
rt-types:ge

neralized-label

+--ro direction?
te-label-directio

n

+--ro te-bandwidth
+--ro (technology)?
+--:(generic)
+--ro generic? te-bandwidth
+--ro disjointness-type?
te-types:te-path-disjointness
+--ro computed-path-error-infos
+--ro computed-path-error-info* []
+--ro error-description? string
+--ro error-timestamp?
| yang:date-and-time
+--ro error-reason? identityref
+--ro lsp-provisioning-error-infos
+--ro lsp-provisioning-error-info* []
+--ro error-description? string
+--ro error-timestamp?
| yang:date-and-time
+--ro error-node-id?
| te-types:te-node-id
+--ro error-link-id?
| te-types:te-tp-id
+--ro lsp-id? uint16
+--ro lsps
+--ro lsp* [node lsp-id]
+--ro tunnel-name?
| -> /te/lsps/lsp/tunnel-name
+--ro node -> /te/lsps/lsp/node
+--ro lsp-id -> /te/lsps/lsp/lsp-id
+--rw candidate-secondary-reverse-paths
+--rw candidate-secondary-reverse-path*
| [secondary-path]
+--rw secondary-path leafref
+--rw candidate-secondary-paths
+--rw candidate-secondary-path* [secondary-path]
+--rw secondary-path leafref
+--ro active? boolean
+--rw secondary-paths
+--rw secondary-path* [name]
+--rw name string

```

```

+--rw path-computation-method?          identityref
+--rw path-computation-server
|   +--rw id?      te-gen-node-id
|   +--rw type?    enumeration
+--rw compute-only?                      empty
+--rw use-path-computation?              boolean
+--rw lockdown?                          empty
+--ro path-scope?                        identityref
+--rw preference?                        uint8
+--rw association-objects
|   +--rw association-object* [association-key]
|   |   +--rw association-key    string
|   |   +--rw type?             identityref
|   |   +--rw id?               uint16
|   |   +--rw source
|   |   |   +--rw id?      te-gen-node-id
|   |   |   +--rw type?    enumeration
|   +--rw association-object-extended*
|   |   [association-key]
|   |   +--rw association-key    string
|   |   +--rw type?             identityref
|   |   +--rw id?               uint16
|   |   +--rw source
|   |   |   +--rw id?      te-gen-node-id
|   |   |   +--rw type?    enumeration
|   +--rw global-source?          uint32
|   +--rw extended-id?            yang:hex-string
+--rw optimizations
|   +--rw (algorithm)?
|   |   +--:(metric) {path-optimization-metric}?
|   |   |   +--rw optimization-metric* [metric-type]
|   |   |   |   +--rw metric-type
|   |   |   |   |   identityref
|   |   |   |   +--rw weight?
|   |   |   |   |   uint8
|   |   |   +--rw explicit-route-exclude-objects
|   |   |   |   +--rw route-object-exclude-object*
|   |   |   |   |   [index]
|   |   |   |   |   +--rw index
|   |   |   |   |   |   uint32
|   |   |   |   +--rw (type)?
|   |   |   |   |   +--:(numbered-node-hop)
|   |   |   |   |   |   +--rw numbered-node-hop
|   |   |   |   |   |   |   +--rw node-id
|   |   |   |   |   |   |   |   te-node-id
|   |   |   |   |   |   |   +--rw hop-type?
|   |   |   |   |   |   |   |   te-hop-type
|   |   |   |   +--:(numbered-link-hop)

```

```

+--rw numbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
|   +--rw (technology)?
|   |   +--:(generic)
|   |   +--rw generic?
|   |   rt-types:ge
+--rw direction?
|   te-label-direction
+--:(srlg)
+--rw srlg
|   +--rw srlg?   uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
|   +--rw node-id
|   |   te-node-id
+--rw hop-type?

```

						te-hop-type
					+	---:(numbered-link-hop)
						+---rw numbered-link-hop
						+---rw link-tp-id
						te-tp-id
						+---rw hop-type?
						te-hop-type
						+---rw direction?
						te-link-direction
					+	---:(unnumbered-link-hop)
						+---rw unnumbered-link-hop
						+---rw link-tp-id
						te-tp-id
						+---rw node-id
						te-node-id
						+---rw hop-type?
						te-hop-type
						+---rw direction?
						te-link-direction
					+	---:(as-number)
						+---rw as-number-hop
						+---rw as-number
						inet:as-number
						+---rw hop-type?
						te-hop-type
					+	---:(label)
						+---rw label-hop
						+---rw te-label
						+---rw (technology)?
						+---:(generic)
						+---rw generic?
						rt-types:ge
neralized-label						
						+---rw direction?
						te-label-directio
n						
						+---rw tiebreakers
						+---rw tiebreaker* [tiebreaker-type]
						+---rw tiebreaker-type identityref
					+	---:(objective-function)
						{path-optimization-objective-function}?
						+---rw objective-function
						+---rw objective-function-type?
						identityref
					+	---rw named-path-constraint? leafref
						{te-types:named-path-constraints}?
					+	---rw te-bandwidth
						+---rw (technology)?

```

    +---:(generic)
      +---rw generic?    te-bandwidth
+---rw link-protection?          identityref
+---rw setup-priority?          uint8
+---rw hold-priority?           uint8
+---rw signaling-type?          identityref
+---rw path-metric-bounds
  +---rw path-metric-bound* [metric-type]
  +---rw metric-type          identityref
  +---rw upper-bound?         uint64
+---rw path-affinities-values
  +---rw path-affinities-value* [usage]
  +---rw usage                 identityref
  +---rw value?                admin-groups
+---rw path-affinity-names
  +---rw path-affinity-name* [usage]
  +---rw usage                 identityref
  +---rw affinity-name* [name]
  +---rw name                  string
+---rw path-srlgs-lists
  +---rw path-srlgs-list* [usage]
  +---rw usage                 identityref
  +---rw values*              srlg
+---rw path-srlgs-names
  +---rw path-srlgs-name* [usage]
  +---rw usage                 identityref
  +---rw names*               string
+---rw disjointness?
  +---rw te-path-disjointness
+---rw explicit-route-objects-always
  +---rw route-object-exclude-always* [index]
  +---rw index                  uint32
  +---rw (type)?
    +---:(numbered-node-hop)
      +---rw numbered-node-hop
      +---rw node-id            te-node-id
      +---rw hop-type?          te-hop-type
    +---:(numbered-link-hop)
      +---rw numbered-link-hop
      +---rw link-tp-id         te-tp-id
      +---rw hop-type?          te-hop-type
      +---rw direction?         te-link-direction
    +---:(unnumbered-link-hop)
      +---rw unnumbered-link-hop
      +---rw link-tp-id         te-tp-id
      +---rw node-id            te-node-id
      +---rw hop-type?          te-hop-type
      +---rw direction?         te-link-direction

```


				<pre> +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--rw route-object-include-exclude* [index] +--rw explicit-route-usage? identityref +--rw index uint32 +--rw (type)? +--:(numbered-node-hop) +--rw numbered-node-hop +--rw node-id te-node-id +--rw hop-type? te-hop-type +--:(numbered-link-hop) +--rw numbered-link-hop +--rw link-tp-id te-tp-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(unnumbered-link-hop) +--rw unnumbered-link-hop +--rw link-tp-id te-tp-id +--rw node-id te-node-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--:(srlg) </pre>

```

        +---rw srlg
            +---rw srlg?   uint32
+---rw path-in-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?   int32
            +---rw range-bitmap?   yang:hex-string
+---rw path-out-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?

```

```

|                                     te-label-direction
|                                     +---rw label-step
|                                     |   +---rw (technology)?
|                                     |   +---:(generic)
|                                     |   +---rw generic?    int32
|                                     +---rw range-bitmap?   yang:hex-string
+---rw protection
|   +---rw enable?                                boolean
|   +---rw protection-type?                       identityref
|   +---rw protection-reversion-disable?          boolean
|   +---rw hold-off-time?                         uint32
|   +---rw wait-to-revert?                       uint16
|   +---rw aps-signal-id?                        uint8
+---rw restoration
|   +---rw enable?                                boolean
|   +---rw restoration-type?
|   |   identityref
|   +---rw restoration-scheme?
|   |   identityref
|   +---rw restoration-reversion-disable?         boolean
|   +---rw hold-off-time?                       uint32
|   +---rw wait-to-restore?                     uint16
|   +---rw wait-to-revert?                     uint16
+---ro computed-paths-properties
|   +---ro computed-path-properties* [k-index]
|   |   +---ro k-index                        uint8
|   |   +---ro path-properties
|   |   |   +---ro path-metric* [metric-type]
|   |   |   |   +---ro metric-type            identityref
|   |   |   |   +---ro accumulative-value?    uint64
|   |   |   +---ro path-affinities-values
|   |   |   |   +---ro path-affinities-value* [usage]
|   |   |   |   |   +---ro usage              identityref
|   |   |   |   |   +---ro value?            admin-groups
|   |   |   +---ro path-affinity-names
|   |   |   |   +---ro path-affinity-name* [usage]
|   |   |   |   |   +---ro usage              identityref
|   |   |   |   |   +---ro affinity-name* [name]
|   |   |   |   |   |   +---ro name          string
|   |   +---ro path-srlgs-lists
|   |   |   +---ro path-srlgs-list* [usage]
|   |   |   |   +---ro usage              identityref
|   |   |   |   +---ro values*          srlg
|   |   +---ro path-srlgs-names
|   |   |   +---ro path-srlgs-name* [usage]
|   |   |   |   +---ro usage              identityref
|   |   |   |   +---ro names*          string
|   +---ro path-route-objects

```

				+--ro path-route-object* [index]
				+--ro index
				uint32
				+--ro (type)?
				+--:(numbered-node-hop)
				+--ro numbered-node-hop
				+--ro node-id te-node-id
				+--ro hop-type?
				te-hop-type
				+--:(numbered-link-hop)
				+--ro numbered-link-hop
				+--ro link-tp-id te-tp-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(unnumbered-link-hop)
				+--ro unnumbered-link-hop
				+--ro link-tp-id te-tp-id
				+--ro node-id
				te-node-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(as-number)
				+--ro as-number-hop
				+--ro as-number
				inet:as-number
				+--ro hop-type?
				te-hop-type
				+--:(label)
				+--ro label-hop
				+--ro te-label
				+--ro (technology)?
				+--:(generic)
				+--ro generic?
				rt-types:gener
				+--ro direction?
				te-label-direction
				+--ro te-bandwidth
				+--ro (technology)?
				+--:(generic)
				+--ro generic? te-bandwidth
				+--ro disjointness-type?
				te-types:te-path-disjointness
				+--ro computed-path-error-infos

alized-label

```

    +--ro computed-path-error-info* []
      +--ro error-description?  string
      +--ro error-timestamp?    yang:date-and-time
      +--ro error-reason?       identityref
+--ro lsp-provisioning-error-infos
  +--ro lsp-provisioning-error-info* []
    +--ro error-description?  string
    +--ro error-timestamp?    yang:date-and-time
    +--ro error-node-id?      te-types:te-node-id
    +--ro error-link-id?      te-types:te-tp-id
    +--ro lsp-id?              uint16
+--ro lsps
  +--ro lsp* [node lsp-id]
    +--ro tunnel-name?
      |      -> /te/lsps/lsp/tunnel-name
    +--ro node      -> /te/lsps/lsp/node
    +--ro lsp-id     -> /te/lsps/lsp/lsp-id
+--rw secondary-reverse-paths
  +--rw secondary-reverse-path* [name]
    +--rw name                      string
    +--rw path-computation-method?  identityref
    +--rw path-computation-server
      +--rw id?      te-gen-node-id
      +--rw type?    enumeration
    +--rw compute-only?              empty
    +--rw use-path-computation?      boolean
    +--rw lockdown?                  empty
    +--ro path-scope?                identityref
    +--rw preference?                uint8
    +--rw association-objects
      +--rw association-object* [association-key]
        +--rw association-key      string
        +--rw type?                identityref
        +--rw id?                  uint16
        +--rw source
          +--rw id?      te-gen-node-id
          +--rw type?    enumeration
      +--rw association-object-extended*
        [association-key]
        +--rw association-key      string
        +--rw type?                identityref
        +--rw id?                  uint16
        +--rw source
          +--rw id?      te-gen-node-id
          +--rw type?    enumeration
        +--rw global-source?        uint32
        +--rw extended-id?          yang:hex-string
+--rw optimizations

```

```

+--rw (algorithm)?
+--:(metric) {path-optimization-metric}?
+--rw optimization-metric* [metric-type]
+--rw metric-type
|   identityref
+--rw weight?
|   uint8
+--rw explicit-route-exclude-objects
+--rw route-object-exclude-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
|   +--:(generic)

```

```

+---rw generic?
      rt-types:generic
n
+---rw direction?
      te-label-direction
+---:(srlg)
  +---rw srlg
  +---rw srlg?   uint32
+---rw explicit-route-include-objects
  +---rw route-object-include-object*
    [index]
  +---rw index
    |   uint32
  +---rw (type)?
    +---:(numbered-node-hop)
      +---rw numbered-node-hop
      +---rw node-id
      |   te-node-id
      +---rw hop-type?
      |   te-hop-type
    +---:(numbered-link-hop)
      +---rw numbered-link-hop
      +---rw link-tp-id
      |   te-tp-id
      +---rw hop-type?
      |   te-hop-type
      +---rw direction?
      |   te-link-direction
    +---:(unnumbered-link-hop)
      +---rw unnumbered-link-hop
      +---rw link-tp-id
      |   te-tp-id
      +---rw node-id
      |   te-node-id
      +---rw hop-type?
      |   te-hop-type
      +---rw direction?
      |   te-link-direction
    +---:(as-number)
      +---rw as-number-hop
      +---rw as-number
      |   inet:as-number
      +---rw hop-type?
      |   te-hop-type
    +---:(label)
      +---rw label-hop
      +---rw te-label

```

```

+---rw (technology)?
+---:(generic)
+---rw generic?
rt-types:generic
n
+---rw direction?
te-label-direction
+---rw tiebreakers
+---rw tiebreaker* [tiebreaker-type]
+---rw tiebreaker-type identityref
+---:(objective-function)
{path-optimization-objective-function}?
+---rw objective-function
+---rw objective-function-type?
identityref
+---rw named-path-constraint? leafref
{te-types:named-path-constraints}?
+---rw te-bandwidth
+---rw (technology)?
+---:(generic)
+---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
+---rw path-metric-bound* [metric-type]
+---rw metric-type identityref
+---rw upper-bound? uint64
+---rw path-affinities-values
+---rw path-affinities-value* [usage]
+---rw usage identityref
+---rw value? admin-groups
+---rw path-affinity-names
+---rw path-affinity-name* [usage]
+---rw usage identityref
+---rw affinity-name* [name]
+---rw name string
+---rw path-srlgs-lists
+---rw path-srlgs-list* [usage]
+---rw usage identityref
+---rw values* srlg
+---rw path-srlgs-names
+---rw path-srlgs-name* [usage]
+---rw usage identityref
+---rw names* string
+---rw disjointness?

```


				te-path-disjointness
			+--rw	explicit-route-objects-always
			+--rw	route-object-exclude-always* [index]
			+--rw	index uint32
			+--rw	(type)?
			+--:((numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--:((numbered-link-hop)
			+--rw	numbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:((unnumbered-link-hop)
			+--rw	unnumbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:((as-number)
			+--rw	as-number-hop
			+--rw	as-number inet:as-number
			+--rw	hop-type? te-hop-type
			+--:((label)
			+--rw	label-hop
			+--rw	te-label
			+--rw	(technology)?
			+--:((generic)
			+--rw	generic?
				rt-types:generalized-la
			+--rw	direction?
				te-label-direction
			+--rw	route-object-include-exclude* [index]
			+--rw	explicit-route-usage? identityref
			+--rw	index uint32
			+--rw	(type)?
			+--:((numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--:((numbered-link-hop)
			+--rw	numbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:((unnumbered-link-hop)

				<pre> +--rw unnumbered-link-hop +--rw link-tp-id te-tp-id +--rw node-id te-node-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--:(srlg) +--rw srlg +--rw srlg? uint32 +--rw path-in-segment! +--rw label-restrictions +--rw label-restriction* [index] +--rw restriction? enumeration +--rw index uint32 +--rw label-start +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-end +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-step +--rw (technology)? +--:(generic) +--rw generic? int32 +--rw range-bitmap? yang:hex-string </pre>

```

+--rw path-out-segment!
  +--rw label-restrictions
    +--rw label-restriction* [index]
      +--rw restriction?      enumeration
      +--rw index             uint32
      +--rw label-start
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-end
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-step
        +--rw (technology)?
          +--:(generic)
            +--rw generic?    int32
        +--rw range-bitmap?   yang:hex-string
+--rw protection
  +--rw enable?                boolean
  +--rw protection-type?       identityref
  +--rw protection-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-revert?        uint16
  +--rw aps-signal-id?         uint8
+--rw restoration
  +--rw enable?                boolean
  +--rw restoration-type?
    | identityref
  +--rw restoration-scheme?
    | identityref
  +--rw restoration-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-restore?       uint16
  +--rw wait-to-revert?        uint16
+--ro computed-paths-properties
  +--ro computed-path-properties* [k-index]
    +--ro k-index              uint8
  +--ro path-properties
    +--ro path-metric* [metric-type]

```

```

|   +--ro metric-type          identityref
|   +--ro accumulative-value?  uint64
+--ro path-affinities-values
|   +--ro path-affinities-value* [usage]
|   +--ro usage                identityref
|   +--ro value?               admin-groups
+--ro path-affinity-names
|   +--ro path-affinity-name* [usage]
|   +--ro usage                identityref
|   +--ro affinity-name* [name]
|   +--ro name                 string
+--ro path-srlgs-lists
|   +--ro path-srlgs-list* [usage]
|   +--ro usage                identityref
|   +--ro values*             srlg
+--ro path-srlgs-names
|   +--ro path-srlgs-name* [usage]
|   +--ro usage                identityref
|   +--ro names*              string
+--ro path-route-objects
|   +--ro path-route-object* [index]
|   +--ro index
|   |   uint32
|   +--ro (type)?
|   |   +--:(numbered-node-hop)
|   |   |   +--ro numbered-node-hop
|   |   |   |   +--ro node-id      te-node-id
|   |   |   |   +--ro hop-type?
|   |   |   |   |   te-hop-type
|   |   |   +--:(numbered-link-hop)
|   |   |   |   +--ro numbered-link-hop
|   |   |   |   |   +--ro link-tp-id  te-tp-id
|   |   |   |   |   +--ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +--ro direction?
|   |   |   |   |   |   te-link-direction
|   |   |   +--:(unnumbered-link-hop)
|   |   |   |   +--ro unnumbered-link-hop
|   |   |   |   |   +--ro link-tp-id  te-tp-id
|   |   |   |   |   +--ro node-id
|   |   |   |   |   |   te-node-id
|   |   |   |   |   +--ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +--ro direction?
|   |   |   |   |   |   te-link-direction
|   |   +--:(as-number)
|   |   |   +--ro as-number-hop
|   |   |   +--ro as-number

```

```

|                                     |      inet:as-number
|                                     |      +---ro hop-type?
|                                     |          te-hop-type
|         +---:(label)
|             +---ro label-hop
|                 +---ro te-label
|                     +---ro (technology)?
|                         |--:(generic)
|                             +---ro generic?
|                                 rt-types:gener
alized-label
|                                                     +---ro direction?
|                                                         te-label-direction
|           +---ro te-bandwidth
|               +---ro (technology)?
|                   |--:(generic)
|                       +---ro generic?    te-bandwidth
+---ro disjointness-type?
        te-types:te-path-disjointness
+---ro computed-path-error-infos
|   +---ro computed-path-error-info* []
|       +---ro error-description?     string
|       +---ro error-timestamp?       yang:date-and-time
|       +---ro error-reason?          identityref
+---ro lsp-provisioning-error-infos
|   +---ro lsp-provisioning-error-info* []
|       +---ro error-description?     string
|       +---ro error-timestamp?       yang:date-and-time
|       +---ro error-node-id?         te-types:te-node-id
|       +---ro error-link-id?         te-types:te-tp-id
|       +---ro lsp-id?                uint16
+---ro lsps
|   +---ro lsp* [node lsp-id]
|       +---ro tunnel-name?
|           | -> /te/lsp/lsp/tunnel-name
|       +---ro node                    -> /te/lsp/lsp/node
|       +---ro lsp-id                  -> /te/lsp/lsp/lsp-id
+----x tunnel-action
|   +---w input
|       | +---w action-type?     identityref
+---ro output
|       +---ro action-result?     identityref
+----x protection-external-commands
|   +---w input
|       +---w protection-external-command?
|           | identityref
+---w protection-group-ingress-node-id?
|       te-types:te-node-id
```

```

|         +---w protection-group-egress-node-id?
|         |         te-types:te-node-id
|         +---w path-ref?                                path-ref
|         +---w traffic-type?
|         |         enumeration
|         +---w extra-traffic-tunnel-ref?                tunnel-ref
+---ro lsp
  +---ro lsp* [tunnel-name lsp-id node]
    +---ro tunnel-name                                string
    +---ro lsp-id                                    uint16
    +---ro node
      |         te-types:te-node-id
    +---ro source?
      |         te-types:te-node-id
    +---ro destination?
      |         te-types:te-node-id
    +---ro tunnel-id?                                uint16
    +---ro extended-tunnel-id?                        yang:dotted-quad
    +---ro operational-state?                          identityref
    +---ro signaling-type?                             identityref
    +---ro origin-type?                                enumeration
    +---ro lsp-resource-status?                         enumeration
    +---ro lockout-of-normal?                           boolean
    +---ro freeze?                                     boolean
    +---ro lsp-protection-role?                         enumeration
    +---ro lsp-protection-state?                       identityref
    +---ro protection-group-ingress-node-id?
      |         te-types:te-node-id
    +---ro protection-group-egress-node-id?
      |         te-types:te-node-id
    +---ro lsp-record-route-information
      +---ro lsp-record-route-information* [index]
        +---ro index                                uint32
        +---ro (type)?
          +---:(numbered-node-hop)
            +---ro numbered-node-hop
              +---ro node-id          te-node-id
              +---ro flags*           path-attribute-flags
          +---:(numbered-link-hop)
            +---ro numbered-link-hop
              +---ro link-tp-id       te-tp-id
              +---ro flags*           path-attribute-flags
          +---:(unnumbered-link-hop)
            +---ro unnumbered-link-hop
              +---ro link-tp-id       te-tp-id
              +---ro node-id?         te-node-id
              +---ro flags*           path-attribute-flags
          +---:(label)

```

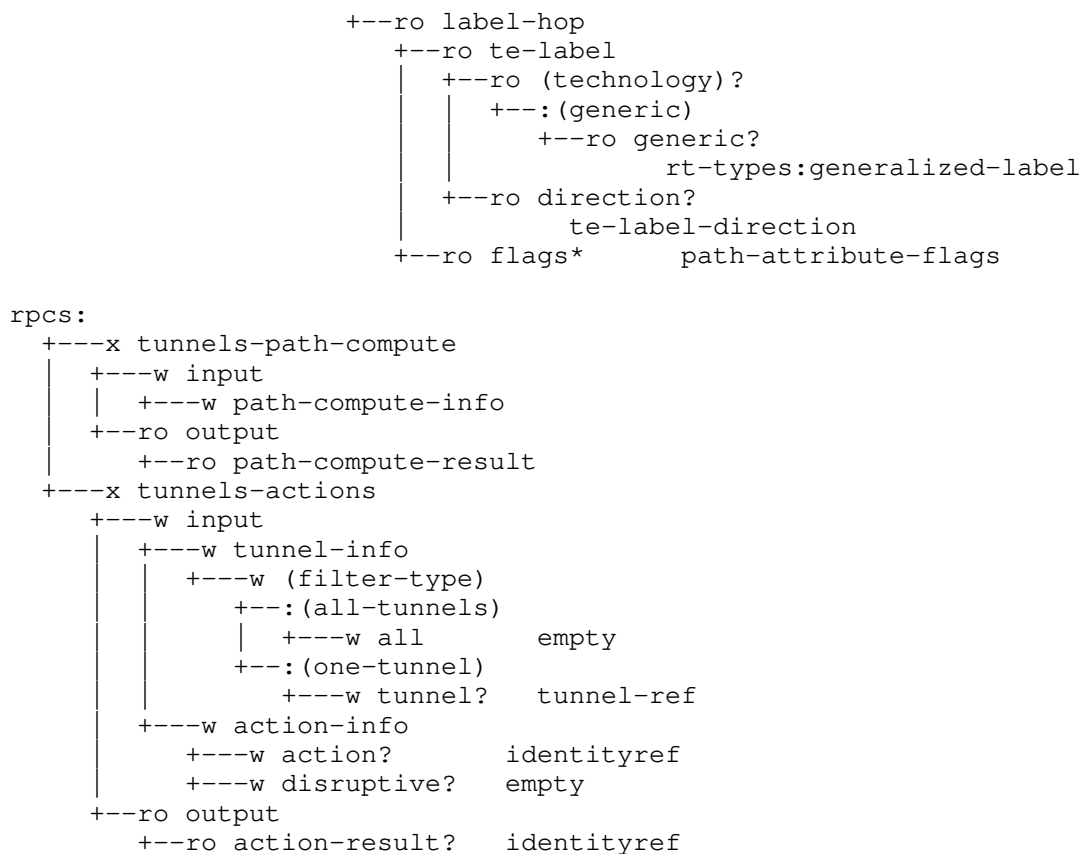


Figure 8: TE Tunnel generic model YANG tree diagram

5.3. YANG Module

The generic TE YANG module 'ietf-te' imports the following modules:

- ```
* ietf-yang-types and ietf-inet-types defined in [RFC6991]
* ietf-te-types defined in [RFC8776]
```

This module references the following documents: [RFC6991], [RFC4875], [RFC7551], [RFC4206], [RFC4427], [RFC4872], [RFC3945], [RFC3209], [RFC6780], [RFC8800], and [RFC7308].

```
<CODE BEGINS> file "ietf-te@2021-10-22.yang"
module ietf-te {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te";

 /* Replace with IANA when assigned */

 prefix te;

 /* Import TE generic types */

 import ietf-te-types {
 prefix te-types;
 reference
 "RFC8776: Common YANG Data Types for Traffic Engineering.";
 }
 import ietf-inet-types {
 prefix inet;
 reference
 "RFC6991: Common YANG Data Types.";
 }
 import ietf-yang-types {
 prefix yang;
 reference
 "RFC6991: Common YANG Data Types.";
 }
}

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group.";
contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 Editor: Tarek Saad
 <mailto:tsaad@juniper.net>

 Editor: Rakesh Gandhi
 <mailto:rgandhi@cisco.com>

 Editor: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editor: Himanshu Shah
 <mailto:hshah@ciena.com>

 Editor: Xufeng Liu
 <mailto:xufeng.liu.ietf@gmail.com>
```



```
Editor: Igor Bryskin
 <mailto:i_bryskin@yahoo.com>;
description
 "YANG data module for TE configuration, state, and RPCs.
 The model fully conforms to the Network Management
 Datastore Architecture (NMDA).

 Copyright (c) 2019 IETF Trust and the persons
 identified as authors of the code. All rights reserved.

 Redistribution and use in source and binary forms, with or
 without modification, is permitted pursuant to, and subject
 to the license terms contained in, the Simplified BSD License
 set forth in Section 4.c of the IETF Trust's Legal Provisions
 Relating to IETF Documents
 (https://trustee.ietf.org/license-info).
 This version of this YANG module is part of RFC XXXX; see
 the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision 2021-10-22 {
 description
 "Latest update to TE generic YANG module.";
 reference
 "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
 and Interfaces.";
}

identity path-computation-error-reason {
 description
 "Base identity for path computation error reasons.";
}

identity path-computation-error-no-topology {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no topology
 with the provided topology-identifier.";
}

identity path-computation-error-no-dependent-server {
 base path-computation-error-reason;
 description
 "Path computation has failed because one or more dependent
```

```
 path computation servers are unavailable.
 The dependent path computation server could be
 a Backward-Recursive Path Computation (BRPC) downstream
 PCE or a child PCE.";
 reference
 "RFC5441, RFC8685";
}

identity path-computation-error-pce-unavailable {
 base path-computation-error-reason;
 description
 "Path computation has failed because PCE is not available.";
 reference
 "RFC5440";
}

identity path-computation-error-no-inclusion-hop {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no
 node or link provided by one or more inclusion hops.";
 reference
 "RFC8685";
}

identity path-computation-error-destination-unknown-in-domain {
 base path-computation-error-reason;
 description
 "Path computation has failed because the destination node is
 unknown in indicated destination domain.";
 reference
 "RFC8685";
}

identity path-computation-error-no-resource {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no
 available resource in one or more domains.";
 reference
 "RFC8685";
}

identity path-computation-error-child-pce-unresponsive {
 base path-computation-error-reason;
 description
 "Path computation has failed because child PCE is not
 responsive.";
```

```
 reference
 "RFC8685";
 }

 identity path-computation-error-destination-domain-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because the destination domain
 was unknown.";
 reference
 "RFC8685";
 }

 identity path-computation-error-p2mp {
 base path-computation-error-reason;
 description
 "Path computation has failed because of P2MP reachability
 problem.";
 reference
 "RFC8306";
 }

 identity path-computation-error-no-gco-migration {
 base path-computation-error-reason;
 description
 "Path computation has failed because of no Global Concurrent
 Optimization (GCO) migration path found.";
 reference
 "RFC5557";
 }

 identity path-computation-error-no-gco-solution {
 base path-computation-error-reason;
 description
 "Path computation has failed because of no GCO solution
 found.";
 reference
 "RFC5557";
 }

 identity path-computation-error-path-not-found {
 base path-computation-error-reason;
 description
 "Path computation no path found error reason.";
 reference
 "RFC5440";
 }
```

```
identity path-computation-error-pks-expansion {
 base path-computation-error-reason;
 description
 "Path computation has failed because of Path-Key Subobject
 (PKS) expansion failure.";
 reference
 "RFC5520";
}

identity path-computation-error-brpc-chain-unavailable {
 base path-computation-error-reason;
 description
 "Path computation has failed because PCE BRPC chain
 unavailable.";
 reference
 "RFC5441";
}

identity path-computation-error-source-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because source node is unknown.";
 reference
 "RFC5440";
}

identity path-computation-error-destination-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because destination node is
 unknown.";
 reference
 "RFC5440";
}

identity path-computation-error-no-server {
 base path-computation-error-reason;
 description
 "Path computation has failed because path computation
 server is unavailable.";
 reference
 "RFC5440";
}

identity tunnel-actions-type {
 description
 "TE tunnel actions type.";
}
```

```
identity tunnel-action-reoptimize {
 base tunnel-actions-type;
 description
 "Reoptimize tunnel action type.";
}

identity tunnel-admin-auto {
 base te-types:tunnel-admin-state-type;
 description
 "Tunnel administrative auto state. The administrative status
 in state datastore transitions to 'tunnel-admin-up' when the
 tunnel used by the client layer, and to 'tunnel-admin-down'
 when it is not used by the client layer.";
}

identity association-type-diversity {
 base te-types:association-type;
 description
 "Association Type diversity used to associate LSPs whose paths
 are to be diverse from each other.";
 reference
 "RFC8800";
}

identity protocol-origin-type {
 description
 "Base identity for protocol origin type.";
}

identity protocol-origin-api {
 base protocol-origin-type;
 description
 "Protocol origin is via Application Programmable Interface
 (API).";
}

identity protocol-origin-pcep {
 base protocol-origin-type;
 description
 "Protocol origin is Path Computation Engine Protocol (PCEP).";
 reference "RFC5440";
}

identity protocol-origin-bgp {
 base protocol-origin-type;
 description
 "Protocol origin is Border Gateway Protocol (BGP).";
 reference "RFC5512";
}

typedef tunnel-ref {
```

```
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te:name";
 }
 description
 "This type is used by data models that need to reference
 configured TE tunnel.";
 }

 typedef path-ref {
 type union {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/"
 + "te:primary-paths/te:primary-path/te:name";
 }
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/"
 + "te:secondary-paths/te:secondary-path/te:name";
 }
 }
 description
 "This type is used by data models that need to reference
 configured primary or secondary path of a TE tunnel.";
 }

 typedef te-gen-node-id {
 type union {
 type te-types:te-node-id;
 type inet:ip-address;
 }
 description
 "Generic type that identifies a node in a TE topology.";
 }

 /**
 * TE tunnel generic groupings
 */

 grouping te-generic-node-id {
 description
 "A reusable grouping for a TE generic node identifier.";
 leaf id {
 type te-gen-node-id;
 description
 "The identifier of the node. Can be represented as IP
 address or dotted quad address.";
 }
 leaf type {
 type enumeration {
```

```
 enum ip {
 description
 "IP address representation of the node identifier.";
 }
 enum dotted-quad {
 description
 "Dotted quad address representation of the node
 identifier.";
 }
 }
 description
 "Type of node identifier representation.";
}

grouping primary-path {
 description
 "The tunnel primary path properties.";
 uses path-common-properties;
 uses path-preference;
 uses k-requested-paths;
 uses path-compute-info;
 uses path-state;
}

grouping primary-reverse-path {
 description
 "The tunnel primary reverse path properties.";
 reference
 "RFC7551";
 uses path-common-properties;
 uses path-compute-info;
 uses path-state;
}

grouping secondary-path {
 description
 "The tunnel secondary path properties.";
 uses path-common-properties;
 uses path-preference;
 uses path-compute-info;
 uses protection-restoration-properties;
 uses path-state;
}

grouping secondary-reverse-path {
 description
 "The tunnel secondary reverse path properties.";
```

```
 uses path-common-properties;
 uses path-preference;
 uses path-compute-info;
 uses protection-restoration-properties;
 uses path-state;
}

grouping path-common-properties {
 description
 "Common path attributes.";
 leaf name {
 type string;
 description
 "TE path name.";
 }
 leaf path-computation-method {
 type identityref {
 base te-types:path-computation-method;
 }
 default "te-types:path-locally-computed";
 description
 "The method used for computing the path, either
 locally computed, queried from a server or not
 computed at all (explicitly configured).";
 }
 container path-computation-server {
 when "derived-from-or-self(..path-computation-method, "
 + "'te-types:path-externally-queried') " {
 description
 "The path-computation server when the path is
 externally queried.";
 }
 uses te-generic-node-id;
 description
 "Address of the external path computation
 server.";
 }
 leaf compute-only {
 type empty;
 description
 "When set, the path is computed and updated whenever
 the topology is updated. No resources are committed
 or reserved in the network.";
 }
 leaf use-path-computation {
 when "derived-from-or-self(..path-computation-method, "
 + "'te-types:path-locally-computed') " {
 type boolean;
 }
 }
}
```



```
 default "true";
 description
 "When 'true' indicates the path is dynamically computed
 and/or validated against the Traffic-Engineering Database
 (TED), and when 'false' indicates no validation against
 the TED is required.";
}
leaf lockdown {
 type empty;
 description
 "Indicates no reoptimization to be attempted for this path.";
}
leaf path-scope {
 type identityref {
 base te-types:path-scope-type;
 }
 default "te-types:path-scope-end-to-end";
 config false;
 description
 "Path scope if segment or an end-to-end path.";
}
}

/* This grouping will be re-used in path-computation rpc */

grouping path-compute-info {
 description
 "Attributes used for path computation request.";
 uses tunnel-associations-properties;
 uses te-types:generic-path-optimization;
 leaf named-path-constraint {
 if-feature "te-types:named-path-constraints";
 type leafref {
 path "/te:te/te:globals/te:named-path-constraints/"
 + "te:named-path-constraint/te:name";
 }
 description
 "Reference to a globally defined named path constraint set.";
 }
 uses path-constraints-common;
}

/* This grouping will be re-used in path-computation rpc */

grouping path-preference {
 description
 "The path preference.";
 leaf preference {
```

```
 type uint8 {
 range "1..255";
 }
 default "1";
 description
 "Specifies a preference for this path. The lower the number
 higher the preference.";
 }
}

/* This grouping will be re-used in path-computation rpc */

grouping k-requested-paths {
 description
 "The k-shortest paths requests.";
 leaf k-requested-paths {
 type uint8;
 default "1";
 description
 "The number of k-shortest-paths requested from the path
 computation server and returned sorted by its optimization
 objective. The value 0 all possible paths.";
 }
}

grouping path-properties {
 description
 "TE computed path properties grouping.";
 uses te-types:generic-path-properties {
 augment "path-properties" {
 description
 "additional path properties returned by path computation.";
 uses te-types:te-bandwidth;
 leaf disjointness-type {
 type te-types:te-path-disjointness;
 config false;
 description
 "The type of resource disjointness.
 When reported for a primary path, it represents the
 minimum level of disjointness of all the secondary
 paths.
 When reported for a secondary path, it represents the
 disjointness of the secondary path.";
 }
 }
 }
}
```

```
grouping path-state {
 description
 "TE per path state parameters.";
 uses path-computation-response;
 uses lsp-provisioning-error-info {
 augment "lsp-provisioning-error-infos/"
 + "lsp-provisioning-error-info" {
 description
 "Augmentation of LSP provisioning information under a
 specific path.";
 leaf lsp-id {
 type uint16;
 description
 "The LSP-ID for which path computation was performed.";
 }
 }
 }
}
container lsps {
 config false;
 description
 "The TE LSPs container.";
 list lsp {
 key "node lsp-id";
 description
 "List of LSPs associated with the tunnel.";
 leaf tunnel-name {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:tunnel-name";
 }
 description "TE tunnel name.";
 }
 leaf node {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:node";
 }
 description "The node where the LSP state resides on.";
 }
 leaf lsp-id {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:lsp-id";
 }
 description "The TE LSP identifier.";
 }
 }
}

/* This grouping will be re-used in path-computation rpc */
```

```
grouping path-computation-response {
 description
 "Attributes reported by path computation response.";
 container computed-paths-properties {
 config false;
 description
 "Computed path properties container.";
 list computed-path-properties {
 key "k-index";
 description
 "List of computed paths.";
 leaf k-index {
 type uint8;
 description
 "The k-th path returned from the computation server.
 A lower k value path is more optimal than higher k
 value path(s)";
 }
 uses path-properties {
 description
 "The TE path computed properties.";
 }
 }
 }
}
container computed-path-error-infos {
 config false;
 description
 "Path computation information container.";
 list computed-path-error-info {
 description
 "List of path computation info entries.";
 leaf error-description {
 type string;
 description
 "Textual representation of the error occurred during
 path computation.";
 }
 leaf error-timestamp {
 type yang:date-and-time;
 description
 "Timestamp of last path computation attempt.";
 }
 leaf error-reason {
 type identityref {
 base path-computation-error-reason;
 }
 description
 "Reason for the path computation error.";
 }
 }
}
```

```
 }
 }
}

grouping lsp-provisioning-error-info {
 description
 "Grouping for LSP provisioning error information.";
 container lsp-provisioning-error-infos {
 config false;
 description
 "LSP provisioning error information.";
 list lsp-provisioning-error-info {
 description
 "List of LSP provisioning error info entries.";
 leaf error-description {
 type string;
 description
 "Textual representation of the error occurred during
 path computation.";
 }
 leaf error-timestamp {
 type yang:date-and-time;
 description
 "Timestamp of when the reported error occurred.";
 }
 leaf error-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Node identifier of node where error occurred.";
 }
 leaf error-link-id {
 type te-types:te-tp-id;
 default "0";
 description
 "Link ID where the error occurred.";
 }
 }
 }
}

grouping protection-restoration-properties-state {
 description
 "Protection parameters grouping.";
 leaf lockout-of-normal {
 type boolean;
 default "false";
 }
}
```

```
description
 "When set to 'True', it represents a lockout of normal
 traffic external command. When set to 'False', it
 represents a clear lockout of normal traffic external
 command. The lockout of normal traffic command applies
 to this Tunnel.";
reference
 "RFC4427";
}
leaf freeze {
 type boolean;
 default "false";
 description
 "When set to 'True', it represents a freeze external command.
 When set to 'False', it represents a clear freeze external
 command. The freeze command applies to all the Tunnels which
 are sharing the protection resources with this Tunnel.";
 reference
 "RFC4427";
}
leaf lsp-protection-role {
 type enumeration {
 enum working {
 description
 "A working LSP must be a primary LSP whilst a protecting
 LSP can be either a primary or a secondary LSP. Also,
 known as protected LSPs when working LSPs are associated
 with protecting LSPs.";
 }
 enum protecting {
 description
 "A secondary LSP is an LSP that has been provisioned
 in the control plane only; e.g. resource allocation
 has not been committed at the data plane.";
 }
 }
 default "working";
 description
 "LSP role type.";
 reference
 "RFC4872, section 4.2.1";
}
leaf lsp-protection-state {
 type identityref {
 base te-types:lsp-protection-state;
 }
 default "te-types:normal";
 description
```

```
 "The state of the APS state machine controlling which
 tunnels is using the resources of the protecting LSP.";
 }
 leaf protection-group-ingress-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Indicates the te-node-id of the protection group
 ingress node when the APS state represents an external
 command (LoP, SF, MS) applied to it or a WTR timer
 running on it. If the external command is not applied to
 the ingress node or the WTR timer is not running on it,
 this attribute is not specified. A value 0.0.0.0 is used
 when the te-node-id of the protection group ingress node is
 unknown (e.g., because the ingress node is outside the scope
 of control of the server)";
 }
 leaf protection-group-egress-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Indicates the te-node-id of the protection group egress node
 when the APS state represents an external command (LoP, SF,
 MS) applied to it or a WTR timer running on it. If the
 external command is not applied to the ingress node or
 the WTR timer is not running on it, this attribute is not
 specified. A value 0.0.0.0 is used when the te-node-id of
 the protection group ingress node is unknown (e.g., because
 the ingress node is outside the scope of control of the
 server)";
 }
}

grouping protection-restoration-properties {
 description
 "Protection and restoration parameters.";
 container protection {
 description
 "Protection parameters.";
 leaf enable {
 type boolean;
 default "false";
 description
 "A flag to specify if LSP protection is enabled.";
 reference
 "RFC4427";
 }
 leaf protection-type {
```

```
 type identityref {
 base te-types:lsp-protection-type;
 }
 default "te-types:lsp-protection-unprotected";
 description
 "LSP protection type.";
 }
 leaf protection-reversion-disable {
 type boolean;
 default "false";
 description
 "Disable protection reversion to working path.";
 }
 leaf hold-off-time {
 type uint32;
 units "milli-seconds";
 default "0";
 description
 "The time between the declaration of an SF or SD condition
 and the initialization of the protection switching
 algorithm.";
 reference
 "RFC4427";
 }
 leaf wait-to-revert {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP reversion.";
 reference
 "RFC4427";
 }
 leaf aps-signal-id {
 type uint8 {
 range "1..255";
 }
 default "1";
 description
 "The APS signal number used to reference the traffic of
 this tunnel. The default value for normal traffic is 1.
 The default value for extra-traffic is 255. If not
 specified, non-default values can be assigned by the
 server, if and only if, the server controls both
 endpoints.";
 reference
 "RFC4427";
 }
}
```



```
container restoration {
 description
 "Restoration parameters.";
 leaf enable {
 type boolean;
 default "false";
 description
 "A flag to specify if LSP restoration is enabled.";
 reference
 "RFC4427";
 }
 leaf restoration-type {
 type identityref {
 base te-types:lsp-restoration-type;
 }
 default "te-types:lsp-restoration-restore-any";
 description
 "LSP restoration type.";
 }
 leaf restoration-scheme {
 type identityref {
 base te-types:restoration-scheme-type;
 }
 default "te-types:restoration-scheme-preconfigured";
 description
 "LSP restoration scheme.";
 }
 leaf restoration-reversion-disable {
 type boolean;
 default "false";
 description
 "Disable restoration reversion to working path.";
 }
 leaf hold-off-time {
 type uint32;
 units "milli-seconds";
 description
 "The time between the declaration of an SF or SD condition
 and the initialization of the protection switching
 algorithm.";
 reference
 "RFC4427";
 }
 leaf wait-to-restore {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP restoration.";
```

```
 reference
 "RFC4427";
 }
 leaf wait-to-revert {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP reversion.";
 reference
 "RFC4427";
 }
 }
 }
}

grouping tunnel-associations-properties {
 description
 "TE tunnel association grouping.";
 container association-objects {
 description
 "TE tunnel associations.";
 list association-object {
 key "association-key";
 unique "type id source/id source/type";
 description
 "List of association base objects.";
 reference
 "RFC4872";
 leaf association-key {
 type string;
 description
 "Association key used to identify a specific
 association in the list";
 }
 leaf type {
 type identityref {
 base te-types:association-type;
 }
 description
 "Association type.";
 reference
 "RFC4872";
 }
 leaf id {
 type uint16;
 description
 "Association identifier.";
 reference
 "RFC4872";
 }
 }
 }
}
```

```
 }
 container source {
 uses te-generic-node-id;
 description
 "Association source.";
 reference
 "RFC4872";
 }
 }
 list association-object-extended {
 key "association-key";
 unique
 "type id source/id source/type global-source extended-id";
 description
 "List of extended association objects.";
 reference
 "RFC6780";
 leaf association-key {
 type string;
 description
 "Association key used to identify a specific
 association in the list";
 }
 leaf type {
 type identityref {
 base te-types:association-type;
 }
 description
 "Association type.";
 reference
 "RFC4872, RFC6780";
 }
 leaf id {
 type uint16;
 description
 "Association identifier.";
 reference
 "RFC4872, RFC6780";
 }
 }
 container source {
 uses te-generic-node-id;
 description
 "Association source.";
 reference
 "RFC4872, RFC6780";
 }
 leaf global-source {
 type uint32;
```

```
 description
 "Association global source.";
 reference
 "RFC6780";
 }
 leaf extended-id {
 type yang:hex-string;
 description
 "Association extended identifier.";
 reference
 "RFC6780";
 }
}
}

/* TE tunnel configuration/state grouping */
/* These grouping will be re-used in path-computation rpc */

grouping encoding-and-switching-type {
 description
 "Common grouping to define the LSP encoding and
 switching types";
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "LSP encoding type.";
 reference
 "RFC3945";
 }
 leaf switching-type {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "LSP switching type.";
 reference
 "RFC3945";
 }
}

grouping tunnel-common-attributes {
 description
 "Common grouping to define the TE tunnel parameters";
 leaf source {
 type te-types:te-node-id;
```

```
 description
 "TE tunnel source node ID.";
 }
 leaf destination {
 type te-types:te-node-id;
 description
 "TE tunnel destination node identifier.";
 }
 leaf src-tunnel-tp-id {
 type binary;
 description
 "TE tunnel source termination point identifier.";
 }
 leaf dst-tunnel-tp-id {
 type binary;
 description
 "TE tunnel destination termination point identifier.";
 }
 leaf bidirectional {
 type boolean;
 default "false";
 description
 "Indicates a bidirectional co-routed LSP.";
 }
}

grouping tunnel-hierarchy-properties {
 description
 "A grouping for TE tunnel hierarchy information.";
 container hierarchy {
 description
 "Container for TE hierarchy related information.";
 container dependency-tunnels {
 description
 "List of tunnels that this tunnel can be potentially
 dependent on.";
 list dependency-tunnel {
 key "name";
 description
 "A tunnel entry that this tunnel can potentially depend
 on.";
 leaf name {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te:name";
 require-instance false;
 }
 description
 "Dependency tunnel name. The tunnel may not have been
```

```
 instantiated yet.";
 }
 uses encoding-and-switching-type;
}
}
container hierarchical-link {
 description
 "Identifies a hierarchical link (in client layer)
 that this tunnel is associated with.";
 reference
 "RFC4206";
 leaf local-te-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "The local TE node identifier.";
 }
 leaf local-te-link-tp-id {
 type te-types:te-tp-id;
 default "0";
 description
 "The local TE link termination point identifier.";
 }
 leaf remote-te-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Remote TE node identifier.";
 }
 uses te-types:te-topology-identifier {
 description
 "The topology identifier where the hierarchical link
 supported by this TE tunnel is instantiated.";
 }
}
}
}

grouping tunnel-properties {
 description
 "Top level grouping for tunnel properties.";
 leaf name {
 type string;
 description
 "TE tunnel name.";
 }
 leaf alias {
 type string;
 }
}
```

```
 description
 "An alternate name of the TE tunnel that can be modified
 anytime during its lifetime.";
 }
 leaf identifier {
 type uint32;
 description
 "TE tunnel Identifier.";
 reference
 "RFC3209";
 }
 leaf color {
 type uint32;
 description "The color associated with the TE tunnel.";
 reference "RFC9012";
 }
 leaf description {
 type string;
 default "None";
 description
 "Textual description for this TE tunnel.";
 }
 leaf admin-state {
 type identityref {
 base te-types:tunnel-admin-state-type;
 }
 default "te-types:tunnel-admin-state-up";
 description
 "TE tunnel administrative state.";
 }
 leaf operational-state {
 type identityref {
 base te-types:tunnel-state-type;
 }
 config false;
 description
 "TE tunnel operational state.";
 }
 uses encoding-and-switching-type;
 uses tunnel-common-attributes;
 container controller {
 description
 "Contains tunnel data relevant to external controller(s).
 This target node may be augmented by external module(s),
 for example, to add data for PCEP initiated and/or
 delegated tunnels.";
 leaf protocol-origin {
 type identityref {
```

```
 base protocol-origin-type;
 }
 description
 "The protocol origin for instantiating the tunnel.";
}
leaf controller-entity-id {
 type string;
 description
 "An identifier unique within the scope of visibility that
 associated with the entity that controls the tunnel";
 reference "RFC8232";
}
leaf reoptimize-timer {
 type uint16;
 units "seconds";
 description
 "Frequency of reoptimization of a traffic engineered LSP.";
}
uses tunnel-associations-properties;
uses protection-restoration-properties;
uses te-types:tunnel-constraints;
uses tunnel-hierarchy-properties;
container primary-paths {
 description
 "The set of primary paths.";
 list primary-path {
 key "name";
 description
 "List of primary paths for this tunnel.";
 uses primary-path;
 container primary-reverse-path {
 description
 "The reverse primary path properties.";
 uses primary-reverse-path;
 container candidate-secondary-reverse-paths {
 description
 "The set of referenced candidate reverse secondary
 paths from the full set of secondary reverse paths
 which may be used for this primary path.";
 list candidate-secondary-reverse-path {
 key "secondary-path";
 ordered-by user;
 description
 "List of candidate secondary reverse path(s)";
 leaf secondary-path {
 type leafref {
 path "../.../.../.../.../..."
 }
 }
 }
 }
 }
 }
}
```



```

 + "te:secondary-reverse-paths/"
 + "te:secondary-reverse-path/te:name";
 }
 description
 "A reference to the secondary reverse path that
 should be utilised when the containing primary
 reverse path option is in use.";
 }
}
}
}
container candidate-secondary-paths {
 description
 "The set of candidate secondary paths which may be used
 for this primary path. When secondary paths are
 specified in the list the path of the secondary LSP in
 use must be restricted to those path options referenced.
 The priority of the secondary paths is specified within
 the list. Higher priority values are less preferred -
 that is to say that a path with priority 0 is the most
 preferred path. In the case that the list is empty, any
 secondary path option may be utilised when the current
 primary path is in use.";
 list candidate-secondary-path {
 key "secondary-path";
 ordered-by user;
 description
 "List of candidate secondary paths for this tunnel.";
 leaf secondary-path {
 type leafref {
 path "../../../../../te:secondary-paths/"
 + "te:secondary-path/te:name";
 }
 description
 "A reference to the secondary path that should be
 utilised when the containing primary path option is
 in use.";
 }
 leaf active {
 type boolean;
 config false;
 description
 "Indicates the current active path option that has
 been selected of the candidate secondary paths.";
 }
 }
}
}
}

```

```
 }
 container secondary-paths {
 description
 "The set of secondary paths.";
 list secondary-path {
 key "name";
 description
 "List of secondary paths for this tunnel.";
 uses secondary-path;
 }
 }
 container secondary-reverse-paths {
 description
 "The set of secondary reverse paths.";
 list secondary-reverse-path {
 key "name";
 description
 "List of secondary paths for this tunnel.";
 uses secondary-reverse-path;
 }
 }
 }

 grouping tunnel-actions {
 description
 "Tunnel actions.";
 action tunnel-action {
 description
 "Tunnel action.";
 input {
 leaf action-type {
 type identityref {
 base tunnel-actions-type;
 }
 description
 "Tunnel action type.";
 }
 }
 output {
 leaf action-result {
 type identityref {
 base te-types:te-action-result;
 }
 description
 "The result of the tunnel action operation.";
 }
 }
 }
 }
}
```

```
}

grouping tunnel-protection-actions {
 description
 "Protection external command actions.";
 action protection-external-commands {
 input {
 leaf protection-external-command {
 type identityref {
 base te-types:protection-external-commands;
 }
 description
 "Protection external command.";
 }
 leaf protection-group-ingress-node-id {
 type te-types:te-node-id;
 description
 "When specified, indicates whether the action is
 applied on ingress node.
 By default, if neither ingress nor egress node-id
 is set, the action applies to ingress node only.";
 }
 leaf protection-group-egress-node-id {
 type te-types:te-node-id;
 description
 "When specified, indicates whether the action is
 applied on egress node.
 By default, if neither ingress nor egress node-id
 is set, the action applies to ingress node only.";
 }
 leaf path-ref {
 type path-ref;
 description
 "Indicates to which path the external command applies
 to.";
 }
 leaf traffic-type {
 type enumeration {
 enum normal-traffic {
 description
 "The manual-switch or forced-switch command applies
 to the normal traffic (this Tunnel).";
 }
 enum null-traffic {
 description
 "The manual-switch or forced-switch command applies
 to the null traffic.";
 }
 }
 }
 }
 }
}
```

```
 enum extra-traffic {
 description
 "The manual-switch or forced-switch command applies
 to the extra traffic (the extra-traffic Tunnel
 sharing protection bandwidth with this Tunnel).";
 }
 }
 description
 "Indicates whether the manual-switch or forced-switch
 commands applies to the normal traffic, the null traffic
 or the extra-traffic.";
 reference
 "RFC4427";
}
leaf extra-traffic-tunnel-ref {
 type tunnel-ref;
 description
 "In case there are multiple extra-traffic tunnels sharing
 protection bandwidth with this Tunnel (m:n protection),
 represents which extra-traffic Tunnel the manual-switch
 or forced-switch to extra-traffic command applies to.";
}
}
}

/** End of TE tunnel groupings */
/**
 * LSP related generic groupings
 */

grouping lsp-record-route-information-state {
 description
 "LSP Recorded route information grouping.";
 container lsp-record-route-information {
 description
 "RSVP recorded route object information.";
 list lsp-record-route-information {
 when "../origin-type = 'ingress'" {
 description
 "Applicable on ingress LSPs only.";
 }
 }
 key "index";
 description
 "Record route list entry.";
 uses te-types:record-route-state;
 }
}
```

```
}

grouping lsps-grouping {
 description
 "LSPs state operational data grouping.";
 container lsps {
 config false;
 description
 "TE LSPs state container.";
 list lsp {
 key "tunnel-name lsp-id node";
 unique "source destination tunnel-id lsp-id "
 + "extended-tunnel-id";
 description
 "List of LSPs associated with the tunnel.";
 leaf tunnel-name {
 type string;
 description "The TE tunnel name.";
 }
 leaf lsp-id {
 type uint16;
 description
 "Identifier used in the SENDER_TEMPLATE and the
 FILTER_SPEC that can be changed to allow a sender to
 share resources with itself.";
 reference
 "RFC3209";
 }
 leaf node {
 type te-types:te-node-id;
 description
 "The node where the TE LSP state resides on.";
 }
 uses lsp-properties-state;
 uses lsp-record-route-information-state;
 }
 }
}

/**** End of TE LSP groupings ****/
/**
 * TE global generic groupings
 */
/* Global named admin-groups configuration data */

grouping named-admin-groups-properties {
 description
 "Global named administrative groups configuration
```

```
 grouping.";
 leaf name {
 type string;
 description
 "A string name that uniquely identifies a TE
 interface named admin-group.";
 }
 leaf bit-position {
 type uint32;
 description
 "Bit position representing the administrative group.";
 reference
 "RFC3209 and RFC7308";
 }
}

grouping named-admin-groups {
 description
 "Global named administrative groups configuration
 grouping.";
 container named-admin-groups {
 description
 "TE named admin groups container.";
 list named-admin-group {
 if-feature "te-types:extended-admin-groups";
 if-feature "te-types:named-extended-admin-groups";
 key "name";
 description
 "List of named TE admin-groups.";
 uses named-admin-groups-properties;
 }
 }
}

/* Global named admin-srlgs configuration data */

grouping named-srlgs {
 description
 "Global named SRLGs configuration grouping.";
 container named-srlgs {
 description
 "TE named SRLGs container.";
 list named-srlg {
 if-feature "te-types:named-srlg-groups";
 key "name";
 description
 "A list of named SRLG groups.";
 leaf name {
```

```
 type string;
 description
 "A string name that uniquely identifies a TE
 interface named SRLG.";
 }
 leaf value {
 type te-types:srlg;
 description
 "An SRLG value.";
 }
 leaf cost {
 type uint32;
 description
 "SRLG associated cost. Used during path to append
 the path cost when traversing a link with this SRLG.";
 }
}
}
}

/* Global named paths constraints configuration data */

grouping path-constraints-common {
 description
 "Global named path constraints configuration
 grouping.";
 uses te-types:common-path-constraints-attributes {
 description
 "The constraints applicable to the path. This includes:
 - The path bandwidth constraint
 - The path link protection type constraint
 - The path setup/hold priority constraint
 - path signaling type constraint
 - path metric bounds constraint. The unit of path metric
 bound is interpreted in the context of the metric-type.
 For example for metric-type 'path-metric-loss', the bound
 is multiples of the basic unit 0.000003% as described
 in RFC7471 for OSPF, and RFC8570 for ISIS.
 - path affinity constraints
 - path SRLG constraints";
 }
 uses te-types:generic-path-disjointness;
 uses te-types:path-constraints-route-objects;
 container path-in-segment {
 presence "The end-to-end tunnel starts in a previous domain;
 this tunnel is a segment in the current domain.";
 description
 }
}
```

```
 "If an end-to-end tunnel crosses multiple domains using
 the same technology, some additional constraints have to be
 taken in consideration in each domain.
 This TE tunnel segment is stitched to the upstream TE tunnel
 segment.";
 uses te-types:label-set-info;
}
container path-out-segment {
 presence
 "The end-to-end tunnel is not terminated in this domain;
 this tunnel is a segment in the current domain.";
 description
 "If an end-to-end tunnel crosses multiple domains using
 the same technology, some additional constraints have to be
 taken in consideration in each domain.
 This TE tunnel segment is stitched to the downstream TE
 tunnel segment.";
 uses te-types:label-set-info;
}
}

grouping named-path-constraints {
 description
 "Global named path constraints configuration
 grouping.";
 container named-path-constraints {
 description
 "TE named path constraints container.";
 list named-path-constraint {
 if-feature "te-types:named-path-constraints";
 key "name";
 leaf name {
 type string;
 description
 "A string name that uniquely identifies a
 path constraint set.";
 }
 uses path-constraints-common;
 description
 "A list of named path constraints.";
 }
 }
}

/* TE globals container data */

grouping globals-grouping {
 description
```



```
 "Globals TE system-wide configuration data grouping.";
 container globals {
 description
 "Globals TE system-wide configuration data container.";
 uses named-admin-groups;
 uses named-srlgs;
 uses named-path-constraints;
 }
}

/* TE tunnels container data */

grouping tunnels-grouping {
 description
 "Tunnels TE configuration data grouping.";
 container tunnels {
 description
 "Tunnels TE configuration data container.";
 list tunnel {
 key "name";
 description
 "The list of TE tunnels.";
 uses tunnel-properties;
 uses tunnel-actions;
 uses tunnel-protection-actions;
 }
 }
}

/* TE LSPs ephemeral state container data */

grouping lsp-properties-state {
 description
 "LSPs state operational data grouping.";
 leaf source {
 type te-types:te-node-id;
 description
 "Tunnel sender address extracted from
 SENDER_TEMPLATE object.";
 reference
 "RFC3209";
 }
 leaf destination {
 type te-types:te-node-id;
 description
 "The tunnel endpoint address extracted from SESSION object.";
 reference
 "RFC3209";
 }
}
```

```
 }
 leaf tunnel-id {
 type uint16;
 description
 "The tunnel identifier used in the SESSION that remains
 constant over the life of the tunnel.";
 reference
 "RFC3209";
 }
 leaf extended-tunnel-id {
 type yang:dotted-quad;
 description
 "The LSP Extended Tunnel ID.";
 reference
 "RFC3209";
 }
 leaf operational-state {
 type identityref {
 base te-types:lsp-state-type;
 }
 description
 "The LSP operational state.";
 }
 leaf signaling-type {
 type identityref {
 base te-types:path-signaling-type;
 }
 description
 "The signaling protocol used to set up this LSP.";
 }
 leaf origin-type {
 type enumeration {
 enum ingress {
 description
 "Origin ingress.";
 }
 enum egress {
 description
 "Origin egress.";
 }
 enum transit {
 description
 "Origin transit.";
 }
 }
 default "ingress";
 description
 "The origin of the LSP relative to the location of the local
```

```
 switch in the path.";
 }
 leaf lsp-resource-status {
 type enumeration {
 enum primary {
 description
 "A primary LSP is a fully established LSP for which the
 resource allocation has been committed at the data
 plane.";
 }
 enum secondary {
 description
 "A secondary LSP is an LSP that has been provisioned
 in the control plane only; e.g. resource allocation
 has not been committed at the data plane.";
 }
 }
 default "primary";
 description
 "LSP resource allocation state.";
 reference
 "RFC4872, section 4.2.1";
 }
 uses protection-restoration-properties-state;
}

/** End of TE global groupings */
/**
 * TE container
 */

container te {
 presence "Enable TE feature.";
 description
 "TE global container.";
 /* TE Global Data */
 uses globals-grouping;

 /* TE Tunnel Data */
 uses tunnels-grouping;

 /* TE LSPs Data */
 uses lsps-grouping;
}

/* TE Tunnel RPCs/execution Data */

rpc tunnels-path-compute {
```

```
description
 "TE tunnels RPC nodes.";
input {
 container path-compute-info {
 /*
 * An external path compute module may augment this
 * target.
 */
 description
 "RPC input information.";
 }
}
output {
 container path-compute-result {
 /*
 * An external path compute module may augment this
 * target.
 */
 description
 "RPC output information.";
 }
}

rpc tunnels-actions {
 description
 "TE tunnels actions RPC";
 input {
 container tunnel-info {
 description
 "TE tunnel information.";
 choice filter-type {
 mandatory true;
 description
 "Filter choice.";
 case all-tunnels {
 leaf all {
 type empty;
 mandatory true;
 description
 "Apply action on all TE tunnels.";
 }
 }
 case one-tunnel {
 leaf tunnel {
 type tunnel-ref;
 description
 "Apply action on the specific TE tunnel.";
 }
 }
 }
 }
 }
}
```

```

 }
 }
}
container action-info {
 description
 "TE tunnel action information.";
 leaf action {
 type identityref {
 base tunnel-actions-type;
 }
 description
 "The action type.";
 }
 leaf disruptive {
 when "derived-from-or-self(..../action, "
 + "'te:tunnel-action-reoptimize')";
 type empty;
 description
 "Specifies whether or not the reoptimization action
 is allowed to be disruptive.";
 }
}
}
output {
 leaf action-result {
 type identityref {
 base te-types:te-action-result;
 }
 description
 "The result of the tunnel action operation.";
 }
}
}
}
<CODE ENDS>

```

Figure 9: TE Tunnel data model YANG module

## 6. TE Device YANG Model

The device TE YANG module ('ietf-te-device') models data that is specific to managing a TE device. This module augments the generic TE YANG module.

## 6.1. Module Structure

### 6.1.1. TE Interfaces

This branch of the model manages TE interfaces that are present on a device. Examples of TE interface properties are:

- \* Maximum reservable bandwidth, bandwidth constraints (BC)
- \* Flooding parameters
  - Flooding intervals and threshold values
- \* interface attributes
  - (Extended) administrative groups
  - SRLG values
  - TE metric value
- \* Fast reroute backup tunnel properties (such as static, auto-tunnel)

The derived state associated with interfaces is grouped under the interface "state" sub-container as shown in Figure 10. This covers state data such as:

- \* Bandwidth information: maximum bandwidth, available bandwidth at different priorities and for each class-type (CT)
- \* List of admitted LSPs
  - Name, bandwidth value and pool, time, priority
- \* Statistics: state counters, flooding counters, admission counters (accepted/rejected), preemption counters
- \* Adjacency information
  - Neighbor address
  - Metric value

```

module: ietf-te-device
 augment /te:te:
 +--rw interfaces
 .
 +-- rw te-dev:te-attributes
 <<intended configuration>>
 .
 +-- ro state
 <<derived state associated with the TE interface>>

```

Figure 10: TE interface state YANG subtree

## 6.2. Tree Diagram

Figure 11 shows the tree diagram of the device TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te-device
 augment /te:te:
 +--rw interfaces
 |
 | +--rw threshold-type? enumeration
 | +--rw delta-percentage? rt-types:percentage
 | +--rw threshold-specification? enumeration
 | +--rw up-thresholds* rt-types:percentage
 | +--rw down-thresholds* rt-types:percentage
 | +--rw up-down-thresholds* rt-types:percentage
 | +--rw interface* [interface]
 | +--rw interface if:interface-ref
 | +--rw te-metric?
 | |
 | | te-types:te-metric
 | +--rw (admin-group-type)?
 | |
 | | +--:(value-admin-groups)
 | | |
 | | | +--rw (value-admin-group-type)?
 | | | |
 | | | | +--:(admin-groups)
 | | | | |
 | | | | | +--rw admin-group?
 | | | | | |
 | | | | | | te-types:admin-group
 | | | | | | +--:(extended-admin-groups)
 | | | | | | {te-types:extended-admin-groups}?
 | | | | | | +--rw extended-admin-group?
 | | | | | | |
 | | | | | | | te-types:extended-admin-group
 | | | | | | +--:(named-admin-groups)
 | | | | | | +--rw named-admin-groups* [named-admin-group]
 | | | | | | |
 | | | | | | | {te-types:extended-admin-groups, te-types:named-
 | | | | | | | extended-admin-groups}?
 | | | | | | | +--rw named-admin-group leafref
 | | | | | | +--rw (srlg-type)?
 | | | | | | |
 | | | | | | | +--:(value-srlgs)
 | | | | | | | +--rw values* [value]

```

```

 | +---rw value uint32
 +---:(named-srlgs)
 | +---rw named-srlgs* [named-srlg]
 | | {te-types:named-srlg-groups}?
 | +---rw named-srlg leafref
+---rw threshold-type? enumeration
+---rw delta-percentage?
| rt-types:percentage
+---rw threshold-specification? enumeration
+---rw up-thresholds*
| rt-types:percentage
+---rw down-thresholds*
| rt-types:percentage
+---rw up-down-thresholds*
| rt-types:percentage
+---rw switching-capabilities* [switching-capability]
| +---rw switching-capability identityref
| +---rw encoding? identityref
+---ro state
 +---ro te-advertisements-state
 | +---ro flood-interval? uint32
 | +---ro last-flooded-time? uint32
 | +---ro next-flooded-time? uint32
 | +---ro last-flooded-trigger? enumeration
 | +---ro advertised-level-areas* [level-area]
 | +---ro level-area uint32
+---rw performance-thresholds
augment /te:te/te:globals:
+---rw lsp-install-interval? uint32
+---rw lsp-cleanup-interval? uint32
+---rw lsp-invalidation-interval? uint32
augment /te:te/te:tunnels/te:tunnel:
+---rw path-invalidation-action? identityref
+---rw lsp-install-interval? uint32
+---rw lsp-cleanup-interval? uint32
+---rw lsp-invalidation-interval? uint32
augment /te:te/te:lsps/te:lsp:
+---ro lsp-timers
| +---ro life-time? uint32
| +---ro time-to-install? uint32
| +---ro time-to-destroy? uint32
+---ro downstream-info
| +---ro nhop? te-types:te-tp-id
| +---ro outgoing-interface? if:interface-ref
| +---ro neighbor
| | +---ro id? te-gen-node-id
| | +---ro type? enumeration
+---ro label? rt-types:generalized-label

```



```

+--ro upstream-info
 +--ro phop? te-types:te-tp-id
 +--ro neighbor
 | +--ro id? te-gen-node-id
 | +--ro type? enumeration
 +--ro label? rt-types:generalized-label

rpcs:
 +---x link-state-update
 +---w input
 +---w (filter-type)
 +---:(match-all)
 | +---w all empty
 +---:(match-one-interface)
 +---w interface? if:interface-ref

```

Figure 11: TE Tunnel device model YANG tree diagram

### 6.3. YANG Module

The device TE YANG module 'ietf-te-device' imports the following module(s):

- \* ietf-yang-types and ietf-inet-types defined in [RFC6991]
- \* ietf-interfaces defined in [RFC8343]
- \* ietf-routing-types defined in [RFC8294]
- \* ietf-te-types defined in [RFC8776]
- \* ietf-te defined in this document

```

<CODE BEGINS> file "ietf-te-device@2021-10-22.yang"
module ietf-te-device {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

 /* Replace with IANA when assigned */

 prefix te-dev;

 /* Import TE module */

 import ietf-te {
 prefix te;
 reference
 "draft-ietf-teas-yang-te: A YANG Data Model for Traffic

```

```
 Engineering Tunnels and Interfaces";
}

/* Import TE types */

import ietf-te-types {
 prefix te-types;
 reference
 "RFC8776: Common YANG Data Types for Traffic Engineering.";
}
import ietf-interfaces {
 prefix if;
 reference
 "RFC8343: A YANG Data Model for Interface Management";
}
import ietf-routing-types {
 prefix rt-types;
 reference
 "RFC8294: Common YANG Data Types for the Routing Area";
}

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";
contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 Editor: Tarek Saad
 <mailto:tsaad@juniper.net>

 Editor: Rakesh Gandhi
 <mailto:rgandhi@cisco.com>

 Editor: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editor: Himanshu Shah
 <mailto:hshah@ciena.com>

 Editor: Xufeng Liu
 <mailto:xufeng.liu.ietf@gmail.com>

 Editor: Igor Bryskin
 <mailto:i_bryskin@yahoo.com>";
description
 "YANG data module for TE device configurations,
 state, and RPCs. The model fully conforms to the
```

Network Management Datastore Architecture (NMDA).

Copyright (c) 2019 IETF Trust and the persons  
identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD License  
set forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<https://trustee.ietf.org/license-info>).  
This version of this YANG module is part of RFC XXXX; see  
the RFC itself for full legal notices.";

```
// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.
```

```
revision 2021-10-22 {
 description
 "Latest update to TE device YANG module.";
 reference
 "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
 and Interfaces";
}
```

```
/**
 * TE LSP device state grouping
 */
```

```
grouping lsps-device-info {
 description
 "TE LSP device state grouping.";
 container lsp-timers {
 when "../te:origin-type = 'ingress'" {
 description
 "Applicable to ingress LSPs only.";
 }
 description
 "Ingress LSP timers.";
 leaf life-time {
 type uint32;
 units "seconds";
 description
 "TE LSP lifetime.";
 }
 leaf time-to-install {
```

```
 type uint32;
 units "seconds";
 description
 "TE LSP installation delay time.";
 }
 leaf time-to-destroy {
 type uint32;
 units "seconds";
 description
 "TE LSP expiration delay time.";
 }
}
container downstream-info {
 when "../te:origin-type != 'egress'" {
 description
 "Downstream information of the LSP.";
 }
 description
 "downstream information.";
 leaf nhop {
 type te-types:te-tp-id;
 description
 "downstream next-hop address.";
 }
 leaf outgoing-interface {
 type if:interface-ref;
 description
 "downstream interface.";
 }
 container neighbor {
 uses te:te-generic-node-id;
 description
 "downstream neighbor address.";
 }
 leaf label {
 type rt-types:generalized-label;
 description
 "downstream label.";
 }
}
container upstream-info {
 when "../te:origin-type != 'ingress'" {
 description
 "Upstream information of the LSP.";
 }
 description
 "upstream information.";
 leaf phop {
```

```
 type te-types:te-tp-id;
 description
 "upstream next-hop or previous-hop address.";
 }
 container neighbor {
 uses te:te-generic-node-id;
 description
 "upstream neighbor address.";
 }
 leaf label {
 type rt-types:generalized-label;
 description
 "upstream label.";
 }
}

/**
 * Device general groupings.
 */

grouping lsp-device-timers {
 description
 "Device TE LSP timers configs.";
 leaf lsp-install-interval {
 type uint32;
 units "seconds";
 description
 "TE LSP installation delay time.";
 }
 leaf lsp-cleanup-interval {
 type uint32;
 units "seconds";
 description
 "TE LSP cleanup delay time.";
 }
 leaf lsp-invalidation-interval {
 type uint32;
 units "seconds";
 description
 "TE LSP path invalidation before taking action delay time.";
 }
}

/**
 * TE global device groupings
 */
/* TE interface container data */
```

```
grouping interfaces-grouping {
 description
 "TE interface configuration data grouping.";
 container interfaces {
 description
 "Configuration data model for TE interfaces.";
 uses te-all-attributes;
 list interface {
 key "interface";
 description
 "TE interfaces.";
 leaf interface {
 type if:interface-ref;
 description
 "TE interface name.";
 }
 /* TE interface parameters */
 uses te-attributes;
 }
 }
}

/**
 * TE interface device groupings
 */

grouping te-admin-groups-config {
 description
 "TE interface affinities grouping.";
 choice admin-group-type {
 description
 "TE interface administrative groups
 representation type.";
 case value-admin-groups {
 choice value-admin-group-type {
 description
 "choice of admin-groups.";
 case admin-groups {
 description
 "Administrative group/Resource
 class/Color.";
 leaf admin-group {
 type te-types:admin-group;
 description
 "TE interface administrative group.";
 }
 }
 }
 case extended-admin-groups {
```

```
 if-feature "te-types:extended-admin-groups";
 description
 "Extended administrative group/Resource
 class/Color.";
 leaf extended-admin-group {
 type te-types:extended-admin-group;
 description
 "TE interface extended administrative group.";
 }
 }
}
}
case named-admin-groups {
 list named-admin-groups {
 if-feature "te-types:extended-admin-groups";
 if-feature "te-types:named-extended-admin-groups";
 key "named-admin-group";
 description
 "A list of named admin-group entries.";
 leaf named-admin-group {
 type leafref {
 path "../..../te:globals/"
 + "te:named-admin-groups/te:named-admin-group/"
 + "te:name";
 }
 description
 "A named admin-group entry.";
 }
 }
}
}
}

/* TE interface SRLGs */

grouping te-srlgs-config {
 description
 "TE interface SRLG grouping.";
 choice srlg-type {
 description
 "Choice of SRLG configuration.";
 case value-srlgs {
 list values {
 key "value";
 description
 "List of SRLG values that
 this link is part of.";
 leaf value {
```

```
 type uint32 {
 range "0..4294967295";
 }
 description
 "Value of the SRLG";
 }
}
}
case named-srlgs {
 list named-srlgs {
 if-feature "te-types:named-srlg-groups";
 key "named-srlg";
 description
 "A list of named SRLG entries.";
 leaf named-srlg {
 type leafref {
 path "../..../te:globals/"
 + "te:named-srlgs/te:named-srlg/te:name";
 }
 description
 "A named SRLG entry.";
 }
 }
}
}
}
}

grouping te-igp-flooding-bandwidth-config {
 description
 "Configurable items for igp flooding bandwidth
 threshold configuration.";
 leaf threshold-type {
 type enumeration {
 enum delta {
 description
 "'delta' indicates that the local
 system should flood IGP updates when a
 change in reserved bandwidth >= the specified
 delta occurs on the interface.";
 }
 enum threshold-crossed {
 description
 "THRESHOLD-CROSSED indicates that
 the local system should trigger an update (and
 hence flood) the reserved bandwidth when the
 reserved bandwidth changes such that it crosses,
 or becomes equal to one of the threshold values.";
 }
 }
 }
}
```



```
}
description
 "The type of threshold that should be used to specify the
 values at which bandwidth is flooded. 'delta' indicates that
 the local system should flood IGP updates when a change in
 reserved bandwidth >= the specified delta occurs on the
 interface. Where 'threshold-crossed' is specified, the local
 system should trigger an update (and hence flood) the
 reserved bandwidth when the reserved bandwidth changes such
 that it crosses, or becomes equal to one of the threshold
 values."
}
leaf delta-percentage {
 when "../threshold-type = 'delta'" {
 description
 "The percentage delta can only be specified when the
 threshold type is specified to be a percentage delta of
 the reserved bandwidth."
 }
 type rt-types:percentage;
 description
 "The percentage of the maximum-reservable-bandwidth
 considered as the delta that results in an IGP update
 being flooded."
}
leaf threshold-specification {
 when "../threshold-type = 'threshold-crossed'" {
 description
 "The selection of whether mirrored or separate threshold
 values are to be used requires user specified thresholds
 to be set."
 }
 type enumeration {
 enum mirrored-up-down {
 description
 "mirrored-up-down indicates that a single set of
 threshold values should be used for both increasing
 and decreasing bandwidth when determining whether
 to trigger updated bandwidth values to be flooded
 in the IGP TE extensions."
 }
 enum separate-up-down {
 description
 "separate-up-down indicates that a separate
 threshold values should be used for the increasing
 and decreasing bandwidth when determining whether
 to trigger updated bandwidth values to be flooded
 in the IGP TE extensions."
 }
 }
}
```

```
 }
 }
 description
 "This value specifies whether a single set of threshold
 values should be used for both increasing and decreasing
 bandwidth when determining whether to trigger updated
 bandwidth values to be flooded in the IGP TE extensions.
 'mirrored-up-down' indicates that a single value (or set of
 values) should be used for both increasing and decreasing
 values, where 'separate-up-down' specifies that the
 increasing and decreasing values will be separately
 specified.";
}
leaf-list up-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'separate-up-down'" {
 description
 "A list of up-thresholds can only be specified when the
 bandwidth update is triggered based on crossing a
 threshold and separate up and down thresholds are
 required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth) at which bandwidth updates are to be
 triggered when the bandwidth is increasing.";
}
leaf-list down-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'separate-up-down'" {
 description
 "A list of down-thresholds can only be specified when the
 bandwidth update is triggered based on crossing a
 threshold and separate up and down thresholds are
 required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth) at which bandwidth updates are to be
 triggered when the bandwidth is decreasing.";
}
leaf-list up-down-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'mirrored-up-down'" {
 description
 "A list of thresholds corresponding to both increasing
```

```
 and decreasing bandwidths can be specified only when an
 update is triggered based on crossing a threshold, and
 the same up and down thresholds are required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth of the interface) at which bandwidth
 updates are flooded - used both when the bandwidth is
 increasing and decreasing.";
}
}

/* TE interface metric */

grouping te-metric-config {
 description
 "TE interface metric grouping.";
 leaf te-metric {
 type te-types:te-metric;
 description
 "TE interface metric.";
 }
}

/* TE interface switching capabilities */

grouping te-switching-cap-config {
 description
 "TE interface switching capabilities.";
 list switching-capabilities {
 key "switching-capability";
 description
 "List of interface capabilities for this interface.";
 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for this interface.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by this interface.";
 }
 }
}
```

```
 }
 }

 grouping te-advertisements-state {
 description
 "TE interface advertisements state grouping.";
 container te-advertisements-state {
 description
 "TE interface advertisements state container.";
 leaf flood-interval {
 type uint32;
 description
 "The periodic flooding interval.";
 }
 leaf last-flooded-time {
 type uint32;
 units "seconds";
 description
 "Time elapsed since last flooding in seconds.";
 }
 leaf next-flooded-time {
 type uint32;
 units "seconds";
 description
 "Time remained for next flooding in seconds.";
 }
 leaf last-flooded-trigger {
 type enumeration {
 enum link-up {
 description
 "Link-up flooding trigger.";
 }
 enum link-down {
 description
 "Link-down flooding trigger.";
 }
 enum threshold-up {
 description
 "Bandwidth reservation up threshold.";
 }
 enum threshold-down {
 description
 "Bandwidth reservation down threshold.";
 }
 enum bandwidth-change {
 description
 "Bandwidth capacity change.";
 }
 }
 }
 }
 }
```

```
 enum user-initiated {
 description
 "Initiated by user.";
 }
 enum srlg-change {
 description
 "SRLG property change.";
 }
 enum periodic-timer {
 description
 "Periodic timer expired.";
 }
 }
 default "periodic-timer";
 description
 "Trigger for the last flood.";
}
list advertised-level-areas {
 key "level-area";
 description
 "List of level-areas that the TE interface is advertised
 in.";
 leaf level-area {
 type uint32;
 description
 "The IGP area or level where the TE interface link state
 is advertised in.";
 }
}
}
}

/* TE interface attributes grouping */

grouping te-attributes {
 description
 "TE attributes configuration grouping.";
 uses te-metric-config;
 uses te-admin-groups-config;
 uses te-srlgs-config;
 uses te-igp-flooding-bandwidth-config;
 uses te-switching-cap-config;
 container state {
 config false;
 description
 "State parameters for interface TE metric.";
 uses te-advertisements-state;
 }
}
```

```
}

grouping te-all-attributes {
 description
 "TE attributes configuration grouping for all
 interfaces.";
 uses te-igp-flooding-bandwidth-config;
}

/** End of TE interfaces device groupings */
/**
 * TE device augmentations
 */

augment "/te:te" {
 description
 "TE global container.";
 /* TE Interface Configuration Data */
 uses interfaces-grouping;
 container performance-thresholds {
 description
 "Performance parameters configurable thresholds.";
 }
}

/* TE globals device augmentation */

augment "/te:te/te:globals" {
 description
 "Global TE device specific configuration parameters.";
 uses lsp-device-timers;
}

/* TE tunnels device configuration augmentation */

augment "/te:te/te:tunnels/te:tunnel" {
 description
 "Tunnel device dependent augmentation.";
 leaf path-invalidation-action {
 type identityref {
 base te-types:path-invalidation-action-type;
 }
 description
 "Tunnel path invalidation action.";
 }
 uses lsp-device-timers;
}
```



- \* Subscribe to notifications on a per-client basis.
- \* Specify subtree filters or XML Path Language (XPath) filters so that only contents of interest will be sent.
- \* Specify either periodic or on-demand notifications.

## 8. TE Generic and Helper YANG Modules

## 9. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-te  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-device  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

Name: ietf-te  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te  
Prefix: te  
Reference: RFCXXXX

Name: ietf-te-device  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-device  
Prefix: te-device  
Reference: RFCXXXX

## 10. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].



The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/te/globals"`: This module specifies the global TE configurations on a device. Unauthorized access to this container could cause the device to ignore packets it should receive and process.

`"/te/tunnels"`: This list specifies the configuration and state of TE Tunnels present on the device or controller. Unauthorized access to this list could cause the device to ignore packets it should receive and process. An attacker may also use state to derive information about the network topology, and subsequently orchestrate further attacks.

`"/te/interfaces"`: This list specifies the configuration and state TE interfaces on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

`"/te/lspss"`: this list contains information state about established LSPs in the network. An attacker can use this information to derive information about the network topology, and subsequently orchestrate further attacks.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

`"/te/tunnels-actions"`: using this RPC, an attacker can modify existing paths that may be carrying live traffic, and hence result to interruption to services carried over the network.

`"/te/tunnels-path-compute"`: using this RPC, an attacker can retrieve secured information about the network provider which can be used to orchestrate further attacks.

The security considerations spelled out in the YANG 1.1 specification [RFC7950] apply for this document as well.

## 11. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of this model.

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, and Raqib Jones for providing useful feedback on this document.

## 12. Contributors

Himanshu Shah  
Ciena

Email: hshah@ciena.com

Xia Chen  
Huawei Technologies

Email: jescia.chenxia@huawei.com

Bin Wen  
Comcast

Email: Bin\_Wen@cable.comcast.com

## 13. Appendix A: Data Tree Examples

This section contains examples of use of the model with RESTCONF [RFC8040] and JSON encoding.

For the example we will use a 4 node MPLS network where RSVP-TE MPLS Tunnels can be setup. The loopbacks of each router are shown. The network in Figure 13 will be used in the examples described in the following sections.

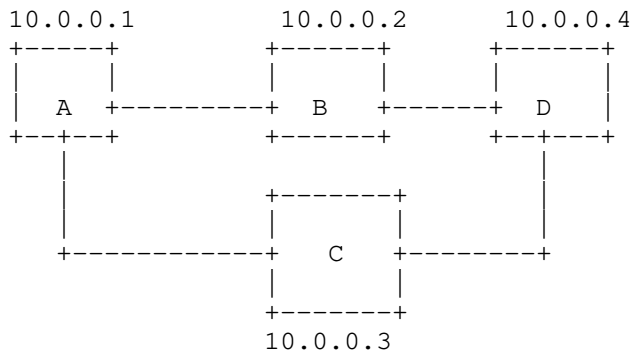


Figure 13: TE network used in data tree examples

### 13.1. Basic Tunnel Setup

This example uses the TE Tunnel YANG data model defined in this document to create an RSVP-TE signaled Tunnel of packet LSP encoding type. First, the TE Tunnel is created with no specific restrictions or constraints (e.g., protection or restoration). The TE Tunnel ingresses on router A and egresses on router D.

In this case, the TE Tunnel is created without specifying additional information about the primary paths.

```

POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
 "ietf-te:tunnel": [
 {
 "name": "Example_LSP_Tunnel_A_2",
 "encoding": "te-types:lsp-encoding-packet",
 "admin-state": "te-types:tunnel-state-up",
 "source": "10.0.0.1",
 "destination": "10.0.0.4",
 "bidirectional": "false",
 "signaling-type": "te-types:path-setup-rsvp"
 }
]
}

```

### 13.2. Global Named Path Constraints

This example uses the YANG data model to create a 'named path constraint' that can be reference by TE Tunnels. The path constraint, in this case, limits the TE Tunnel hops for the computed path.

```
POST /restconf/data/ietf-te:te/globals/named-path-constraints HTTP/1.1
```

```
Host: example.com
```

```
Accept: application/yang-data+json
```

```
Content-Type: application/yang-data+json
```

```
{
 "ietf-te:named-path-constraint": {
 "name": "max-hop-3",
 "path-metric-bounds": {
 "path-metric-bound": {
 "metric-type": "te-types:path-metric-hop",
 "upper-bound": "3"
 }
 }
 }
}
```

### 13.3. Tunnel with Global Path Constraint

In this example, the previously created 'named path constraint' is applied to the TE Tunnel created in Section 13.1.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
 "ietf-te:ietf-tunnel": [
 {
 "name": "Example_LSP_Tunnel_A_4_1",
 "encoding": "te-types:lsp-encoding-packet",
 "description": "Simple_LSP_with_named_path",
 "admin-state": "te-types:tunnel-state-up",
 "source": "10.0.0.1",
 "destination": "10.0.0.4",
 "signaling-type": "path-setup-rsvp",
 "bidirectional": "false",
 "primary-paths": [
 {
 "primary-path": {
 "name": "Simple_LSP_1",
 "use-path-computation": "true",
 "named-path-constraint": "max-hop-3"
 }
 }
]
 }
]
}
```

#### 13.4. Tunnel with Per-tunnel Path Constraint

In this example, the a per tunnel path constraint is explicitly indicated under the TE Tunnel created in Section 13.1 to constrain the computed path for the tunnel.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
 "ietf-te:tunnel": [
 {
 "name": "Example_LSP_Tunnel_A_4_2",
 "encoding": "te-types:lsp-encoding-packet",
 "admin-state": "te-types:tunnel-state-up",
 "source": "10.0.0.1",
 "destination": "10.0.0.4",
 "bidirectional": "false",
 "signaling-type": "te-types:path-setup-rsvp",
 "primary-paths": {
 "primary-path": [
 {
 "name": "path1",
 "path-metric-bounds": {
 "path-metric-bound": [
 {
 "metric-type": "te-types:path-metric-hop",
 "upper-bound": "3"
 }
]
 }
 }
]
 }
 }
]
}
```

### 13.5. Tunnel State

In this example, the 'GET' query is sent to return the state stored about the tunnel.

```
GET /restconf/data/ietf-te:te/tunnels/tunnel="Example_LSP_Tunnel_A_4_1"
/p2p-primary-paths/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

The request, with status code 200 would include, for example, the following json:

```

{
 "ietf-te:primary-paths": {
 "primary-path": [
 {
 "name": "path1",
 "path-computation-method": "te-types:path-locally-computed",
 "computed-paths-properties": {
 "computed-path-properties": [
 {
 "k-index": "1",
 "path-properties": {
 "path-route-objects": {
 "path-route-object": [
 {
 "index": "1",
 "numbered-node-hop": {
 "node-id": "10.0.0.2"
 }
 },
 {
 "index": "2",
 "numbered-node-hop": {
 "node-id": "10.0.0.4"
 }
 }
]
 }
 }
 }
]
 }
 },
 {
 "lsp": {
 "lsp": [
 {
 "tunnel-name": "Example_LSP_Tunnel_A_4_1",
 "node": "10.0.0.1 ",
 "lsp-id": "25356"
 }
]
 }
 }
]
 }
}

```

## 14. References

### 14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4872] Lang, J.P., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.



- [RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<https://www.rfc-editor.org/info/rfc6107>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP ASSOCIATION Object Extensions", RFC 6780, DOI 10.17487/RFC6780, October 2012, <<https://www.rfc-editor.org/info/rfc6780>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <<https://www.rfc-editor.org/info/rfc8795>>.

#### 14.2. Informative References

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-16, 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-16.txt>>.
- [I-D.ietf-teas-yang-rsvp]  
Beeram, V. P., Saad, T., Gandhi, R., Liu, X., and I. Bryskin, "A YANG Data Model for Resource Reservation Protocol (RSVP)", Work in Progress, Internet-Draft, draft-ietf-teas-yang-rsvp-17, 9 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-rsvp-17.txt>>.
- [RFC4427] Mannie, E., Ed. and D. Papadimitriou, Ed., "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, DOI 10.17487/RFC4427, March 2006, <<https://www.rfc-editor.org/info/rfc4427>>.
- [RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi, "Path Computation Element Communication Protocol (PCEP) Extension for Label Switched Path (LSP) Diversity Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800, July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.
- [RFC9012] Patel, K., Van de Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", RFC 9012, DOI 10.17487/RFC9012, April 2021, <<https://www.rfc-editor.org/info/rfc9012>>.

#### Authors' Addresses

Tarek Saad  
Juniper Networks  
  
Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Rakesh Gandhi  
Cisco Systems Inc  
  
Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet Draft  
Intended status: Standards Track

Xufeng Liu  
Jabil  
Igor Bryskin  
Huawei Technologies  
Vishnu Pavan Beeram  
Juniper Networks  
Tarek Saad  
Cisco Systems Inc  
Himanshu Shah  
Ciena  
Oscar Gonzalez De Dios  
Telefonica

Expires: September 13, 2017

March 13, 2017

YANG Data Model for TE Topologies  
draft-ietf-teas-yang-te-topo-08

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on September 13, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

This document defines a YANG data model for representing, retrieving and manipulating TE Topologies. The model serves as a base model that other technology specific TE Topology models can augment.

## Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

## Table of Contents

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. Introduction.....                                      | 3  |
| 1.1. Terminology.....                                     | 4  |
| 1.2. Tree Structure - Legend.....                         | 4  |
| 1.3. Prefixes in Data Node Names.....                     | 5  |
| 2. Characterizing TE Topologies.....                      | 5  |
| 3. Modeling Abstractions and Transformations.....         | 7  |
| 3.1. TE Topology.....                                     | 7  |
| 3.2. TE Node.....                                         | 7  |
| 3.3. TE Link.....                                         | 8  |
| 3.4. Transitional TE Link for Multi-Layer Topologies..... | 8  |
| 3.5. TE Link Termination Point (LTP).....                 | 10 |
| 3.6. TE Tunnel Termination Point (TTP).....               | 10 |
| 3.7. TE Node Connectivity Matrix.....                     | 11 |
| 3.8. TTP Local Link Connectivity List (LLCL).....         | 11 |
| 3.9. TE Path.....                                         | 11 |
| 3.10. TE Inter-Layer Lock.....                            | 11 |
| 3.11. Underlay TE topology.....                           | 13 |
| 3.12. Overlay TE topology.....                            | 13 |
| 3.13. Abstract TE topology.....                           | 13 |
| 4. Model Applicability.....                               | 14 |
| 4.1. Native TE Topologies.....                            | 14 |

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| 4.2. Customized TE Topologies.....                                                   | 16  |
| 4.3. Merging TE Topologies Provided by Multiple Providers.....                       | 19  |
| 4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same Provider..... | 22  |
| 5. Modeling Considerations.....                                                      | 25  |
| 5.1. Generic network topology building blocks.....                                   | 25  |
| 5.2. Technology agnostic TE Topology model.....                                      | 25  |
| 5.3. Model Structure.....                                                            | 26  |
| 5.4. Topology Identifiers.....                                                       | 27  |
| 5.5. Generic TE Link Attributes.....                                                 | 27  |
| 5.6. Generic TE Node Attributes.....                                                 | 28  |
| 5.7. TED Information Sources.....                                                    | 30  |
| 5.8. Overlay/Underlay Relationship.....                                              | 31  |
| 5.9. Templates.....                                                                  | 32  |
| 5.10. Scheduling Parameters.....                                                     | 33  |
| 5.11. Notifications.....                                                             | 33  |
| 6. Tree Structure.....                                                               | 34  |
| 7. TE Topology Yang Module.....                                                      | 66  |
| 8. Security Considerations.....                                                      | 117 |
| 9. IANA Considerations.....                                                          | 118 |
| 10. References.....                                                                  | 118 |
| 10.1. Normative References.....                                                      | 118 |
| 10.2. Informative References.....                                                    | 119 |
| 11. Acknowledgments.....                                                             | 119 |
| Contributors.....                                                                    | 119 |
| Authors' Addresses.....                                                              | 119 |

## 1. Introduction

The Traffic Engineering Database (TED) is an essential component of Traffic Engineered (TE) systems that are based on MPLS-TE [RFC2702] and GMPLS [RFC3945]. The TED is a collection of all TE information about all TE nodes and TE links in the network. The TE Topology is a schematic arrangement of TE nodes and TE links present in a given TED. There could be one or more TE Topologies present in a given Traffic Engineered system. The TE Topology is the topology on which path computational algorithms are run to compute Traffic Engineered Paths (TE Paths).

This document defines a YANG [RFC6020] data model for representing and manipulating TE Topologies. This model contains technology agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

### 1.1. Terminology

TED: The Traffic Engineering Database is a collection of all TE information about all TE nodes and TE links in a given network.

TE-Topology: The TE Topology is a schematic arrangement of TE nodes and TE links in a given TED. It forms the basis for a graph suitable for TE path computations.

Native TE Topology: Native TE Topology is a topology that is native to a given provider network. Native TE topology could be discovered via various routing protocols and/or subscribe/publish techniques. This is the topology on which path computational algorithms are run to compute TE Paths.

Customized TE Topology: Customized TE Topology is a custom topology that is produced by a provider for a given Client. This topology typically augments the Client's Native TE Topology. Path computational algorithms aren't typically run on the Customized TE Topology; they are run on the Client's augmented Native TE Topology.

### 1.2. Tree Structure - Legend

A simplified graphical representation of the data model is presented in Section 6. of this document. The following notations are used for the YANG model data tree representation.

<status> <flags> <name> <opts> <type>

<status> is one of:

- + for current
- x for deprecated
- o for obsolete

<flags> is one of:

- rw for read-write configuration data
- ro for read-only non-configuration data
- x for execution rpcs
- n for notifications

<name> is the name of the node

If the node is augmented into the tree from another module, its name is printed as <prefix>:<name>

<opts> is one of:



? for an optional leaf or node  
 ! for a presence container  
 \* for a leaf-list or list  
 Brackets [<keys>] for a list's keys  
 Curly braces {<condition>} for optional feature that make node conditional

Colon : for marking case nodes  
 Ellipses ("...") subtree contents not shown

Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

<type> is the name of the type for leafs and leaf-lists.

### 1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

| Prefix | YANG module     | Reference |
|--------|-----------------|-----------|
| yang   | ietf-yang-types | [RFC6991] |
| inet   | ietf-inet-types | [RFC6991] |

Table 1: Prefixes and corresponding YANG modules

## 2. Characterizing TE Topologies

The data model proposed by this document takes the following characteristics of TE Topologies into account:

- TE Topology is an abstract control-plane representation of the data-plane topology. Hence attributes specific to the data-plane must make their way into the corresponding TE Topology modeling. The TE Topology comprises of dynamic auto-discovered data (data that may change frequently - example: unreserved bandwidth available on data-plane links) as well as fairly static data (data that rarely changes- examples: layer network identification, switching and adaptation capabilities and limitations, fate sharing, administrative colors) associated with data-plane nodes and links. It is possible for a single TE Topology to encompass TE information at multiple switching layers.

- TE Topologies are protocol independent. Information about topological elements may be learnt via link-state protocols, but the topology can exist without being dependent on any particular protocol.
- TE Topology may not be congruent to the routing topology (topology constructed based on routing adjacencies) in a given TE System. There isn't always a one-to-one association between a TE-link and a routing adjacency. For example, the presence of a TE link between a pair of nodes doesn't necessarily imply the existence of a routing-adjacency between these nodes.
- Each TE Topological element has an information source associated with it. In some scenarios, there could be more than one information source associated with each topological element.
- TE Topologies can be hierarchical. Each node and link of a given TE Topology can be associated with respective underlay topology. This means that each node and link of a given TE Topology can be associated with an independent stack of supporting TE Topologies.
- TE Topologies can be customized. TE topologies of a given network presented by the network provider to its client could be customized on per-client request basis. This customization could be performed by provider, by client or by provider/client negotiation. The relationship between a customized topology (as presented to the client) and provider's native topology (as known in its entirety to the provider itself) could be captured as hierarchical (overlay-underlay), but otherwise the two topologies are decoupled from each other.

### 3. Modeling Abstractions and Transformations

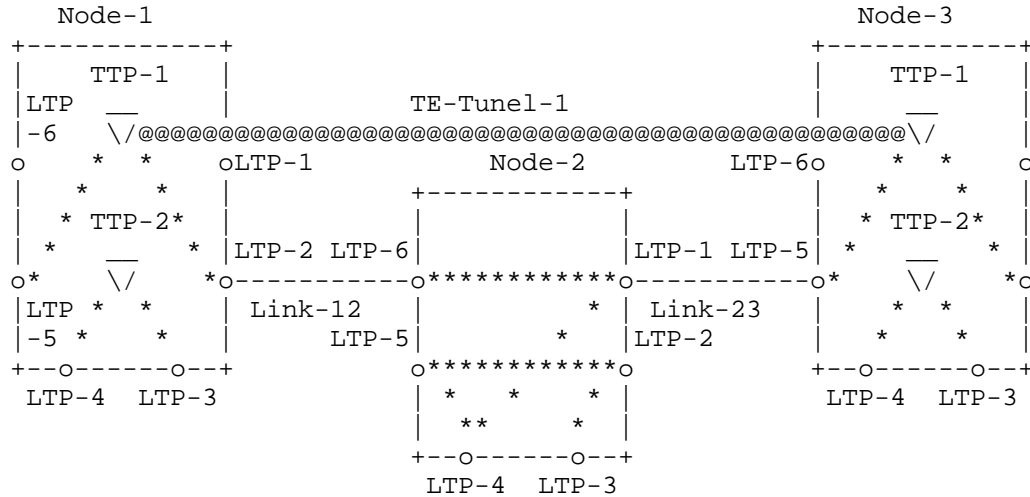


Figure 1: TE Topology Modeling Abstractions

#### 3.1. TE Topology

TE topology is a traffic engineering representation of one or more layers of network topologies. TE topology is comprised of TE nodes (TE graph vertices) interconnected via TE links (TE graph edges). A TE topology is mapped to a TE graph.

#### 3.2. TE Node

TE node is an element of a TE topology (presented as a vertex on TE graph). TE node represents one or several nodes (physical switches), or a fraction of a node. TE node belongs to and is fully defined in exactly one TE topology. TE node is assigned with the TE topology scope unique ID. TE node attributes include information related to the data plane aspects of the associated node(s) (e.g. connectivity matrix), as well as configuration data (such as TE node name). A given TE node can be reached on the TE graph over one of TE links terminated by the TE node.

Multi-layer TE nodes providing switching functions at multiple network layers are an example where a physical node can be decomposed into multiple logical TE nodes (fractions of a node). Some of these (logical) TE nodes may reside in the client layer TE topology while the remaining TE nodes belong to the server layer TE topology.

In Figure 1, Node-1, Node-2, and Node-3 are TE nodes.

### 3.3. TE Link

TE link is an element of a TE topology (presented as an edge on TE graph, arrows indicate one or both directions of the TE link). TE link represents one or several (physical) links or a fraction of a link. TE link belongs to and is fully defined in exactly one TE topology. TE link is assigned with the TE topology scope unique ID. TE link attributes include parameters related to the data plane aspects of the associated link(s) (e.g. unreserved bandwidth, resource maps/pools, etc.), as well as the configuration data (such as remote node/link IDs, SRLGs, administrative colors, etc.). TE link is connected to TE node, terminating the TE link via exactly one TE link termination point (LTP).

In Figure 1, Link-12 and Link-23 are TE links.

### 3.4. Transitional TE Link for Multi-Layer Topologies

Networks are typically composed of multiple network layers where one or multiple signals in the client layer network can be multiplexed and encapsulated into a server layer signal. The server layer signal can be carried in the server layer network across multiple nodes until the server layer signal is terminated and the client layer signals reappear in the node that terminates the server layer signal. Examples of multi-layer networks are: IP over MPLS over Ethernet, low order ODUk signals multiplexed into a high order ODUL (l>k) carried over an OCh signal (optical transport network).

TE links as defined in 3.3. can be used to represent links within a network layer. In case of a multi-layer network, TE nodes and TE links only allow representation of each network layer as a separate TE topology. Each of these single layer TE topologies would be isolated from their client and their server layer TE topology, if present (the highest and the lowest network layer in the hierarchy only have a single adjacent layer below or above, respectively). Multiplexing of client layer signals and encapsulating them into a server layer signal requires a function that is provided inside a node (typically realized in hardware). This function is also called layer transition.

One of the key requirements for path computation is to be able to calculate a path between two endpoints across a multi-layer network based on the TE topology representing this multi-layer network. This means that an additional TE construct is needed that represents

potential layer transitions in the multi-layer TE-topology that connects the TE-topologies representing each separate network layer. The so-called transitional TE link is such a construct and it represents the layer transition function residing inside a node that is decomposed into multiple logical nodes that are represented as TE nodes. Hence, a transitional TE link connects a client layer node with a server layer node. A TE link as defined in 3.3. has LTPs of exactly the same kind on each link end whereas the transitional TE link has client layer LTPs on the client side of the transitional link and in most cases a single server layer LTP on the server side. It should be noted that transitional links are a helper construct in the multi-layer TE topology and they only exist as long as they are not in use (as they represent potential connectivity). When the server layer trail has been established between the server layer LTP of two transitional links in the server layer network, the resulting client layer link in the data plane will be represented as a normal TE link in the client layer topology. The transitional TE links will re-appear when the server layer trail has been torn down.

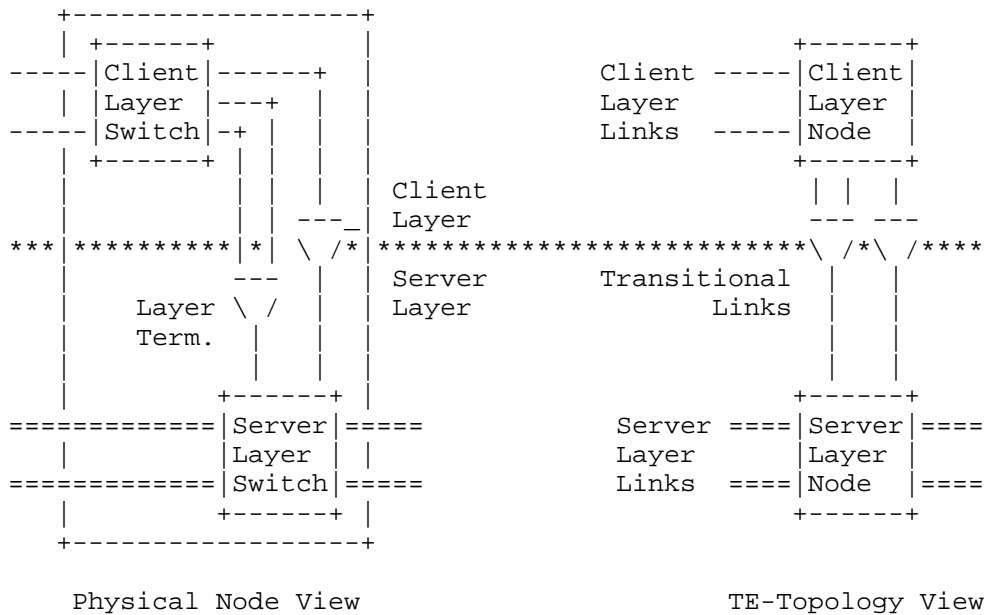


Figure 2: Modeling a Multi-Layer Node (Dual-Layer Example)

### 3.5. TE Link Termination Point (LTP)

TE link termination point (LTP) is a conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1.

In Figure 1, Node-2 has six LTPs: LTP-1 to LTP-6.

### 3.6. TE Tunnel Termination Point (TTP)

TE tunnel termination point (TTP) is an element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned with the TE node scope unique ID. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node.

In Figure 1, Node-1 has two TTPs: TTP-1 and TTP-2.

### 3.7. TE Node Connectivity Matrix

TE node connectivity matrix is a TE node's attribute describing the TE node's switching limitations in a form of valid switching combinations of the TE node's LTPs (see below). From the point of view of a potential TE path arriving at the TE node at a given inbound LTP, the node's connectivity matrix describes valid (permissible) outbound LTPs for the TE path to leave the TE node from.

In Figure 1, the connectivity matrix on Node-2 is:

{<LTP-6, LTP-1>, <LTP-5, LTP-2>, <LTP-5, LTP-4>, <LTP-4, LTP-1>, <LTP-3, LTP-2>}

### 3.8. TTP Local Link Connectivity List (LLCL)

TTP Local Link Connectivity List (LLCL) is a List of TE links terminated by the TTP hosting TE node (i.e. list of the TE link LTPs), which the TTP could be connected to. From the point of view of a potential TE path LLCL provides a list of valid TE links the TE path needs to start/stop on for the connection, taking the TE path, to be successfully terminated on the TTP in question.

In Figure 1, the LLCL on Node-1 is:

{<TTP-1, LTP-5>, <TTP-1, LTP-2>, <TTP-2, LTP-3>, <TTP-2, LTP4>}

### 3.9. TE Path

TE path is an ordered list of TE links and/or TE nodes on the TE topology graph, inter-connecting a pair of TTPs to be taken by a potential connection. TE paths, for example, could be a product of successful path computation performed for a given transport service.

In Figure 1, the TE Path for TE-Tunnel-1 is:

{Node-1:TTP-1, Link-12, Node-2, Link-23, Node-3:TTP1}

### 3.10. TE Inter-Layer Lock

TE inter-layer lock is a modeling concept describing client-server layer adaptation relationships and hence important for the multi-layer traffic engineering. It is an association of M client layer LTPs and N server layer TTPs, within which data arriving at any of the client layer LTPs could be adopted onto any of the server layer TTPs. TE inter-layer lock is identified by inter-layer lock ID, which is unique across all TE topologies provided by the same provider. The client layer LTPs and the server layer TTPs associated within a given

TE inter-layer lock are decorated with the same inter-layer lock ID attribute.

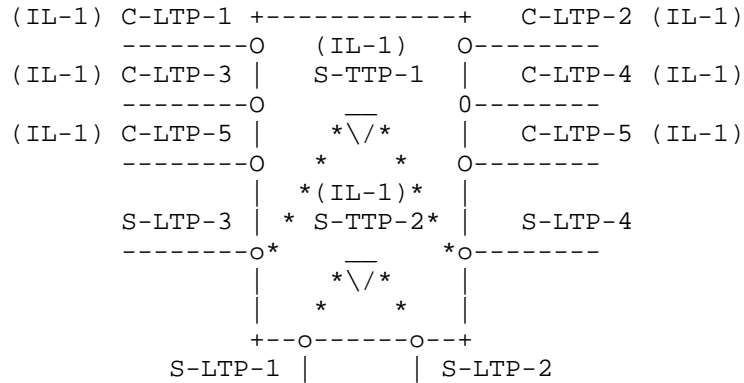


Figure 3: TE Inter-Layer Lock ID Associations

On the picture above a TE inter-layer lock with IL\_1 ID associates 6 client layer LTPs (C-LTP-1 - C-LTP-6) with two server layer TTPs (S-TTP-1 and S-TTP-2). They all have the same attribute - TE inter-layer lock ID: IL-1, which is the only thing that indicates the association. A given LTP may have 0, 1 or more inter-layer lock IDs. In the latter case this means that the data arriving at the LTP may be adopted onto any of TTPs associated with all specified inter-layer locks. For example, C-LTP-1 could have two inter-layer lock IDs - IL-1 and IL-2. This would mean that C-LTP-1 for adaptation purposes could use not just TTPs associated with inter-layer lock IL-1 (i.e. S-TTP-1 and S-TTP-2 on the picture), but any of TTPs associated with inter-layer lock IL-2 as well. Likewise, a given TTP may have one or more inter-layer lock IDs, meaning that it can offer the adaptation service to any of client layer LTPs with inter-layer lock ID matching one of its own. Additionally, each TTP has an attribute - Unreserved Adaptation Bandwidth, which announces its remaining adaptation resources sharable between all potential client LTPs.

LTPs and TTPs associated within the same TE inter-layer lock may be hosted by the same (hybrid, multi-layer) TE node or multiple TE nodes located in the same or separate TE topologies. The latter is especially important since TE topologies of different layer networks could be modeled by separate augmentations of the basic (common to all layers) TE topology model.



### 3.11. Underlay TE topology

Underlay TE topology is a TE topology that serves as a base for constructing of overlay TE topologies

### 3.12. Overlay TE topology

Overlay TE topology is a TE topology constructed based on one or more underlay TE topologies. Each TE node of the overlay TE topology represents an arbitrary segment of an underlay TE topology; each TE link of the overlay TE topology represents an arbitrary TE path in one of the underlay TE topologies. The overlay TE topology and the supporting underlay TE topologies may represent distinct layer networks (e.g. OTN/ODUk and WDM/OCh respectively) or the same layer network.

### 3.13. Abstract TE topology

Abstract TE topology is a topology that contains abstract topological elements (nodes, links, tunnel termination points). Abstract TE topology is an overlay TE topology created by a topology provider and customized for a topology provider's client based on one or more of the provider's native TE topologies (underlay TE topologies), the provider's policies and the client's preferences. For example, a first level topology provider (such as Domain Controller) can create an abstract TE topology for its client (e.g. Multi-Domain Service Coordinator) based on the provider's one or more native TE topologies, local policies/profiles and the client's TE topology configuration requests

Figure 4 shows an example of abstract TE topology.

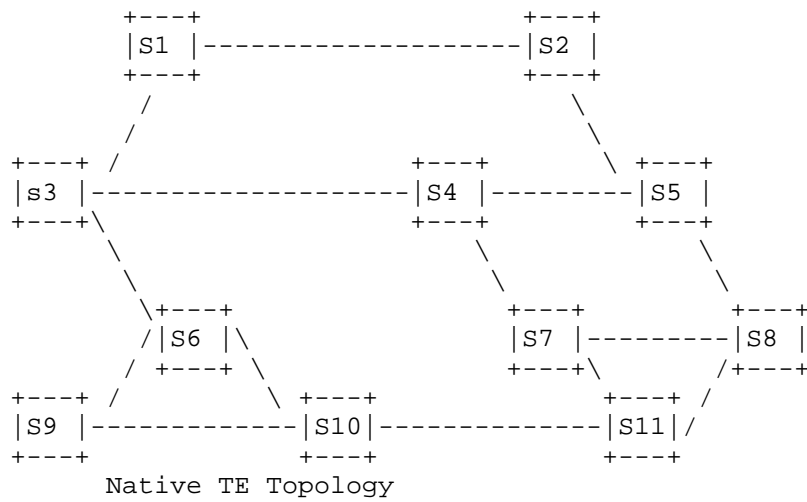
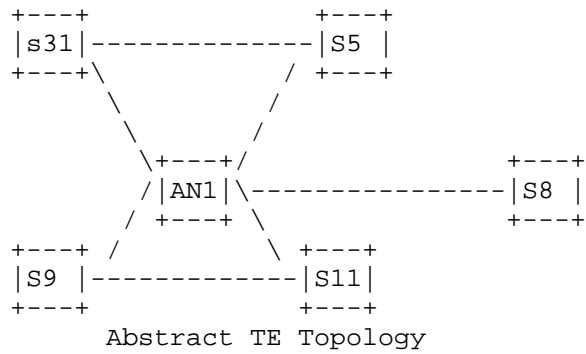


Figure 4: Abstract TE Topology

#### 4. Model Applicability

##### 4.1. Native TE Topologies

The model discussed in this draft can be used to represent and retrieve native TE topologies on a given TE system.

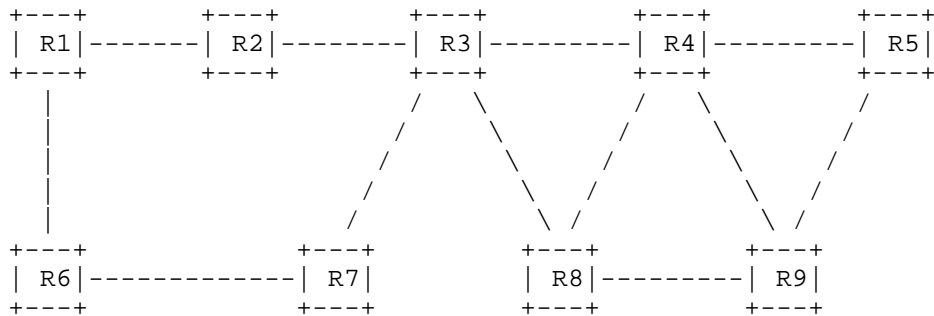


Figure 5a: Example Network Topology

Consider the network topology depicted in Figure 5a (R1 .. R9 are nodes representing routers). An implementation MAY choose to construct a native TE Topology using all nodes and links present in the given TED as depicted in Figure 5b. The data model proposed in this document can be used to retrieve/represent this TE topology.

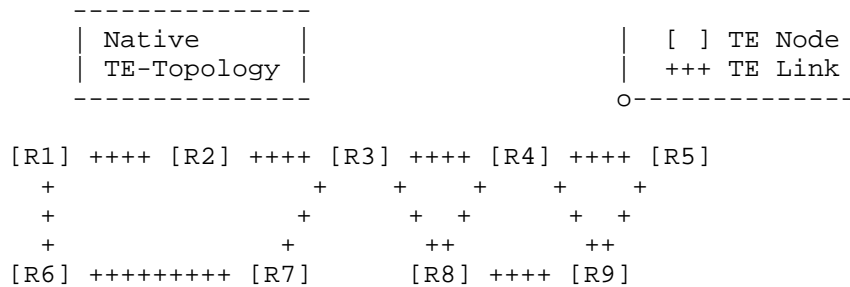


Figure 5b: Native TE Topology as seen on Node R3

Consider the case of the topology being split in a way that some nodes participate in OSPF-TE while others participate in ISIS-TE (Figure 6a). An implementation MAY choose to construct separate TE Topologies based on the information source. The native TE Topologies constructed using only nodes and links that were learnt via a specific information source are depicted in Figure 6b. The data model proposed in this document can be used to retrieve/represent these TE topologies.

Similarly, the data model can be used to represent/retrieve a TE Topology that is constructed using only nodes and links that belong to a particular technology layer. The data model is flexible enough to retrieve and represent many such native TE Topologies.

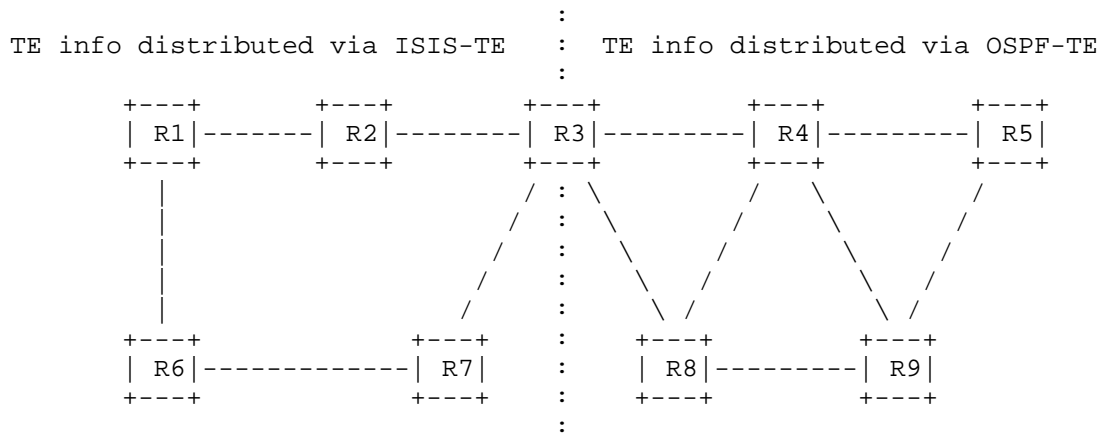


Figure 6a: Example Network Topology

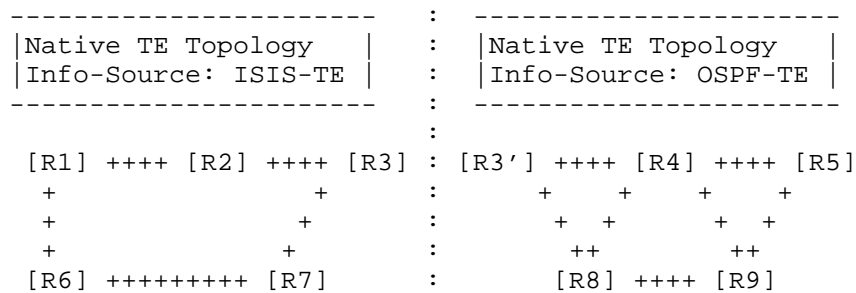


Figure 6b: Native TE Topologies as seen on Node R3

#### 4.2. Customized TE Topologies

Customized TE topology is a topology that was modified by the provider to honor a particular client's requirements or preferences. The model discussed in this draft can be used to represent, retrieve and manipulate customized TE Topologies. The model allows the provider to present the network in abstract TE Terms on a per client basis. These customized topologies contain sufficient information for the path computing client to select paths according to its policies.

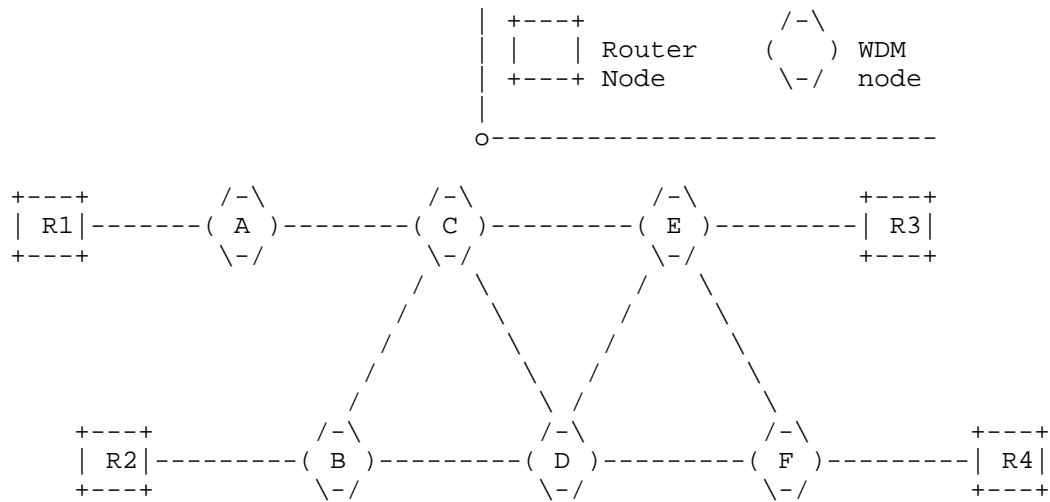


Figure 7: Example packet optical topology

Consider the network topology depicted in Figure 7. This is a typical packet optical transport deployment scenario where the WDM layer network domain serves as a Server Network Domain providing transport connectivity to the packet layer network Domain (Client Network Domain). Nodes R1, R2, R3 and R4 are IP routers that are connected to an Optical WDM transport network. A, B, C, D, E and F are WDM nodes that constitute the Server Network Domain.

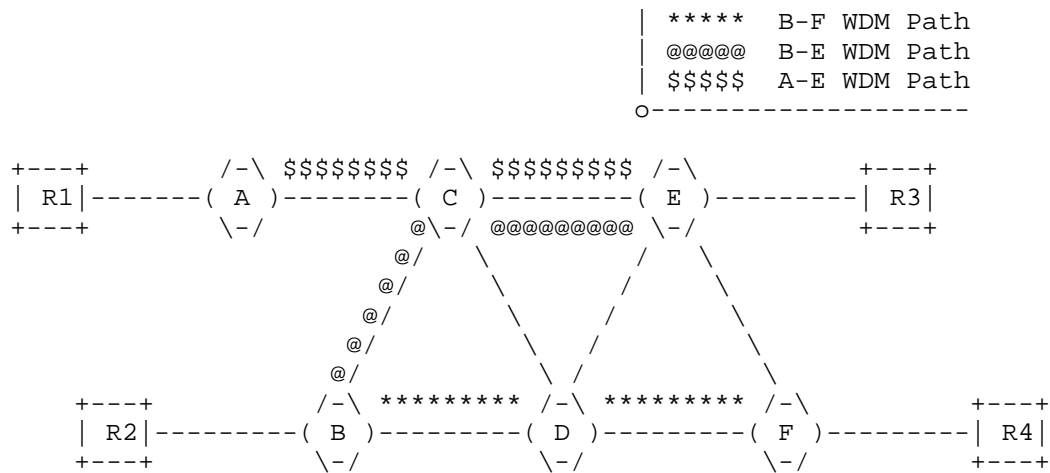


Figure 8a: Paths within the provider domain

```

+++++++ [A] ++++++ [E] ++++++
 ++++++
 ++++++
 ++++++
 ++++++
 ++++++
+++++++ [B] ++++++ [F] ++++++

```

Figure 8b: Customized TE Topology provided to the Client

The goal here is to augment the Client TE Topology with a customized TE Topology provided by the WDM network. Given the availability of the paths A-E, B-F and B-E (Figure 8a), a customized TE Topology as depicted in Figure 8b is provided to the Client. This customized TE Topology is merged with the Client's Native TE Topology and the resulting topology is depicted in Figure 8c.

```

[R1] ++++++ [A] ++++++ [E] ++++++ [R3]
 ++++++
 ++++++
 ++++++
 ++++++
 ++++++
[R2] ++++++ [B] ++++++ [F] ++++++ [R4]

```

Figure 8c: Customized TE Topology merged with the Client's Native TE Topology

The data model proposed in this document can be used to retrieve/represent/manipulate the customized TE Topology depicted in Figure 8b.

A customized TE topology is not necessarily an abstract TE topology. The provider may produce, for example, an abstract TE topology of certain type (e.g. single-abstract-node-with-connectivity-matrix topology, a border\_nodes\_connected\_via\_mesh\_of\_abstract\_links topology, etc.) and expose it to all/some clients in expectation that the clients will use it without customization. On the other hand, a client may request a customized version of the provider's native TE topology (e.g. by requesting removal of TE links which belong to certain layers, are too slow, not protected and/or

have a certain affinity). Note that the resulting TE topology will not be abstract (because it will not contain abstract elements), but customized (modified upon client's instructions).

The client ID field in the TE topology identifier (Section 5.4. indicates which client the TE topology is customized for. Although a authorized client MAY receive a TE topology with the client ID field matching some other client, the client can customize only TE topologies with the client ID field either 0 or matching the ID of the client in question. If the client starts reconfiguration of a topology its client ID will be automatically set in the topology ID field for all future configurations and updates wrt. the topology in question.

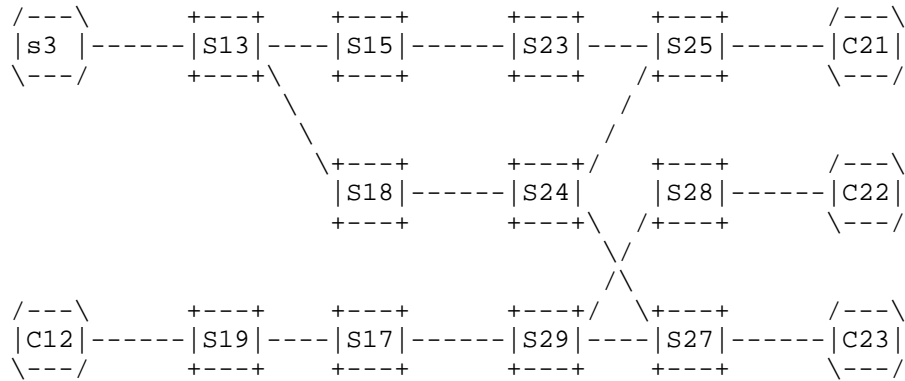
The provider MAY tell the client that a given TE topology cannot be re-negotiated, by setting its own (provider's) ID in the client ID field of the topology ID.

#### 4.3. Merging TE Topologies Provided by Multiple Providers

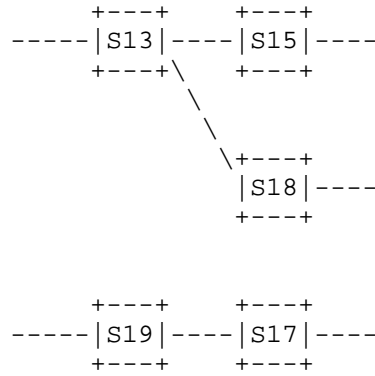
A client may receive TE topologies provided by multiple providers, each of which managing a separate domain of multi-domain network. In order to make use of said topologies, the client is expected to merge the provided TE topologies into one or more client's native TE topologies, each of which homogeneously representing the multi-domain network. This makes it possible for the client to select end-to-end TE paths for its services traversing multiple domains.

In particular, the process of merging TE topologies includes:

- Identifying neighboring domains and locking their topologies horizontally by connecting their inter-domain open-ended TE links;
- Renaming TE node, link, and SRLG IDs to ones allocated from a separate name space; this is necessary because all TE topologies are considered to be, generally speaking, independent with a possibility of clashes among TE node, link or SRLG IDs;
- Locking, vertically, TE topologies associated with different layer networks, according to provided topology inter-layer locks; this is to facilitate inter-layer path computations across multiple TE topologies provided by the same topology provider.



Domain 1 TE Topology



Domain 2 TE Topology

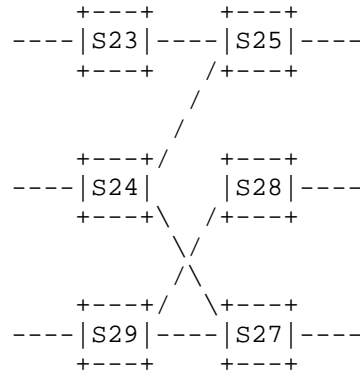


Figure 9: Merging Domain TE Topologies

Figure 9 illustrates the process of merging, by the client, of TE topologies provided by the client's providers. In the Figure, each of the two providers caters to the client (abstract or native) TE topology, describing the network domain under the respective provider's control. The client, by consulting the attributes of the inter-domain TE links - such as inter-domain plug IDs or remote TE node/link IDs (as defined by the TE Topology model) - is able to determine that:

- a) the two domains are adjacent and are inter-connected via three inter-domain TE links, and;



- b) each domain is connected to a separate customer site, connecting the left domain in the Figure to customer devices C-11 and C-12, and the right domain to customer devices C-21, C-22 and C-23.

Therefore, the client inter-connects the open-ended TE links, as shown on the upper part of the Figure.

As mentioned, one way to inter-connect the open-ended inter-domain TE links of neighboring domains is to mandate the providers to specify remote nodeID/linkID attribute in the provided inter-domain TE links. This, however, may prove to be not flexible. For example, the providers may not know the respective remote nodeIDs/ linkIDs. More importantly, this option does not allow for the client to mix-n-match multiple (more than one) topologies catered by the same providers (see below). Another, more flexible, option to resolve the open-ended inter-domain TE links is by decorating them with the inter-domain plug ID attribute. Inter-domain plug ID is a network-wide unique number that identifies on the network a connectivity supporting a given inter-domain TE link. Instead of specifying remote node ID/link ID, an inter-domain TE link may provide a non-zero inter-domain plug ID. It is expected that two neighboring domain TE topologies (provided by separate providers) will have each at least one open-ended inter-domain TE link with an inter-domain plug ID matching to one provided by its neighbor. For example, the inter-domain TE link originating from node S5 of the Domain 1 TE topology (Figure 1) and the inter-domain TE link coming from node S3 of Domain2 TE topology may specify matching inter-domain plug ID (e.g. 175344). This allows for the client to identify adjacent nodes in the separate neighboring TE topologies and resolve the inter-domain TE links connecting them regardless of their respective nodeIDs/linkIDs (which, as mentioned, could be allocated from independent name spaces). Inter-domain plug IDs may be assigned and managed by a central network authority. Alternatively, inter-domain plug IDs could be dynamically auto-discovered (e.g. via LMP protocol).

Furthermore, the client renames the TE nodes, links and SRLGs offered in the abstract TE topologies by assigning to them IDs allocated from a separate name space managed by the client. Such renaming is necessary, because the two abstract TE topologies may have their own name spaces, generally speaking, independent one from another; hence, ID overlaps/clashes are possible. For example, both TE topologies have TE nodes named S7, which, after renaming, appear in the merged TE topology as S17 and S27, respectively.

Once the merging process is complete, the client can use the merged TE topology for path computations across both domains, for example, to compute a TE path connecting C-11 to C-23.

#### 4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same Provider

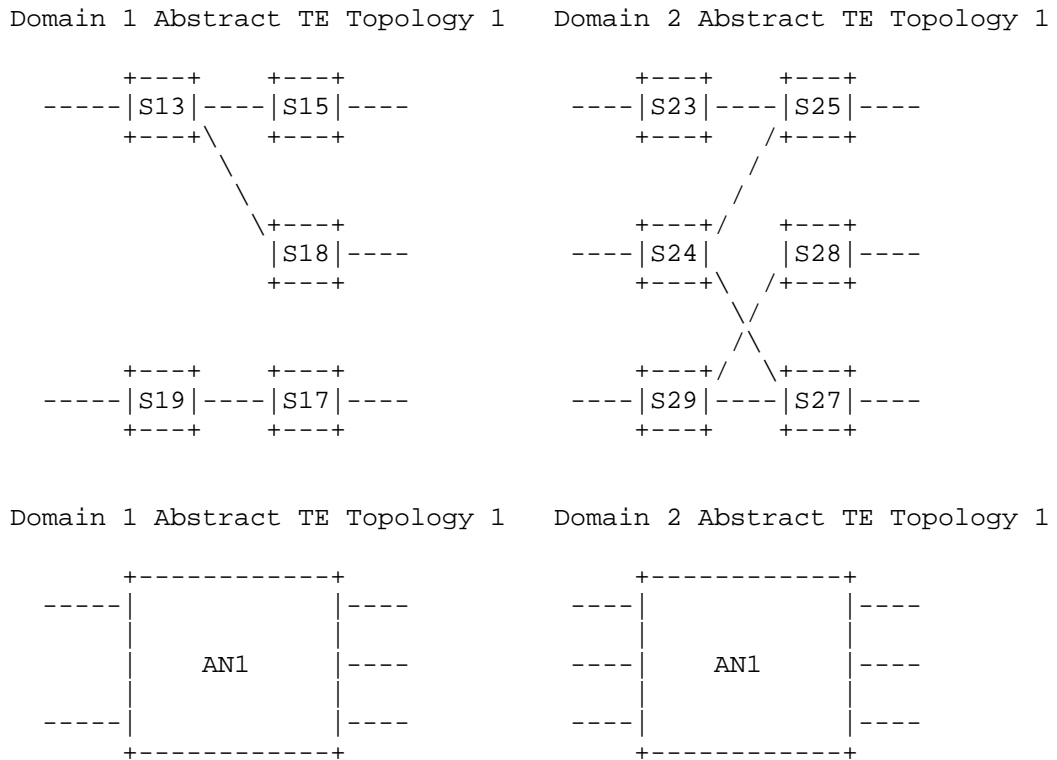


Figure 10: Merging Domain TE Topologies

Based on local configuration, templates and/or policies pushed by the client, a given provider may expose more than one abstract TE topology to the client. For example, one abstract TE topology could be optimized based on a lowest-cost criterion, while another one could be based on best possible delay metrics, while yet another one could be based on maximum bandwidth availability for the client services. Furthermore, the client may request all or some providers to expose additional abstract TE topologies, possibly of a different type and/or optimized differently, as compared to already-provided TE

topologies. In any case, the client should be prepared for a provider to offer to the client more than one abstract TE topology.

It should be up to the client (based on the client's local configuration and/or policies conveyed to the client by the client's clients) to decide how to mix-and-match multiple abstract TE topologies provided by each or some of the providers, as well as how to merge them into the client's native TE topologies. The client also decides how many such merged TE topologies it needs to produce and maintain. For example, in addition to the merged TE topology depicted in the upper part of Figure 1, the client may merge the abstract TE topologies received from the two providers, as shown in Figure 10, into the client's additional native TE topologies, as shown in Figure 11.

Note that allowing for the client mix-n-matching of multiple TE topologies assumes that inter-domain plug IDs (rather than remote nodeID/linkID) option is used for identifying neighboring domains and inter-domain TE link resolution.

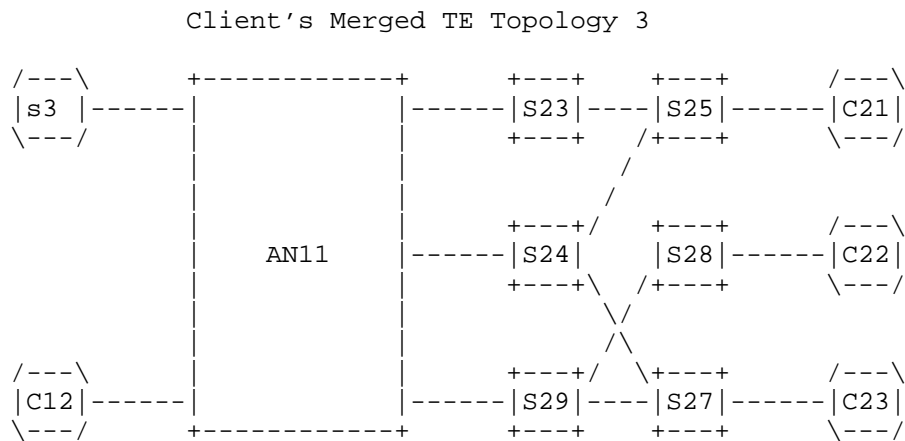
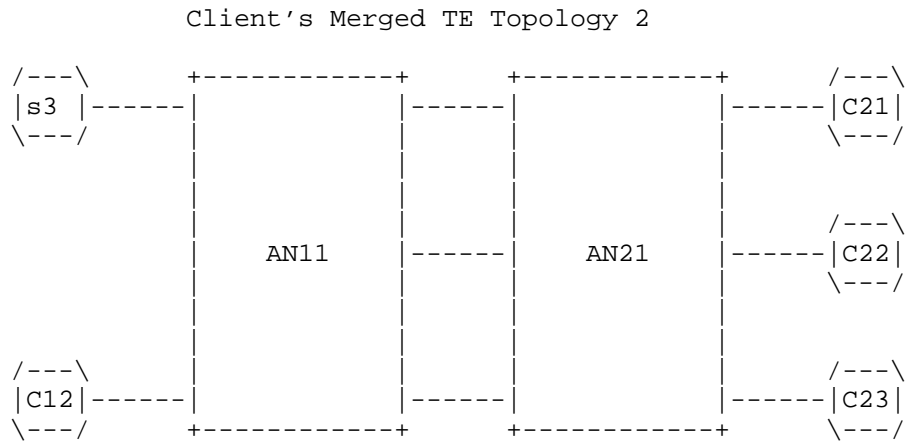


Figure 11: Multiple Native (Merged) Client's TE Topologies

It is important to note that each of the three native (merged) TE topologies could be used by the client for computing TE paths for any of the multi-domain services. The choice as to which topology to use for a given service depends on the service parameters/requirements and the topology's style, optimization criteria and the level of details.

## 5. Modeling Considerations

### 5.1. Generic network topology building blocks

The generic network topology building blocks are discussed in [YANG-NET-TOPO]. The TE Topology model proposed in this document augments and uses the `ietf-network-topology` module defined in [YANG-NET-TOPO].

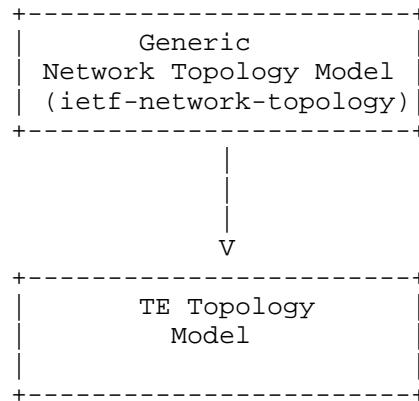


Figure 12: Augmenting the Generic Network Topology Model

### 5.2. Technology agnostic TE Topology model

The TE Topology model proposed in this document is meant to be technology agnostic. Other technology specific TE Topology models can augment and use the building blocks provided by the proposed model.

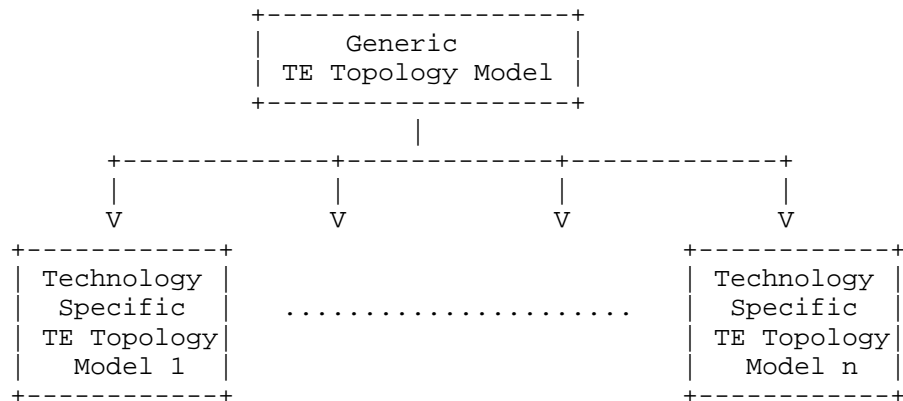


Figure 13: Augmenting the Technology agnostic TE Topology model

### 5.3. Model Structure

The high-level model structure proposed by this document is as shown below:

```

module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
 +--rw te-topology!

augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?
 |
 +--rw link-template* [name] {template}?

augment /nw:networks/nw:network:
 +--rw provider-id? te-types:te-global-id
 +--rw client-id? te-types:te-global-id
 +--rw te-topology-id? te-types:te-topology-id
 +--rw te!
 +--rw config
 |
 +--ro state

augment /nw:networks/nw:network/nw:node:
 +--rw te-node-id? te-types:te-node-id

```

```

 +--rw te!
 +--rw config
 |
 +--ro state
 |
 +--rw tunnel-termination-point* [tunnel-tp-id]
 +--rw tunnel-tp-id binary
 +--rw config
 |
 +--ro state

augment /nw:networks/nw:network/nt:link:
 +--rw te!
 +--rw config
 |
 +--ro state

augment /nw:networks/nw:network/nw:node/nt:termination-point:
 +--rw te-tp-id? te-types:te-tp-id
 +--rw te!
 +--rw config
 |
 +--ro state


```

#### 5.4. Topology Identifiers

The TE-Topology is uniquely identified by a key that has 3 constituents - te-topology-id, provider-id and client-id. The combination of provider-id and te-topology-id uniquely identifies a native TE Topology on a given provider. The client-id is used only when Customized TE Topologies come into play; a value of "0" is used as the client-id for native TE Topologies.

```

augment /nw:networks/nw:network:
 +--rw te!
 +--rw provider-id te-global-id
 +--rw client-id te-global-id
 +--rw te-topology-id te-topology-id

```

#### 5.5. Generic TE Link Attributes

The model covers the definitions for generic TE Link attributes - bandwidth, admin groups, SRLGs, switching capabilities, TE metric extensions etc.

```

+--rw te-link-attributes

 +--rw admin-status? te-admin-status
 |
 +--rw link-index? uint64
 +--rw administrative-group? te-types:admin-groups
 +--rw link-protection-type? enumeration
 +--rw max-link-bandwidth? te-bandwidth
 +--rw max-resv-link-bandwidth? te-bandwidth
 +--rw unreserved-bandwidth* [priority]
 |
 +--rw te-default-metric? uint32
 |
 +--rw te-srlgs
 +--rw te-nsrlgs {nsrlg}?

```

## 5.6. Generic TE Node Attributes

The model covers the definitions for generic TE Node attributes.

The definition of a generic connectivity matrix is shown below:

```

+--rw te-node-attributes

 +--rw connectivity-matrices

 | +--rw connectivity-matrix* [id]
 | | +--rw id uint32
 | | +--rw from
 | | | +--rw tp-ref? leafref
 | | +--rw to
 | | | +--rw tp-ref? leafref
 | | +--rw is-allowed? boolean
 | | +--rw label-restriction* [inclusive-exclusive label-start]

 | | +--rw underlay! {te-topology-hierarchy}?

 | | +--rw max-link-bandwidth? te-bandwidth
 | | +--rw max-resv-link-bandwidth? te-bandwidth
 | | +--rw unreserved-bandwidth* [priority]

 | | +--rw te-default-metric? uint32
 | | +--rw te-delay-metric? uint32
 | | +--rw te-srlgs

```



```

| | +--rw te-nsrlgs {nsrlg}? The definition of
a TTP Local Link Connectivity List is shown below:

```

```

+--rw tunnel-termination-point* [tunnel-tp-id]
+--rw tunnel-tp-id binary
+--rw config
| +--rw switching-capability? identityref
| +--rw encoding? identityref
| +--rw inter-layer-lock-id? uint32
| +--rw protection-type? identityref
| +--rw local-link-connectivities
|
| +--rw local-link-connectivity* [link-tp-ref]
| | +--rw link-tp-ref leafref
| | +--rw label-restriction* [inclusive-exclusive label-
start]
|
| +--rw max-lsp-bandwidth* [priority]
|
| +--rw max-link-bandwidth? te-bandwidth
| +--rw max-resv-link-bandwidth? te-bandwidth
| +--rw unreserved-bandwidth* [priority]
|
| +--rw te-default-metric? uint32
| +--rw te-delay-metric? uint32
| +--rw te-srlgs
| +--rw te-nsrlgs {nsrlg}?
|
+--ro state
| +--ro switching-capability? identityref
| +--ro encoding? identityref
| +--ro inter-layer-lock-id? uint32
| +--ro protection-type? identityref
| +--ro local-link-connectivities
|
| +--ro local-link-connectivity* [link-tp-ref]
| | +--ro link-tp-ref leafref
| | +--ro label-restriction* [inclusive-exclusive label-
start]
|
| +--ro max-lsp-bandwidth* [priority]
|
| +--ro max-link-bandwidth? te-bandwidth
| +--ro max-resv-link-bandwidth? te-bandwidth
| +--ro unreserved-bandwidth* [priority]
|

```

```

| | +--ro te-default-metric? uint32
| | +--ro te-delay-metric? uint32
| | +--ro te-srlgs
| | +--ro te-nsrlgs {nsrlg}?
| |
+--rw supporting-tunnel-termination-point* [node-ref tunnel-tp-
ref]
 +--rw node-ref union
 +--rw tunnel-tp-ref union

```

The attributes directly under container connectivity-matrices are the default attributes for all connectivity-matrix entries when the per entry corresponding attribute is not specified. When a per entry attribute is specified, it overrides the cooresponding attribute directly under the container connectivity-matrices. The same rule applies to the attributes directly under container local-link-connectivities.

Each TTP (Tunnel Termination Point) MAY be supported by one or more supporting TTPs. If the TE node hosting the TTP in question refers to a supporting TE node, then the supporting TTPs are hosted by the supporting TE node. If the TE node refers to an underlay TE topology, the supporting TTPs are hosted by one or more specified TE nodes of the underlay TE topology.

#### 5.7. TED Information Sources

The model allows each TE topological element to have multiple TE information sources (OSPF-TE, ISIS-TE, BGP-LS, User-Configured, System-Processed, Other). Each information source is associated with a credibility preference to indicate precedence. In scenarios where a customized TE Topology is merged into a Client's native TE Topology, the merged topological elements would point to the corresponding customized TE Topology as its information source.

```

augment /nw:networks/nw:network/nw:node:
 +--rw te!

 +--ro state

 | +--ro information-source? te-info-source
 | +--ro information-source-state
 | | +--ro credibility-preference? uint16
 | | +--ro logical-network-element? string
 | | +--ro network-instance? string
 | | +--ro topology

```

```

| | +--ro network-ref? leafref
| | +--ro node-ref? leafref
| | +--ro information-source-entry* [information-source]
| |
augment /nw:networks/nw:network/nt:link:
 +--rw te!

 +--ro state

 | +--ro information-source? te-info-source
 | +--ro information-source-state
 | | +--ro credibility-preference? uint16
 | | +--ro logical-network-element? string
 | | +--ro network-instance? string
 | | +--ro topology
 | | +--ro network-ref? leafref
 | | +--ro link-ref? leafref
 | | +--ro information-source-entry* [information-source]
 | |

```

## 5.8. Overlay/Underlay Relationship

The model captures overlay and underlay relationship for TE nodes/links. For example - in networks where multiple TE Topologies are built hierarchically, this model allows the user to start from a specific topological element in the top most topology and traverse all the way down to the supporting topological elements in the bottom most topology.

This relationship is captured via the "underlay-topology" field for the node and via the "underlay" field for the link. The use of these fields is optional and this functionality is tagged as a "feature" ("te-topology-hierarchy").

```

augment /nw:networks/nw:network/nw:node:
 +--rw te!
 +--rw te-node-id te-node-id
 +--rw config
 | +--rw te-node-template* leafref {template}?
 | +--rw te-node-attributes
 |
 | +--rw underlay-topology {te-topology-hierarchy}?
 | +--rw network-ref? leafref

```

```

augment /nw:networks/nw:network/nt:link:
 +--rw te!
 +--rw config

 +--rw te-link-attributes

 +--rw underlay! {te-topology-hierarchy}?
 +--rw primary-path
 +--rw network-ref? leafref
 +--rw path-element* [path-element-id]

 +--rw backup-path* [index]
 +--rw index uint32
 +--rw network-ref? leafref
 +--rw path-element* [path-element-id]

 +--rw underlay-protection-type? uint16
 +--rw underlay-tunnel-src

 +--rw underlay-tunnel-des


```

## 5.9. Templates

The data model provides the users with the ability to define templates and apply them to link and node configurations. The use of "template" configuration is optional and this functionality is tagged as a "feature" ("template").

```

+--rw topology* [provider-id client-id te-topology-id]

 +--rw node* [te-node-id]
 +--rw te-node-template? leafref {template}?

 +--rw link* [source-te-node-id source-te-link-id dest-te-node-id
dest-te-link-id]
 +--rw te-link-template? leafref {template}?

augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?

```

```

name | +--rw name te-types:te-template-
 | +--rw priority? uint16
 | +--rw reference-change-policy? enumeration
 | +--rw te-node-attributes
 |
 +--rw link-template* [name] {template}?
 +--rw name te-types:te-template-
name +--rw priority? uint16
 +--rw reference-change-policy? enumeration
 +--rw te-link-attributes


```

Multiple templates can be specified to a configuration element. When two or more templates specify values for the same configuration field, the value from the template with the highest priority is used. The reference-change-policy specifies the action that needs to be taken when the template changes on a configuration element that has a reference to this template. The choices of action include taking no action, rejecting the change to the template and applying the change to the corresponding configuration. [Editor's Note: The notion of "templates" has wider applicability. It is possible for this to be discussed in a separate document.]

#### 5.10. Scheduling Parameters

The model allows time scheduling parameters to be specified for each topological element or for the topology as a whole. These parameters allow the provider to present different topological views to the client at different time slots. The use of "scheduling parameters" is optional.

The YANG data model for configuration scheduling is defined in [YANG-SCHEDULE], which allows specifying configuration schedules without altering this data model.

#### 5.11. Notifications

Notifications are a key component of any topology data model.

[YANG-PUSH] and [RFC5277bis] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

- Subscribe notifications on a per client basis

- Specify subtree filters or xpath filters so that only interested contents will be sent.
- Specify either periodic or on-demand notifications.

## 6. Tree Structure

```

module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
 +--rw te-topology!
augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?
 | +--rw name te-types:te-template-
name
 | +--rw priority? uint16
 | +--rw reference-change-policy? enumeration
 | +--rw te-node-attributes
 | +--rw admin-status? te-types:te-admin-status
 | +--rw domain-id? uint32
 | +--rw is-abstract? empty
 | +--rw name? inet:domain-name
 | +--rw signaling-address* inet:ip-address
 | +--rw underlay-topology {te-topology-hierarchy}?
 | +--rw network-ref? leafref
 +--rw link-template* [name] {template}?
 | +--rw name te-types:te-template-
name
 | +--rw priority? uint16
 | +--rw reference-change-policy? enumeration
 | +--rw te-link-attributes
 | +--rw access-type? te-types:te-
link-access-type
 | +--rw external-domain
 | +--rw network-ref? leafref
 | +--rw remote-te-node-id? te-types:te-node-id
 | +--rw remote-te-link-tp-id? te-types:te-tp-id
 | +--rw plug-id? uint32
 | +--rw is-abstract? empty
 | +--rw name? string
 | +--rw underlay! {te-topology-hierarchy}?
 | +--rw primary-path
 | | +--rw network-ref? leafref
 | | +--rw path-element* [path-element-id]
 | | +--rw path-element-id uint32
 | | +--rw index? uint32

```

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |         |
|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
|  | <pre>       +--rw (type)?       +--:(ip-address)           +--rw ip-address-hop               +--rw address?      inet:ip-address               +--rw hop-type?    te-hop-type       +--:(as-number)           +--rw as-number-hop               +--rw as-number?    binary               +--rw hop-type?    te-hop-type       +--:(unnumbered-link)           +--rw unnumbered-hop               +--rw router-id?    inet:ip- </pre>                                                                                                            | address |
|  | <pre>           +--rw interface-id?    uint32           +--rw hop-type?        te-hop-type       +--:(label)           +--rw label-hop               +--rw value?        rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                            | label   |
|  | <pre>       +--:(sid)           +--rw sid-hop               +--rw sid?          rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                     | label   |
|  | <pre> +--rw backup-path* [index] +--rw index          uint32 +--rw network-ref?   leafref +--rw path-element* [path-element-id] +--rw path-element-id uint32 +--rw index?         uint32 +--rw (type)? +--:(ip-address)     +--rw ip-address-hop         +--rw address?      inet:ip-address         +--rw hop-type?    te-hop-type +--:(as-number)     +--rw as-number-hop         +--rw as-number?    binary         +--rw hop-type?    te-hop-type +--:(unnumbered-link)     +--rw unnumbered-hop         +--rw router-id?    inet:ip- </pre> | address |
|  | <pre>           +--rw interface-id?    uint32           +--rw hop-type?        te-hop-type       +--:(label)           +--rw label-hop </pre>                                                                                                                                                                                                                                                                                                                                                                                                    |         |

```

label | | | +-rw value? rt-types:generalized-
 | | +-:(sid)
 | | +-rw sid-hop
label | | +-rw sid? rt-types:generalized-
 | |
 | +-rw protection-type? uint16
 | +-rw tunnels
 | | +-rw sharing? boolean
 | | +-rw tunnel* [tunnel-name]
 | | +-rw tunnel-name string
 | | +-rw sharing? boolean
admin-status +-rw admin-status? te-types:te-
 +-rw link-index? uint64
 +-rw administrative-group? te-
types:admin-groups
 +-rw interface-switching-capability* [switching-
capability encoding]
 | +-rw switching-capability identityref
 | +-rw encoding identityref
 | +-rw max-lsp-bandwidth* [priority]
 | | +-rw priority uint8
 | | +-rw bandwidth? te-bandwidth
 +-rw link-protection-type? enumeration
 +-rw max-link-bandwidth? te-bandwidth
 +-rw max-resv-link-bandwidth? te-bandwidth
 +-rw unreserved-bandwidth* [priority]
 | +-rw priority uint8
 | +-rw bandwidth? te-bandwidth
 +-rw te-default-metric? uint32
 +-rw te-delay-metric? uint32
 +-rw te-srlgs
 | +-rw value* te-types:srlg
 +-rw te-nsrlgs {nsrlg}?
 +-rw id* uint32
augment /nw:networks/nw:network:
 +-rw provider-id? te-types:te-global-id
 +-rw client-id? te-types:te-global-id
 +-rw te-topology-id? te-types:te-topology-id
 +-rw te!
 +-rw config
 +-rw preference? uint8
 +-rw optimization-criterion? identityref
 +-rw nsrlg* [id] {nsrlg}?
 +-rw id uint32

```



```

 | +---rw disjointness? te-path-disjointness
+---ro state
 +---ro preference? uint8
 +---ro optimization-criterion? identityref
 +---ro nsrlg* [id] {nsrlg}?
 | +---ro id uint32
 | +---ro disjointness? te-path-disjointness
 +---ro geolocation
 | +---ro altitude? int64
 | +---ro latitude? geographic-coordinate-degree
 | +---ro longitude? geographic-coordinate-degree
augment /nw:networks/nw:network/nw:node:
 +---rw te-node-id? te-types:te-node-id
 +---rw te!
 +---rw config
 | +---rw te-node-template* leafref {template}?
 | +---rw te-node-attributes
 | | +---rw admin-status? te-types:te-admin-status
 | | +---rw connectivity-matrices
 | | | +---rw number-of-entries? uint16
 | | | +---rw is-allowed? boolean
 | | | +---rw label-restriction* [inclusive-exclusive label-
start] | | | +---rw inclusive-exclusive enumeration
 | | | +---rw label-start rt-types:generalized-
label | | | +---rw label-end? rt-types:generalized-
label | | | +---rw range-bitmap? binary
 | | +---rw underlay! {te-topology-hierarchy}?
 | | +---rw primary-path
 | | | +---rw network-ref? leafref
 | | | +---rw path-element* [path-element-id]
 | | | | +---rw path-element-id uint32
 | | | | +---rw index? uint32
 | | | | +---rw (type)?
 | | | | +---:(ip-address)
 | | | | | +---rw ip-address-hop
 | | | | | | +---rw address? inet:ip-address
 | | | | | | +---rw hop-type? te-hop-type
 | | | | +---:(as-number)
 | | | | | +---rw as-number-hop
 | | | | | | +---rw as-number? binary
 | | | | | | +---rw hop-type? te-hop-type
 | | | | +---:(unnumbered-link)
 | | | | | +---rw unnumbered-hop

```

|         |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------|--|--|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| address |  |  |  |  | <pre>       +--rw router-id?      inet:ip-       +--rw interface-id?   uint32       +--rw hop-type?       te-hop-type       +--:(label)       +--rw label-hop       +--rw value?          rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| label   |  |  |  |  | <pre>       +--:(sid)       +--rw sid-hop       +--rw sid?            rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| label   |  |  |  |  | <pre>       +--rw backup-path* [index]       +--rw index            uint32       +--rw network-ref?     leafref       +--rw path-element* [path-element-id]       +--rw path-element-id  uint32       +--rw index?           uint32       +--rw (type)?       +--:(ip-address)           +--rw ip-address-hop               +--rw address?      inet:ip-address               +--rw hop-type?     te-hop-type           +--:(as-number)               +--rw as-number-hop                   +--rw as-number?  binary                   +--rw hop-type?   te-hop-type           +--:(unnumbered-link)               +--rw unnumbered-hop                   +--rw router-id?   inet:ip- </pre> |
| address |  |  |  |  | <pre>       +--rw interface-id?   uint32       +--rw hop-type?       te-hop-type       +--:(label)       +--rw label-hop       +--rw value?          rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| label   |  |  |  |  | <pre>       +--:(sid)       +--rw sid-hop       +--rw sid?            rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| label   |  |  |  |  | <pre>       +--rw protection-type? uint16       +--rw tunnels           +--rw sharing?      boolean           +--rw tunnel* [tunnel-name]               +--rw tunnel-name  string </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

```

| +--rw sharing? boolean
+--rw max-lsp-bandwidth* [priority]
| +--rw priority uint8
| +--rw bandwidth? te-bandwidth
+--rw max-link-bandwidth? te-bandwidth
+--rw max-resv-link-bandwidth? te-bandwidth
+--rw unreserved-bandwidth* [priority]
| +--rw priority uint8
| +--rw bandwidth? te-bandwidth
+--rw te-default-metric? uint32
+--rw te-delay-metric? uint32
+--rw te-srlgs
| +--rw value* te-types:srlg
+--rw te-nsrlgs {nsrlg}?
| +--rw id* uint32
+--rw connectivity-matrix* [id]
| +--rw id uint32
| +--rw from
| | +--rw tp-ref? leafref
| +--rw to
| | +--rw tp-ref? leafref
| +--rw is-allowed? boolean
| +--rw label-restriction* [inclusive-exclusive
label-start]
| | +--rw inclusive-exclusive enumeration
| | +--rw label-start rt-
types:generalized-label
| | +--rw label-end? rt-
types:generalized-label
| | +--rw range-bitmap? binary
+--rw underlay! {te-topology-hierarchy}?
| +--rw primary-path
| | +--rw network-ref? leafref
| | +--rw path-element* [path-element-id]
| | | +--rw path-element-id uint32
| | | +--rw index? uint32
| | | +--rw (type)?
| | | | +--:(ip-address)
| | | | | +--rw ip-address-hop
| | | | | | +--rw address? inet:ip-address
| | | | | | +--rw hop-type? te-hop-type
| | | | +--:(as-number)
| | | | | +--rw as-number-hop
| | | | | | +--rw as-number? binary
| | | | | | +--rw hop-type? te-hop-type
| | | +--:(unnumbered-link)

```

|                         |  |  |  |                         |                       |                 |
|-------------------------|--|--|--|-------------------------|-----------------------|-----------------|
|                         |  |  |  |                         | +---rw unnumbered-hop |                 |
|                         |  |  |  |                         | +---rw router-id?     | inet:ip-        |
| address                 |  |  |  |                         |                       |                 |
|                         |  |  |  |                         | +---rw interface-id?  | uint32          |
|                         |  |  |  |                         | +---rw hop-type?      | te-hop-type     |
|                         |  |  |  | +++:(label)             |                       |                 |
|                         |  |  |  |                         | +---rw label-hop      |                 |
| types:generalized-label |  |  |  |                         | +---rw value?         | rt-             |
|                         |  |  |  | +++:(sid)               |                       |                 |
|                         |  |  |  |                         | +---rw sid-hop        |                 |
| types:generalized-label |  |  |  |                         | +---rw sid?           | rt-             |
|                         |  |  |  | +---rw backup-path*     | [index]               |                 |
|                         |  |  |  | +---rw index            |                       | uint32          |
|                         |  |  |  | +---rw network-ref?     |                       | leafref         |
|                         |  |  |  | +---rw path-element*    | [path-element-id]     |                 |
|                         |  |  |  | +---rw path-element-id  |                       | uint32          |
|                         |  |  |  | +---rw index?           |                       | uint32          |
|                         |  |  |  | +---rw (type)?          |                       |                 |
|                         |  |  |  | +++:(ip-address)        |                       |                 |
|                         |  |  |  |                         | +---rw ip-address-hop |                 |
|                         |  |  |  |                         | +---rw address?       | inet:ip-address |
|                         |  |  |  |                         | +---rw hop-type?      | te-hop-type     |
|                         |  |  |  | +++:(as-number)         |                       |                 |
|                         |  |  |  |                         | +---rw as-number-hop  |                 |
|                         |  |  |  |                         | +---rw as-number?     | binary          |
|                         |  |  |  |                         | +---rw hop-type?      | te-hop-type     |
|                         |  |  |  | +++:(unnumbered-link)   |                       |                 |
|                         |  |  |  |                         | +---rw unnumbered-hop |                 |
| address                 |  |  |  |                         | +---rw router-id?     | inet:ip-        |
|                         |  |  |  |                         |                       |                 |
|                         |  |  |  |                         | +---rw interface-id?  | uint32          |
|                         |  |  |  |                         | +---rw hop-type?      | te-hop-type     |
|                         |  |  |  | +++:(label)             |                       |                 |
|                         |  |  |  |                         | +---rw label-hop      |                 |
| types:generalized-label |  |  |  |                         | +---rw value?         | rt-             |
|                         |  |  |  | +++:(sid)               |                       |                 |
|                         |  |  |  |                         | +---rw sid-hop        |                 |
| types:generalized-label |  |  |  |                         | +---rw sid?           | rt-             |
|                         |  |  |  | +---rw protection-type? |                       | uint16          |
|                         |  |  |  | +---rw tunnels          |                       |                 |
|                         |  |  |  | +---rw sharing?         |                       | boolean         |
|                         |  |  |  | +---rw tunnel*          | [tunnel-name]         |                 |

```

| | | | |--rw tunnel-name string
| | | | |--rw sharing? boolean
|--rw max-lsp-bandwidth* [priority]
| | | | |--rw priority uint8
| | | | |--rw bandwidth? te-bandwidth
|--rw max-link-bandwidth? te-bandwidth
|--rw max-resv-link-bandwidth? te-bandwidth
|--rw unreserved-bandwidth* [priority]
| | | | |--rw priority uint8
| | | | |--rw bandwidth? te-bandwidth
|--rw te-default-metric? uint32
|--rw te-delay-metric? uint32
|--rw te-srlgs
| | | | |--rw value* te-types:srlg
|--rw te-nsrlgs {nsrlg}?
| | | | |--rw id* uint32
|--rw domain-id? uint32
|--rw is-abstract? empty
|--rw name? inet:domain-name
|--rw signaling-address* inet:ip-address
|--rw underlay-topology {te-topology-hierarchy}?
| | | | |--rw network-ref? leafref
+--ro state
| +--ro te-node-template* leafref {template}?
| +--ro te-node-attributes
| |--ro admin-status? te-types:te-admin-status
| +--ro connectivity-matrices
| |--ro number-of-entries? uint16
| |--ro is-allowed? boolean
start] | |--ro label-restriction* [inclusive-exclusive label-
label | |--ro inclusive-exclusive enumeration
label | |--ro label-start rt-types:generalized-
| |--ro label-end? rt-types:generalized-
label |
| |--ro range-bitmap? binary
+--ro underlay! {te-topology-hierarchy}?
| +--ro primary-path
| |--ro network-ref? leafref
| |--ro path-element* [path-element-id]
| |--ro path-element-id uint32
| |--ro index? uint32
| +--ro (type)?
| |--:(ip-address)
| |--ro ip-address-hop
```

[illegible]

```

+--ro sid-hop
+--ro sid? rt-types:generalized-
label
+--ro protection-type? uint16
+--ro tunnels
+--ro sharing? boolean
+--ro tunnel* [tunnel-name]
+--ro tunnel-name string
+--ro sharing? boolean
+--ro max-lsp-bandwidth* [priority]
+--ro priority uint8
+--ro bandwidth? te-bandwidth
+--ro max-link-bandwidth? te-bandwidth
+--ro max-resv-link-bandwidth? te-bandwidth
+--ro unreserved-bandwidth* [priority]
+--ro priority uint8
+--ro bandwidth? te-bandwidth
+--ro te-default-metric? uint32
+--ro te-delay-metric? uint32
+--ro te-srlgs
+--ro value* te-types:srlg
+--ro te-nsrlgs {nsrlg}?
+--ro id* uint32
+--ro connectivity-matrix* [id]
+--ro id uint32
+--ro from
+--ro tp-ref? leafref
+--ro to
+--ro tp-ref? leafref
+--ro is-allowed? boolean
+--ro label-restriction* [inclusive-exclusive
label-start]
+--ro inclusive-exclusive enumeration
+--ro label-start rt-
types:generalized-label
+--ro label-end? rt-
types:generalized-label
+--ro range-bitmap? binary
+--ro underlay! {te-topology-hierarchy}?
+--ro primary-path
+--ro network-ref? leafref
+--ro path-element* [path-element-id]
+--ro path-element-id uint32
+--ro index? uint32
+--ro (type)?
+--:(ip-address)

```

|                         |  |  |  |                                       |                      |                 |
|-------------------------|--|--|--|---------------------------------------|----------------------|-----------------|
|                         |  |  |  |                                       | +--ro ip-address-hop |                 |
|                         |  |  |  |                                       | +--ro address?       | inet:ip-address |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(as-number)                        |                      |                 |
|                         |  |  |  |                                       | +--ro as-number-hop  |                 |
|                         |  |  |  |                                       | +--ro as-number?     | binary          |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(unnumbered-link)                  |                      |                 |
|                         |  |  |  |                                       | +--ro unnumbered-hop |                 |
| address                 |  |  |  |                                       | +--ro router-id?     | inet:ip-        |
|                         |  |  |  |                                       |                      |                 |
|                         |  |  |  |                                       | +--ro interface-id?  | uint32          |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(label)                            |                      |                 |
|                         |  |  |  |                                       | +--ro label-hop      |                 |
| types:generalized-label |  |  |  |                                       | +--ro value?         | rt-             |
|                         |  |  |  |                                       |                      |                 |
|                         |  |  |  | ++:(sid)                              |                      |                 |
|                         |  |  |  |                                       | +--ro sid-hop        |                 |
| types:generalized-label |  |  |  |                                       | +--ro sid?           | rt-             |
|                         |  |  |  |                                       |                      |                 |
|                         |  |  |  | +--ro backup-path* [index]            |                      |                 |
|                         |  |  |  | +--ro index                           | uint32               |                 |
|                         |  |  |  | +--ro network-ref?                    | leafref              |                 |
|                         |  |  |  | +--ro path-element* [path-element-id] |                      |                 |
|                         |  |  |  | +--ro path-element-id                 | uint32               |                 |
|                         |  |  |  | +--ro index?                          | uint32               |                 |
|                         |  |  |  | +--ro (type)?                         |                      |                 |
|                         |  |  |  | ++:(ip-address)                       |                      |                 |
|                         |  |  |  |                                       | +--ro ip-address-hop |                 |
|                         |  |  |  |                                       | +--ro address?       | inet:ip-address |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(as-number)                        |                      |                 |
|                         |  |  |  |                                       | +--ro as-number-hop  |                 |
|                         |  |  |  |                                       | +--ro as-number?     | binary          |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(unnumbered-link)                  |                      |                 |
|                         |  |  |  |                                       | +--ro unnumbered-hop |                 |
| address                 |  |  |  |                                       | +--ro router-id?     | inet:ip-        |
|                         |  |  |  |                                       |                      |                 |
|                         |  |  |  |                                       | +--ro interface-id?  | uint32          |
|                         |  |  |  |                                       | +--ro hop-type?      | te-hop-type     |
|                         |  |  |  | ++:(label)                            |                      |                 |
|                         |  |  |  |                                       | +--ro label-hop      |                 |
| types:generalized-label |  |  |  |                                       | +--ro value?         | rt-             |



```

types:generalized-label
 +--:(sid)
 +--ro sid-hop
 +--ro sid? rt-
 +--ro protection-type? uint16
 +--ro tunnels
 +--ro sharing? boolean
 +--ro tunnel* [tunnel-name]
 +--ro tunnel-name string
 +--ro sharing? boolean
 +--ro max-lsp-bandwidth* [priority]
 +--ro priority uint8
 +--ro bandwidth? te-bandwidth
 +--ro max-link-bandwidth? te-bandwidth
 +--ro max-resv-link-bandwidth? te-bandwidth
 +--ro unreserved-bandwidth* [priority]
 +--ro priority uint8
 +--ro bandwidth? te-bandwidth
 +--ro te-default-metric? uint32
 +--ro te-delay-metric? uint32
 +--ro te-srlgs
 | +--ro value* te-types:srlg
 +--ro te-nsrlgs {nsrlg}?
 +--ro id* uint32
 +--ro domain-id? uint32
 +--ro is-abstract? empty
 +--ro name? inet:domain-name
 +--ro signaling-address* inet:ip-address
 +--ro underlay-topology {te-topology-hierarchy}?
 +--ro network-ref? leafref
 +--ro oper-status? te-types:te-oper-status
 +--ro geolocation
 | +--ro altitude? int64
 | +--ro latitude? geographic-coordinate-degree
 | +--ro longitude? geographic-coordinate-degree
 +--ro is-multi-access-dr? empty
 +--ro information-source? te-info-source
 +--ro information-source-state
 | +--ro credibility-preference? uint16
 | +--ro logical-network-element? string
 | +--ro network-instance? string
 +--ro topology
 | +--ro network-ref? leafref
 | +--ro node-ref? leafref
 +--ro information-source-entry* [information-source]
 +--ro information-source te-info-source

```

```

| +--ro information-source-state
| | +--ro credibility-preference? uint16
| | +--ro logical-network-element? string
| | +--ro network-instance? string
| | +--ro topology
| | | +--ro network-ref? leafref
| | | +--ro node-ref? leafref
| +--ro connectivity-matrices
| | +--ro number-of-entries? uint16
| | +--ro is-allowed? boolean
| | +--ro label-restriction* [inclusive-exclusive label-
start] |
| | | +--ro inclusive-exclusive enumeration
| | | +--ro label-start rt-types:generalized-
label |
| | | +--ro label-end? rt-types:generalized-
label |
| | | +--ro range-bitmap? binary
| +--ro underlay! {te-topology-hierarchy}?
| | +--ro primary-path
| | | +--ro network-ref? leafref
| | | +--ro path-element* [path-element-id]
| | | | +--ro path-element-id uint32
| | | | +--ro index? uint32
| | | | +--ro (type)?
| | | | | +--:(ip-address)
| | | | | | +--ro ip-address-hop
| | | | | | | +--ro address? inet:ip-address
| | | | | | | +--ro hop-type? te-hop-type
| | | | | +--:(as-number)
| | | | | | +--ro as-number-hop
| | | | | | | +--ro as-number? binary
| | | | | | | +--ro hop-type? te-hop-type
| | | | +--:(unnumbered-link)
| | | | | +--ro unnumbered-hop
| | | | | | +--ro router-id? inet:ip-
address |
| | | | | | +--ro interface-id? uint32
| | | | | | +--ro hop-type? te-hop-type
| | | | +--:(label)
| | | | | +--ro label-hop
| | | | | | +--ro value? rt-types:generalized-
label |
| | | | | +--:(sid)
| | | | | | +--ro sid-hop

```

|         |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| label   |  |  |  | <pre>       +--ro sid?    rt-types:generalized-     </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|         |  |  |  | <pre>     +--ro backup-path* [index]       +--ro index          uint32       +--ro network-ref?   leafref       +--ro path-element* [path-element-id]         +--ro path-element-id  uint32         +--ro index?          uint32         +--ro (type)?           +--:(ip-address)             +--ro ip-address-hop               +--ro address?   inet:ip-address               +--ro hop-type?  te-hop-type           +--:(as-number)             +--ro as-number-hop               +--ro as-number?  binary               +--ro hop-type?   te-hop-type           +--:(unnumbered-link)             +--ro unnumbered-hop               +--ro router-id?   inet:ip-     </pre> |
| address |  |  |  | <pre>       +--ro interface-id?  uint32       +--ro hop-type?      te-hop-type     +--:(label)       +--ro label-hop         +--ro value?    rt-types:generalized-     </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| label   |  |  |  | <pre>     +--:(sid)       +--ro sid-hop         +--ro sid?    rt-types:generalized-     </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| label   |  |  |  | <pre>     +--ro protection-type?  uint16     +--ro tunnels       +--ro sharing?        boolean       +--ro tunnel* [tunnel-name]         +--ro tunnel-name    string         +--ro sharing?        boolean     +--ro max-lsp-bandwidth* [priority]       +--ro priority        uint8       +--ro bandwidth?      te-bandwidth     +--ro max-link-bandwidth?      te-bandwidth     +--ro max-resv-link-bandwidth? te-bandwidth     +--ro unreserved-bandwidth* [priority]       +--ro priority        uint8       +--ro bandwidth?      te-bandwidth     +--ro te-default-metric?      uint32     </pre>                                                                         |

|                         |  |  |  |                                               |                 |
|-------------------------|--|--|--|-----------------------------------------------|-----------------|
|                         |  |  |  | +++ro te-delay-metric?                        | uint32          |
|                         |  |  |  | +++ro te-srlgs                                |                 |
|                         |  |  |  | +++ro value*                                  | te-types:srlg   |
|                         |  |  |  | +++ro te-nsrlgs {nsrlg}?                      |                 |
|                         |  |  |  | +++ro id*                                     | uint32          |
|                         |  |  |  | +++ro connectivity-matrix* [id]               |                 |
|                         |  |  |  | +++ro id                                      | uint32          |
|                         |  |  |  | +++ro from                                    |                 |
|                         |  |  |  | +++ro tp-ref?                                 | leafref         |
|                         |  |  |  | +++ro to                                      |                 |
|                         |  |  |  | +++ro tp-ref?                                 | leafref         |
|                         |  |  |  | +++ro is-allowed?                             | boolean         |
|                         |  |  |  | +++ro label-restriction* [inclusive-exclusive |                 |
| label-start]            |  |  |  |                                               |                 |
|                         |  |  |  | +++ro inclusive-exclusive                     | enumeration     |
|                         |  |  |  | +++ro label-start                             | rt-             |
| types:generalized-label |  |  |  |                                               |                 |
|                         |  |  |  | +++ro label-end?                              | rt-             |
| types:generalized-label |  |  |  |                                               |                 |
|                         |  |  |  | +++ro range-bitmap?                           | binary          |
|                         |  |  |  | +++ro underlay! {te-topology-hierarchy}?      |                 |
|                         |  |  |  | +++ro primary-path                            |                 |
|                         |  |  |  | +++ro network-ref?                            | leafref         |
|                         |  |  |  | +++ro path-element* [path-element-id]         |                 |
|                         |  |  |  | +++ro path-element-id                         | uint32          |
|                         |  |  |  | +++ro index?                                  | uint32          |
|                         |  |  |  | +++ro (type)?                                 |                 |
|                         |  |  |  | +++:(ip-address)                              |                 |
|                         |  |  |  | +++ro ip-address-hop                          |                 |
|                         |  |  |  | +++ro address?                                | inet:ip-address |
|                         |  |  |  | +++ro hop-type?                               | te-hop-type     |
|                         |  |  |  | +++:(as-number)                               |                 |
|                         |  |  |  | +++ro as-number-hop                           |                 |
|                         |  |  |  | +++ro as-number?                              | binary          |
|                         |  |  |  | +++ro hop-type?                               | te-hop-type     |
|                         |  |  |  | +++:(unnumbered-link)                         |                 |
|                         |  |  |  | +++ro unnumbered-hop                          |                 |
|                         |  |  |  | +++ro router-id?                              | inet:ip-        |
| address                 |  |  |  |                                               |                 |
|                         |  |  |  | +++ro interface-id?                           | uint32          |
|                         |  |  |  | +++ro hop-type?                               | te-hop-type     |
|                         |  |  |  | +++:(label)                                   |                 |
|                         |  |  |  | +++ro label-hop                               |                 |
|                         |  |  |  | +++ro value?                                  | rt-             |
| types:generalized-label |  |  |  |                                               |                 |
|                         |  |  |  | +++:(sid)                                     |                 |

```

+--ro sid-hop
+--ro sid? rt-
types:generalized-label
+--ro backup-path* [index]
+--ro index uint32
+--ro network-ref? leafref
+--ro path-element* [path-element-id]
+--ro path-element-id uint32
+--ro index? uint32
+--ro (type)?
+--:(ip-address)
| +--ro ip-address-hop
| +--ro address? inet:ip-address
| +--ro hop-type? te-hop-type
+--:(as-number)
| +--ro as-number-hop
| +--ro as-number? binary
| +--ro hop-type? te-hop-type
+--:(unnumbered-link)
| +--ro unnumbered-hop
| +--ro router-id? inet:ip-
address
| +--ro interface-id? uint32
| +--ro hop-type? te-hop-type
+--:(label)
| +--ro label-hop
| +--ro value? rt-
types:generalized-label
+--:(sid)
+--ro sid-hop
+--ro sid? rt-
types:generalized-label
+--ro protection-type? uint16
+--ro tunnels
+--ro sharing? boolean
+--ro tunnel* [tunnel-name]
+--ro tunnel-name string
+--ro sharing? boolean
+--ro max-lsp-bandwidth* [priority]
| +--ro priority uint8
| +--ro bandwidth? te-bandwidth
+--ro max-link-bandwidth? te-bandwidth
+--ro max-resv-link-bandwidth? te-bandwidth
+--ro unreserved-bandwidth* [priority]
| +--ro priority uint8
| +--ro bandwidth? te-bandwidth

```

```

| | +---ro te-default-metric? uint32
| | +---ro te-delay-metric? uint32
| | +---ro te-srlgs
| | | +---ro value* te-types:srlg
| | +---ro te-nsrlgs {nsrlg}?
| | +---ro id* uint32
| +---ro domain-id? uint32
| +---ro is-abstract? empty
| +---ro name? inet:domain-name
| +---ro signaling-address* inet:ip-address
| +---ro underlay-topology {te-topology-hierarchy}?
| | +---ro network-ref? leafref
+---ro statistics
| +---ro discontinuity-time yang:date-and-time
| +---ro node
| | +---ro disables? yang:counter32
| | +---ro enables? yang:counter32
| | +---ro maintenance-sets? yang:counter32
| | +---ro maintenance-clears? yang:counter32
| | +---ro modifies? yang:counter32
| +---ro connectivity-matrix-entry
| | +---ro creates? yang:counter32
| | +---ro deletes? yang:counter32
| | +---ro disables? yang:counter32
| | +---ro enables? yang:counter32
| | +---ro modifies? yang:counter32
+---rw tunnel-termination-point* [tunnel-tp-id]
| +---rw tunnel-tp-id binary
| +---rw config
| | +---rw switching-capability? identityref
| | +---rw encoding? identityref
| | +---rw inter-layer-lock-id? uint32
| | +---rw protection-type? identityref
| | +---rw client-layer-adaptation
| | | +---rw switching-capability* [switching-capability
encoding] |
| | | +---rw switching-capability identityref
| | | +---rw encoding identityref
| | | +---rw bandwidth? te-bandwidth
| +---rw local-link-connectivities
| | +---rw number-of-entries? uint16
| | +---rw is-allowed? boolean
| | +---rw label-restriction* [inclusive-exclusive label-
start] |
| | | +---rw inclusive-exclusive enumeration

```

|         |  |  |                                          |                       |
|---------|--|--|------------------------------------------|-----------------------|
| label   |  |  | +++rw label-start                        | rt-types:generalized- |
| label   |  |  | +++rw label-end?                         | rt-types:generalized- |
|         |  |  | +++rw range-bitmap?                      | binary                |
|         |  |  | +++rw underlay! {te-topology-hierarchy}? |                       |
|         |  |  | +++rw primary-path                       |                       |
|         |  |  | +++rw network-ref?                       | leafref               |
|         |  |  | +++rw path-element* [path-element-id]    |                       |
|         |  |  | +++rw path-element-id                    | uint32                |
|         |  |  | +++rw index?                             | uint32                |
|         |  |  | +++rw (type)?                            |                       |
|         |  |  | +++:(ip-address)                         |                       |
|         |  |  | +++rw ip-address-hop                     |                       |
|         |  |  | +++rw address?                           | inet:ip-address       |
|         |  |  | +++rw hop-type?                          | te-hop-type           |
|         |  |  | +++:(as-number)                          |                       |
|         |  |  | +++rw as-number-hop                      |                       |
|         |  |  | +++rw as-number?                         | binary                |
|         |  |  | +++rw hop-type?                          | te-hop-type           |
|         |  |  | +++:(unnumbered-link)                    |                       |
|         |  |  | +++rw unnumbered-hop                     |                       |
| address |  |  | +++rw router-id?                         | inet:ip-              |
|         |  |  | +++rw interface-id?                      | uint32                |
|         |  |  | +++rw hop-type?                          | te-hop-type           |
|         |  |  | +++:(label)                              |                       |
|         |  |  | +++rw label-hop                          |                       |
| label   |  |  | +++rw value?                             | rt-types:generalized- |
|         |  |  | +++:(sid)                                |                       |
|         |  |  | +++rw sid-hop                            |                       |
| label   |  |  | +++rw sid?                               | rt-types:generalized- |
|         |  |  | +++rw backup-path* [index]               |                       |
|         |  |  | +++rw index                              | uint32                |
|         |  |  | +++rw network-ref?                       | leafref               |
|         |  |  | +++rw path-element* [path-element-id]    |                       |
|         |  |  | +++rw path-element-id                    | uint32                |
|         |  |  | +++rw index?                             | uint32                |
|         |  |  | +++rw (type)?                            |                       |
|         |  |  | +++:(ip-address)                         |                       |
|         |  |  | +++rw ip-address-hop                     |                       |
|         |  |  | +++rw address?                           | inet:ip-address       |
|         |  |  | +++rw hop-type?                          | te-hop-type           |
|         |  |  | +++:(as-number)                          |                       |

|                         |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------|--|--|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |  |  |  |  | +--rw as-number-hop<br>+--rw as-number?    binary<br>+--rw hop-type?    te-hop-type<br>+--:(unnumbered-link)<br>+--rw unnumbered-hop<br>+--rw router-id?    inet:ip-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| address                 |  |  |  |  | +--rw interface-id?    uint32<br>+--rw hop-type?    te-hop-type<br>+--:(label)<br>+--rw label-hop<br>+--rw value?    rt-types:generalized-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| label                   |  |  |  |  | +--:(sid)<br>+--rw sid-hop<br>+--rw sid?    rt-types:generalized-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| label                   |  |  |  |  | +--rw protection-type?    uint16<br>+--rw tunnels<br>+--rw sharing?    boolean<br>+--rw tunnel* [tunnel-name]<br>+--rw tunnel-name    string<br>+--rw sharing?    boolean<br>+--rw max-lsp-bandwidth* [priority]<br>+--rw priority    uint8<br>+--rw bandwidth?    te-bandwidth<br>+--rw max-link-bandwidth?    te-bandwidth<br>+--rw max-resv-link-bandwidth?    te-bandwidth<br>+--rw unreserved-bandwidth* [priority]<br>+--rw priority    uint8<br>+--rw bandwidth?    te-bandwidth<br>+--rw te-default-metric?    uint32<br>+--rw te-delay-metric?    uint32<br>+--rw te-srlgs<br>+--rw value*    te-types:srlg<br>+--rw te-nsrlgs {nsrlg}?<br>+--rw id*    uint32<br>+--rw local-link-connectivity* [link-tp-ref]<br>+--rw link-tp-ref    leafref<br>+--rw is-allowed?    boolean<br>+--rw label-restriction* [inclusive-exclusive |
| label-start]            |  |  |  |  | +--rw inclusive-exclusive    enumeration<br>+--rw label-start    rt-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| types:generalized-label |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |



|                         |  |  |                                          |                 |
|-------------------------|--|--|------------------------------------------|-----------------|
|                         |  |  | +--rw label-end?                         | rt-             |
| types:generalized-label |  |  |                                          |                 |
|                         |  |  | +--rw range-bitmap?                      | binary          |
|                         |  |  | +--rw underlay! {te-topology-hierarchy}? |                 |
|                         |  |  | +--rw primary-path                       |                 |
|                         |  |  | +--rw network-ref?                       | leafref         |
|                         |  |  | +--rw path-element* [path-element-id]    |                 |
|                         |  |  | +--rw path-element-id                    | uint32          |
|                         |  |  | +--rw index?                             | uint32          |
|                         |  |  | +--rw (type)?                            |                 |
|                         |  |  | +--:(ip-address)                         |                 |
|                         |  |  | +--rw ip-address-hop                     |                 |
|                         |  |  | +--rw address?                           | inet:ip-address |
|                         |  |  | +--rw hop-type?                          | te-hop-type     |
|                         |  |  | +--:(as-number)                          |                 |
|                         |  |  | +--rw as-number-hop                      |                 |
|                         |  |  | +--rw as-number?                         | binary          |
|                         |  |  | +--rw hop-type?                          | te-hop-type     |
|                         |  |  | +--:(unnumbered-link)                    |                 |
|                         |  |  | +--rw unnumbered-hop                     |                 |
| address                 |  |  | +--rw router-id?                         | inet:ip-        |
|                         |  |  |                                          |                 |
|                         |  |  | +--rw interface-id?                      | uint32          |
|                         |  |  | +--rw hop-type?                          | te-hop-type     |
|                         |  |  | +--:(label)                              |                 |
|                         |  |  | +--rw label-hop                          |                 |
|                         |  |  | +--rw value?                             | rt-             |
| types:generalized-label |  |  |                                          |                 |
|                         |  |  | +--:(sid)                                |                 |
|                         |  |  | +--rw sid-hop                            |                 |
|                         |  |  | +--rw sid?                               | rt-             |
| types:generalized-label |  |  |                                          |                 |
|                         |  |  | +--rw backup-path* [index]               |                 |
|                         |  |  | +--rw index                              | uint32          |
|                         |  |  | +--rw network-ref?                       | leafref         |
|                         |  |  | +--rw path-element* [path-element-id]    |                 |
|                         |  |  | +--rw path-element-id                    | uint32          |
|                         |  |  | +--rw index?                             | uint32          |
|                         |  |  | +--rw (type)?                            |                 |
|                         |  |  | +--:(ip-address)                         |                 |
|                         |  |  | +--rw ip-address-hop                     |                 |
|                         |  |  | +--rw address?                           | inet:ip-address |
|                         |  |  | +--rw hop-type?                          | te-hop-type     |
|                         |  |  | +--:(as-number)                          |                 |
|                         |  |  | +--rw as-number-hop                      |                 |
|                         |  |  | +--rw as-number?                         | binary          |

|                         |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------|--|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |  |  |  | +--rw hop-type?      te-hop-type<br>+--:(unnumbered-link)<br>+--rw unnumbered-hop<br>+--rw router-id?      inet:ip-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| address                 |  |  |  | +--rw interface-id?    uint32<br>+--rw hop-type?      te-hop-type<br>+--:(label)<br>+--rw label-hop<br>+--rw value?      rt-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| types:generalized-label |  |  |  | +--:(sid)<br>+--rw sid-hop<br>+--rw sid?      rt-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| types:generalized-label |  |  |  | +--rw protection-type?    uint16<br>+--rw tunnels<br>+--rw sharing?      boolean<br>+--rw tunnel* [tunnel-name]<br>+--rw tunnel-name      string<br>+--rw sharing?      boolean<br>+--rw max-lsp-bandwidth* [priority]<br>+--rw priority      uint8<br>+--rw bandwidth?      te-bandwidth<br>+--rw max-link-bandwidth?      te-bandwidth<br>+--rw max-resv-link-bandwidth?    te-bandwidth<br>+--rw unreserved-bandwidth* [priority]<br>+--rw priority      uint8<br>+--rw bandwidth?      te-bandwidth<br>+--rw te-default-metric?      uint32<br>+--rw te-delay-metric?      uint32<br>+--rw te-srlgs<br>+--rw value*      te-types:srlg<br>+--rw te-nsrlgs {nsrlg}?<br>+--rw id*      uint32 |
|                         |  |  |  | +--ro state<br>+--ro switching-capability?      identityref<br>+--ro encoding?      identityref<br>+--ro inter-layer-lock-id?      uint32<br>+--ro protection-type?      identityref<br>+--ro client-layer-adaptation<br>+--ro switching-capability* [switching-capability                                                                                                                                                                                                                                                                                                                                                                                                                      |
| encoding]               |  |  |  | +--ro switching-capability      identityref<br>+--ro encoding      identityref<br>+--ro bandwidth?      te-bandwidth                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|         |  |  |  |  |                                                      |                       |
|---------|--|--|--|--|------------------------------------------------------|-----------------------|
|         |  |  |  |  | +++ro local-link-connectivities                      |                       |
|         |  |  |  |  | +++ro number-of-entries?                             | uint16                |
|         |  |  |  |  | +++ro is-allowed?                                    | boolean               |
|         |  |  |  |  | +++ro label-restriction* [inclusive-exclusive label- |                       |
| start]  |  |  |  |  |                                                      |                       |
|         |  |  |  |  | +++ro inclusive-exclusive                            | enumeration           |
| label   |  |  |  |  | +++ro label-start                                    | rt-types:generalized- |
|         |  |  |  |  |                                                      |                       |
| label   |  |  |  |  | +++ro label-end?                                     | rt-types:generalized- |
|         |  |  |  |  |                                                      |                       |
|         |  |  |  |  | +++ro range-bitmap?                                  | binary                |
|         |  |  |  |  | +++ro underlay! {te-topology-hierarchy}?             |                       |
|         |  |  |  |  | +++ro primary-path                                   |                       |
|         |  |  |  |  | +++ro network-ref?                                   | leafref               |
|         |  |  |  |  | +++ro path-element* [path-element-id]                |                       |
|         |  |  |  |  | +++ro path-element-id                                | uint32                |
|         |  |  |  |  | +++ro index?                                         | uint32                |
|         |  |  |  |  | +++ro (type)?                                        |                       |
|         |  |  |  |  | +++:(ip-address)                                     |                       |
|         |  |  |  |  | +++ro ip-address-hop                                 |                       |
|         |  |  |  |  | +++ro address?                                       | inet:ip-address       |
|         |  |  |  |  | +++ro hop-type?                                      | te-hop-type           |
|         |  |  |  |  | +++:(as-number)                                      |                       |
|         |  |  |  |  | +++ro as-number-hop                                  |                       |
|         |  |  |  |  | +++ro as-number?                                     | binary                |
|         |  |  |  |  | +++ro hop-type?                                      | te-hop-type           |
|         |  |  |  |  | +++:(unnumbered-link)                                |                       |
|         |  |  |  |  | +++ro unnumbered-hop                                 |                       |
| address |  |  |  |  | +++ro router-id?                                     | inet:ip-              |
|         |  |  |  |  |                                                      |                       |
|         |  |  |  |  | +++ro interface-id?                                  | uint32                |
|         |  |  |  |  | +++ro hop-type?                                      | te-hop-type           |
|         |  |  |  |  | +++:(label)                                          |                       |
|         |  |  |  |  | +++ro label-hop                                      |                       |
| label   |  |  |  |  | +++ro value?                                         | rt-types:generalized- |
|         |  |  |  |  |                                                      |                       |
|         |  |  |  |  | +++:(sid)                                            |                       |
|         |  |  |  |  | +++ro sid-hop                                        |                       |
| label   |  |  |  |  | +++ro sid?                                           | rt-types:generalized- |
|         |  |  |  |  |                                                      |                       |
|         |  |  |  |  | +++ro backup-path* [index]                           |                       |
|         |  |  |  |  | +++ro index                                          | uint32                |
|         |  |  |  |  | +++ro network-ref?                                   | leafref               |
|         |  |  |  |  | +++ro path-element* [path-element-id]                |                       |
|         |  |  |  |  | +++ro path-element-id                                | uint32                |
|         |  |  |  |  | +++ro index?                                         | uint32                |

|         |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |  |  |  | <pre>       +--ro (type)?       +--:(ip-address)           +--ro ip-address-hop               +--ro address?      inet:ip-address               +--ro hop-type?     te-hop-type       +--:(as-number)           +--ro as-number-hop               +--ro as-number?    binary               +--ro hop-type?     te-hop-type       +--:(unnumbered-link)           +--ro unnumbered-hop               +--ro router-id?     inet:ip- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| address |  |  |  | <pre>               +--ro interface-id?  uint32               +--ro hop-type?     te-hop-type       +--:(label)           +--ro label-hop               +--ro value?        rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| label   |  |  |  | <pre>       +--:(sid)       +--ro sid-hop       +--ro sid?        rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| label   |  |  |  | <pre>       +--ro protection-type?  uint16       +--ro tunnels           +--ro sharing?      boolean           +--ro tunnel* [tunnel-name]               +--ro tunnel-name  string               +--ro sharing?      boolean       +--ro max-lsp-bandwidth* [priority]           +--ro priority      uint8           +--ro bandwidth?    te-bandwidth       +--ro max-link-bandwidth?      te-bandwidth       +--ro max-resv-link-bandwidth? te-bandwidth       +--ro unreserved-bandwidth* [priority]           +--ro priority      uint8           +--ro bandwidth?    te-bandwidth       +--ro te-default-metric?        uint32       +--ro te-delay-metric?          uint32       +--ro te-srlgs           +--ro value*      te-types:srlg       +--ro te-nsrlgs {nsrlg}?           +--ro id*        uint32       +--ro local-link-connectivity* [link-tp-ref]           +--ro link-tp-ref          leafref       +--ro is-allowed?              boolean </pre> |

|                         |  |  |  |                                               |  |
|-------------------------|--|--|--|-----------------------------------------------|--|
|                         |  |  |  | +--ro label-restriction* [inclusive-exclusive |  |
| label-start]            |  |  |  | +--ro inclusive-exclusive enumeration         |  |
|                         |  |  |  | +--ro label-start rt-                         |  |
| types:generalized-label |  |  |  | +--ro label-end? rt-                          |  |
| types:generalized-label |  |  |  | +--ro range-bitmap? binary                    |  |
|                         |  |  |  | +--ro underlay! {te-topology-hierarchy}?      |  |
|                         |  |  |  | +--ro primary-path                            |  |
|                         |  |  |  | +--ro network-ref? leafref                    |  |
|                         |  |  |  | +--ro path-element* [path-element-id]         |  |
|                         |  |  |  | +--ro path-element-id uint32                  |  |
|                         |  |  |  | +--ro index? uint32                           |  |
|                         |  |  |  | +--ro (type)?                                 |  |
|                         |  |  |  | +--:(ip-address)                              |  |
|                         |  |  |  | +--ro ip-address-hop                          |  |
|                         |  |  |  | +--ro address? inet:ip-address                |  |
|                         |  |  |  | +--ro hop-type? te-hop-type                   |  |
|                         |  |  |  | +--:(as-number)                               |  |
|                         |  |  |  | +--ro as-number-hop                           |  |
|                         |  |  |  | +--ro as-number? binary                       |  |
|                         |  |  |  | +--ro hop-type? te-hop-type                   |  |
|                         |  |  |  | +--:(unnumbered-link)                         |  |
|                         |  |  |  | +--ro unnumbered-hop                          |  |
|                         |  |  |  | +--ro router-id? inet:ip-                     |  |
| address                 |  |  |  | +--ro interface-id? uint32                    |  |
|                         |  |  |  | +--ro hop-type? te-hop-type                   |  |
|                         |  |  |  | +--:(label)                                   |  |
|                         |  |  |  | +--ro label-hop                               |  |
|                         |  |  |  | +--ro value? rt-                              |  |
| types:generalized-label |  |  |  | +--:(sid)                                     |  |
|                         |  |  |  | +--ro sid-hop                                 |  |
|                         |  |  |  | +--ro sid? rt-                                |  |
| types:generalized-label |  |  |  | +--ro backup-path* [index]                    |  |
|                         |  |  |  | +--ro index uint32                            |  |
|                         |  |  |  | +--ro network-ref? leafref                    |  |
|                         |  |  |  | +--ro path-element* [path-element-id]         |  |
|                         |  |  |  | +--ro path-element-id uint32                  |  |
|                         |  |  |  | +--ro index? uint32                           |  |
|                         |  |  |  | +--ro (type)?                                 |  |
|                         |  |  |  | +--:(ip-address)                              |  |
|                         |  |  |  | +--ro ip-address-hop                          |  |

[illegible]

```

 | +--ro tunnel-termination-point
 | | +--ro disables? yang:counter32
 | | +--ro enables? yang:counter32
 | | +--ro maintenance-clears? yang:counter32
 | | +--ro maintenance-sets? yang:counter32
 | | +--ro modifies? yang:counter32
 | | +--ro downs? yang:counter32
 | | +--ro ups? yang:counter32
 | | +--ro in-service-clears? yang:counter32
 | | +--ro in-service-sets? yang:counter32
 | +--ro local-link-connectivity
 | | +--ro creates? yang:counter32
 | | +--ro deletes? yang:counter32
 | | +--ro disables? yang:counter32
 | | +--ro enables? yang:counter32
 | | +--ro modifies? yang:counter32
 | +--rw supporting-tunnel-termination-point* [node-ref tunnel-
tp-ref]
 | +--rw node-ref inet:uri
 | +--rw tunnel-tp-ref binary
augment /nw:networks/nw:network/nt:link:
+--rw te!
+--rw config
+--rw (bundle-stack-level)?
+--:(bundle)
+--rw bundled-links
+--rw bundled-link* [sequence]
+--rw sequence uint32
+--rw src-tp-ref? leafref
+--rw des-tp-ref? leafref
+--:(component)
+--rw component-links
+--rw component-link* [sequence]
+--rw sequence uint32
+--rw src-interface-ref? string
+--rw des-interface-ref? string
+--rw te-link-template* leafref {template}?
+--rw te-link-attributes
+--rw access-type? te-types:te-link-
access-type
+--rw external-domain
+--rw network-ref? leafref
+--rw remote-te-node-id? te-types:te-node-id
+--rw remote-te-link-tp-id? te-types:te-tp-id
+--rw plug-id? uint32
+--rw is-abstract? empty

```

```

+---rw name? string
+---rw underlay! {te-topology-hierarchy}?
| +---rw primary-path
| +---rw network-ref? leafref
| +---rw path-element* [path-element-id]
| +---rw path-element-id uint32
| +---rw index? uint32
| +---rw (type)?
| +---:(ip-address)
| +---rw ip-address-hop
| +---rw address? inet:ip-address
| +---rw hop-type? te-hop-type
| +---:(as-number)
| +---rw as-number-hop
| +---rw as-number? binary
| +---rw hop-type? te-hop-type
| +---:(unnumbered-link)
| +---rw unnumbered-hop
| +---rw router-id? inet:ip-address
| +---rw interface-id? uint32
| +---rw hop-type? te-hop-type
| +---:(label)
| +---rw label-hop
| +---rw value? rt-types:generalized-
label
|
| +---:(sid)
| +---rw sid-hop
| +---rw sid? rt-types:generalized-label
+---rw backup-path* [index]
| +---rw index uint32
| +---rw network-ref? leafref
| +---rw path-element* [path-element-id]
| +---rw path-element-id uint32
| +---rw index? uint32
| +---rw (type)?
| +---:(ip-address)
| +---rw ip-address-hop
| +---rw address? inet:ip-address
| +---rw hop-type? te-hop-type
| +---:(as-number)
| +---rw as-number-hop
| +---rw as-number? binary
| +---rw hop-type? te-hop-type
| +---:(unnumbered-link)
| +---rw unnumbered-hop
| +---rw router-id? inet:ip-address

```



```

| | | | +---rw interface-id? uint32
| | | | +---rw hop-type? te-hop-type
| | | | +---:(label)
| | | | | +---rw label-hop
| | | | | +---rw value? rt-types:generalized-
label
| | | | | +---:(sid)
| | | | | | +---rw sid-hop
| | | | | | +---rw sid? rt-types:generalized-label
| | | | | +---rw protection-type? uint16
| | | | | +---rw tunnels
| | | | | | +---rw sharing? boolean
| | | | | | +---rw tunnel* [tunnel-name]
| | | | | | | +---rw tunnel-name string
| | | | | | | +---rw sharing? boolean
| | | | | +---rw admin-status? te-types:te-
admin-status
| | | | | +---rw link-index? uint64
| | | | | +---rw administrative-group? te-types:admin-
groups
| | | | | +---rw interface-switching-capability* [switching-
capability encoding]
| | | | | | +---rw switching-capability identityref
| | | | | | +---rw encoding identityref
| | | | | | +---rw max-lsp-bandwidth* [priority]
| | | | | | | +---rw priority uint8
| | | | | | | +---rw bandwidth? te-bandwidth
| | | | | +---rw link-protection-type? enumeration
| | | | | +---rw max-link-bandwidth? te-bandwidth
| | | | | +---rw max-resv-link-bandwidth? te-bandwidth
| | | | | +---rw unreserved-bandwidth* [priority]
| | | | | | +---rw priority uint8
| | | | | | +---rw bandwidth? te-bandwidth
| | | | | +---rw te-default-metric? uint32
| | | | | +---rw te-delay-metric? uint32
| | | | | +---rw te-srlgs
| | | | | | +---rw value* te-types:srlg
| | | | | +---rw te-nsrlgs {nsrlg}?
| | | | | | +---rw id* uint32
+---ro state
| | | | | +---ro (bundle-stack-level)?
| | | | | | +---:(bundle)
| | | | | | | +---ro bundled-links
| | | | | | | | +---ro bundled-link* [sequence]
| | | | | | | | | +---ro sequence uint32
| | | | | | | | | +---ro src-tp-ref? leafref

```

```

| +--ro des-tp-ref? leafref
| +---:(component)
| +--ro component-links
| +--ro component-link* [sequence]
| +--ro sequence uint32
| +--ro src-interface-ref? string
| +--ro des-interface-ref? string
| +--ro te-link-template* leafref {template}?
| +--ro te-link-attributes
| +--ro access-type? te-types:te-link-
access-type
| +--ro external-domain
| +--ro network-ref? leafref
| +--ro remote-te-node-id? te-types:te-node-id
| +--ro remote-te-link-tp-id? te-types:te-tp-id
| +--ro plug-id? uint32
| +--ro is-abstract? empty
| +--ro name? string
| +--ro underlay! {te-topology-hierarchy}?
| +--ro primary-path
| +--ro network-ref? leafref
| +--ro path-element* [path-element-id]
| +--ro path-element-id uint32
| +--ro index? uint32
| +--ro (type)?
| +---:(ip-address)
| +--ro ip-address-hop
| +--ro address? inet:ip-address
| +--ro hop-type? te-hop-type
| +---:(as-number)
| +--ro as-number-hop
| +--ro as-number? binary
| +--ro hop-type? te-hop-type
| +---:(unnumbered-link)
| +--ro unnumbered-hop
| +--ro router-id? inet:ip-address
| +--ro interface-id? uint32
| +--ro hop-type? te-hop-type
| +---:(label)
| +--ro label-hop
| +--ro value? rt-types:generalized-
label
| +---:(sid)
| +--ro sid-hop
| +--ro sid? rt-types:generalized-label
| +--ro backup-path* [index]

```

|              |          |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|----------|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |          |  |  | <pre> +--ro index                uint32 +--ro network-ref?         leafref +--ro path-element* [path-element-id]   +--ro path-element-id     uint32   +--ro index?              uint32   +--ro (type)?     +--:(ip-address)       +--ro ip-address-hop         +--ro address?      inet:ip-address         +--ro hop-type?     te-hop-type     +--:(as-number)       +--ro as-number-hop         +--ro as-number?    binary         +--ro hop-type?     te-hop-type     +--:(unnumbered-link)       +--ro unnumbered-hop         +--ro router-id?    inet:ip-address         +--ro interface-id? uint32         +--ro hop-type?     te-hop-type     +--:(label)       +--ro label-hop         +--ro value?        rt-types:generalized- </pre> |
| label        |          |  |  | <pre> +--:(sid)   +--ro sid-hop     +--ro sid?      rt-types:generalized-label +--ro protection-type? uint16 +--ro tunnels   +--ro sharing?     boolean   +--ro tunnel* [tunnel-name]     +--ro tunnel-name string     +--ro sharing?    boolean +--ro admin-status?      te-types:te- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| admin-status |          |  |  | <pre> +--ro link-index?      uint64 +--ro administrative-group? te-types:admin- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| groups       |          |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| capability   | encoding |  |  | <pre> +--ro interface-switching-capability* [switching- +--ro switching-capability identityref +--ro encoding              identityref +--ro max-lsp-bandwidth* [priority]   +--ro priority            uint8   +--ro bandwidth?         te-bandwidth +--ro link-protection-type? enumeration +--ro max-link-bandwidth?  te-bandwidth </pre>                                                                                                                                                                                                                                                                                                                                                                                                    |

```

+--ro max-resv-link-bandwidth? te-bandwidth
+--ro unreserved-bandwidth* [priority]
| +--ro priority uint8
| +--ro bandwidth? te-bandwidth
+--ro te-default-metric? uint32
+--ro te-delay-metric? uint32
+--ro te-srlgs
| +--ro value* te-types:srlg
+--ro te-nsrlgs {nsrlg}?
| +--ro id* uint32
+--ro oper-status? te-types:te-oper-status
+--ro is-transitional? empty
+--ro information-source? te-info-source
+--ro information-source-state
| +--ro credibility-preference? uint16
| +--ro logical-network-element? string
| +--ro network-instance? string
| +--ro topology
| | +--ro network-ref? leafref
| | +--ro link-ref? leafref
+--ro information-source-entry* [information-source]
| +--ro information-source te-info-source
| +--ro information-source-state
| | +--ro credibility-preference? uint16
| | +--ro logical-network-element? string
| | +--ro network-instance? string
| | +--ro topology
| | | +--ro network-ref? leafref
| | | +--ro link-ref? leafref
| +--ro link-index? uint64
| +--ro administrative-group? te-types:admin-
groups
capability encoding
| +--ro interface-switching-capability* [switching-
| | +--ro switching-capability identityref
| | +--ro encoding identityref
| | +--ro max-lsp-bandwidth* [priority]
| | | +--ro priority uint8
| | | +--ro bandwidth? te-bandwidth
| +--ro link-protection-type? enumeration
| +--ro max-link-bandwidth? te-bandwidth
| +--ro max-resv-link-bandwidth? te-bandwidth
| +--ro unreserved-bandwidth* [priority]
| | +--ro priority uint8
| | +--ro bandwidth? te-bandwidth
| +--ro te-default-metric? uint32

```

```

| | | |--ro te-delay-metric? uint32
| | | |--ro te-srlgs
| | | | |--ro value* te-types:srlg
| | | |--ro te-nsrlgs {nsrlg}?
| | | | |--ro id* uint32
|--ro recovery
| | |--ro restoration-status? te-types:te-recovery-status
| | |--ro protection-status? te-types:te-recovery-status
+--ro underlay {te-topology-hierarchy}?
| | |--ro dynamic? boolean
| | |--ro committed? boolean
+--ro statistics
| | |--ro discontinuity-time yang:date-and-time
| | |--ro disables? yang:counter32
| | |--ro enables? yang:counter32
| | |--ro maintenance-clears? yang:counter32
| | |--ro maintenance-sets? yang:counter32
| | |--ro modifies? yang:counter32
| | |--ro downs? yang:counter32
| | |--ro ups? yang:counter32
| | |--ro fault-clears? yang:counter32
| | |--ro fault-detects? yang:counter32
| | |--ro protection-switches? yang:counter32
| | |--ro protection-reverts? yang:counter32
| | |--ro restoration-failures? yang:counter32
| | |--ro restoration-starts? yang:counter32
| | |--ro restoration-successes? yang:counter32
| | |--ro restoration-reversion-failures? yang:counter32
| | |--ro restoration-reversion-starts? yang:counter32
| | |--ro restoration-reversion-successes? yang:counter32
augment /nw:networks/nw:network/nw:node/nt:termination-point:
 |--rw te-tp-id? te-types:te-tp-id
 |--rw te!
 |--rw config
 | | |--rw interface-switching-capability* [switching-capability
encoding]
 | | | |--rw switching-capability identityref
 | | | |--rw encoding identityref
 | | | |--rw max-lsp-bandwidth* [priority]
 | | | | |--rw priority uint8
 | | | | |--rw bandwidth? te-bandwidth
 | | |--rw inter-layer-lock-id? uint32
+--ro state
 |--ro interface-switching-capability* [switching-capability
encoding]
 | |--ro switching-capability identityref

```

```
| +--ro encoding identityref
| +--ro max-lsp-bandwidth* [priority]
| +--ro priority uint8
| +--ro bandwidth? te-bandwidth
+--ro inter-layer-lock-id? uint32
+--ro geolocation
 +--ro altitude? int64
 +--ro latitude? geographic-coordinate-degree
 +--ro longitude? geographic-coordinate-degree
```

## 7. TE Topology Yang Module

```
<CODE BEGINS> file "ietf-te-topology@2017-03-12.yang"
module ietf-te-topology {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology";

 prefix "tet";

 import ietf-yang-types {
 prefix "yang";
 }

 import ietf-inet-types {
 prefix "inet";
 }

 import ietf-te-types {
 prefix "te-types";
 }

 import ietf-network {
 prefix "nw";
 }

 import ietf-network-topology {
 prefix "nt";
 }

 import ietf-routing-types {
 prefix "rt-types";
 }
}
```

```
organization
 "Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";

contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 WG Chair: Lou Berger
 <mailto:lberger@labn.net>

 WG Chair: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editor: Xufeng Liu
 <mailto:Xufeng_Liu@jabil.com>

 Editor: Igor Bryskin
 <mailto:Igor.Bryskin@huawei.com>

 Editor: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editor: Tarek Saad
 <mailto:tsaad@cisco.com>

 Editor: Himanshu Shah
 <mailto:hshah@ciena.com>

 Editor: Oscar Gonzalez De Dios
 <mailto:oscar.gonzalezdedios@telefonica.com>";

description "TE topology model";

revision "2017-03-12" {
 description "Initial revision";
 reference "TBD";
}
```

```

/*
 * Features
 */
feature nsrlg {
 description
 "This feature indicates that the system supports NSRLG
 (Not Sharing Risk Link Group).";
}

feature te-topology-hierarchy {
 description
 "This feature indicates that the system allows underlay
 and/or overlay TE topology hierarchy.";
}

feature template {
 description
 "This feature indicates that the system supports
 template configuration.";
}

/*
 * Typedefs
 */
typedef geographic-coordinate-degree {
 type decimal64 {
 fraction-digits 8;
 }
 description
 "Decimal degree (DD) used to express latitude and longitude
 geographic coordinates.";
} // geographic-coordinate-degree

typedef te-bandwidth {
 type string {
 pattern
 '0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
 + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?))?[pP](\+)?(12[0-7]|'
 + '1[01]\d|0?\d?\d)?|0[xX][\da-fA-F]{1,8}|\d+'
 + '(,(0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'

```



```

+ '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?)?)?[pP](\+)?(12[0-7]|'
+ '1[01]\d|0?\d?\d?)|0[xX][\da-fA-F]{1,8}|\d+))*';
}
description
 "This is the generic bandwidth type that is a string containing
 a list of numbers separated by commas, with each of these
 number can be non-negative decimal, hex integer, or hex float:
 (dec | hex | float)[*('','(dec | hex | float))].
 For packet switching type, a float number is used, such as
 0x1p10.
 For OTN switching type, a list of integers can be used, such
 as '0,2,3,1', indicating 2 odu0's and 1 odu3.
 For DWDM, a list of pairs of slot number and width can be
 used, such as '0, 2, 3, 3', indicating a frequency slot 0 with
 slot width 2 and a frequency slot 3 with slot width 3.";
} // te-bandwidth

typedef te-info-source {
 type enumeration {
 enum "unknown" {
 description "The source is unknown.";
 }
 enum "locally-configured" {
 description "Configured entity.";
 }
 enum "ospfv2" {
 description "OSPFv2.";
 }
 enum "ospfv3" {
 description "OSPFv3.";
 }
 enum "isis" {
 description "ISIS.";
 }
 enum "bgp-ls" {
 description "BGP-LS.";
 reference
 "RFC7752: North-Bound Distribution of Link-State and
 Traffic Engineering (TE) Information Using BGP";
 }
 }
}

```

```
 enum "system-processed" {
 description "System processed entity.";
 }
 enum "other" {
 description "Other source.";
 }
}
description
 "Describing the type of source that has provided the
 related information, and the source credibility.";
} // te-info-source

typedef te-path-disjointness {
 type bits {
 bit node {
 position 0;
 description "Node disjoint.";
 }
 bit link {
 position 1;
 description "Link disjoint.";
 }
 bit srlg {
 position 2;
 description "SRLG (Shared Risk Link Group) disjoint.";
 }
 }
 description
 "Type of the resource disjointness for a TE tunnel path.";
 reference
 "RFC4872: RSVP-TE Extensions in Support of End-to-End
 Generalized Multi-Protocol Label Switching (GMPLS)
 Recovery";
} // te-path-disjointness

/*
 * Groupings
 */
grouping connectivity-label-restriction-list {
 description
```

```
 "List of abel restrictions specifying what labels may or may
 not be used on a link connectivity.";
list label-restriction {
 key "inclusive-exclusive label-start";
 description
 "List of abel restrictions specifying what labels may or may
 not be used on a link connectivity.";
 reference
 "RFC7579: General Network Element Constraint Encoding
 for GMPLS-Controlled Networks";
 leaf inclusive-exclusive {
 type enumeration {
 enum inclusive {
 description "The label or label range is inclusive.";
 }
 enum exclusive {
 description "The label or label range is exclusive.";
 }
 }
 description
 "Whether the list item is inclusive or exclusive.";
 }
 leaf label-start {
 type rt-types:generalized-label;
 description
 "This is the starting lable if a lable range is specified.
 This is the lable value if a single lable is specified,
 in which case, attribute 'label-end' is not set.";
 }
 leaf label-end {
 type rt-types:generalized-label;
 description
 "The ending lable if a lable range is specified;
 This attribute is not set, If a single lable is
 specified.";
 }
 leaf range-bitmap {
 type binary;
 description
 "When there are gaps between label-start and label-end,
```

```
 this attribute is used to specified the possitions
 of the used labels.";
 }
}
} // connectivity-label-restrictions

grouping connectivity-matrix-entry-attributes {
 description
 "Attributes of connectivity matrix entry.";
 leaf is-allowed {
 type boolean;
 description
 "true - switching is allowed,
 false - switching is disallowed.";
 }
 uses connectivity-label-restriction-list;
 container underlay {
 if-feature te-topology-hierarchy;
 presence
 "Indicates the underlay exists for this link.";
 description "Attributes of the te-link underlay.";
 reference
 "RFC4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";

 uses te-link-underlay-attributes;
 } // underlay
 uses te-link-iscd-attributes;
 uses te-link-connectivity-attributes;
} // connectivity-matrix-entry-attributes

grouping geolocation-container {
 description
 "A container containing a GPS location.";
 container geolocation{
 description
 "A container containing a GPS location.";
 leaf altitude {
 type int64;
 }
 }
}
```

```
 units millimeter;
 description
 "Distance above the sea level.";
 }
 leaf latitude {
 type geographic-coordinate-degree {
 range "-90..90";
 }
 description
 "Relative position north or south on the Earth's surface.";
 }
 leaf longitude {
 type geographic-coordinate-degree {
 range "-180..180";
 }
 description
 "Angular distance east or west on the Earth's surface.";
 }
} // gps-location
} // geolocation-container

grouping information-source-state-attributes {
 description
 "The attributes identifying source that has provided the
 related information, and the source credibility.";
 leaf credibility-preference {
 type uint16;
 description
 "The preference value to calculate the traffic
 engineering database credibility value used for
 tie-break selection between different
 information-source values.
 Higher value is more preferable.";
 }
 leaf logical-network-element {
 type string;
 description
 "When applicable, this is the name of a logical network
 element from which the information is learned.";
 } // logical-network-element
}
```

```
leaf network-instance {
 type string;
 description
 "When applicable, this is the name of a network-instance
 from which the information is learned.";
} // network-instance
} // information-source-state-attributes

grouping information-source-per-link-attributes {
 description
 "Per node container of the attributes identifying source that
 has provided the related information, and the source
 credibility.";
 leaf information-source {
 type te-info-source;
 description
 "Indicates the source of the information.";
 }
 container information-source-state {
 description
 "The container contains state attributes related to
 the information source.";
 uses information-source-state-attributes;
 container topology {
 description
 "When the information is processed by the system,
 the attributes in this container indicate which topology
 is used to process to generate the result information.";
 uses te-topology-ref;
 leaf link-ref {
 type leafref {
 path "/nw:networks/nw:network[nw:network-id = "
 + "current()/../network-ref]/nt:link/nt:link-id";
 require-instance false;
 }
 description
 "A reference to a link-id.";
 }
 } // topology
 } // information-source-state
}
```

```
 } // information-source-per-link-attributes

 grouping information-source-per-node-attributes {
 description
 "Per node container of the attributes identifying source that
 has provided the related information, and the source
 credibility.";
 leaf information-source {
 type te-info-source;
 description
 "Indicates the source of the information.";
 }
 container information-source-state {
 description
 "The container contains state attributes related to
 the information source.";
 uses information-source-state-attributes;
 container topology {
 description
 "When the information is processed by the system,
 the attributes in this container indicate which topology
 is used to process to generate the result information.";
 uses te-topology-ref;
 leaf node-ref {
 type leafref {
 path "/nw:networks/nw:network[nw:network-id = "
 + "current()/../network-ref]/nw:node/nw:node-id";
 require-instance false;
 }
 description
 "A reference to a node-id.";
 }
 } // topology
 } // information-source-state
 } // information-source-per-node-attributes

 grouping interface-switching-capability-list {
 description
 "List of Interface Switching Capabilities Descriptors (ISCD)";
```

```
list interface-switching-capability {
 key "switching-capability encoding";
 description
 "List of Interface Switching Capabilities Descriptors (ISCD)
 for this link.";
 reference
 "RFC3471: Generalized Multi-Protocol Label Switching (GMPLS)
 Signaling Functional Description.
 RFC4203: OSPF Extensions in Support of Generalized
 Multi-Protocol Label Switching (GMPLS).";
 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for this interface.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by this interface.";
 }
 uses te-link-iscd-attributes;
} // interface-switching-capability
} // interface-switching-capability-list

grouping statistics-per-link {
 description
 "Statistics attributes per TE link.";
 leaf discontinuity-time {
 type yang:date-and-time;
 mandatory true;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
```



```
 management subsystem re-initialized itself.";
 }
 /* Administrative attributes */
 leaf disables {
 type yang:counter32;
 description
 "Number of times that link was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that link was enabled.";
 }
 leaf maintenance-clears {
 type yang:counter32;
 description
 "Number of times that link was put out of maintenance.";
 }
 leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that link was put in maintenance.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that link was modified.";
 }
 /* Operational attributes */
 leaf downs {
 type yang:counter32;
 description
 "Number of times that link was set to operational down.";
 }
 leaf ups {
 type yang:counter32;
 description
 "Number of times that link was set to operational up.";
 }
 /* Recovery attributes */
```

```
leaf fault-clears {
 type yang:counter32;
 description
 "Number of times that link experienced fault clear event.";
}
leaf fault-detects {
 type yang:counter32;
 description
 "Number of times that link experienced fault detection.";
}
leaf protection-switches {
 type yang:counter32;
 description
 "Number of times that link experienced protection
 switchover.";
}
leaf protection-reverts {
 type yang:counter32;
 description
 "Number of times that link experienced protection
 reversion.";
}
leaf restoration-failures {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 failure.";
}
leaf restoration-starts {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 start.";
}
leaf restoration-successes {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 success.";
}
```

```
 leaf restoration-reversion-failures {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 failure.";
 }
 leaf restoration-reversion-starts {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 start.";
 }
 leaf restoration-reversion-successes {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 success.";
 }
 } // statistics-per-link

grouping statistics-per-node {
 description
 "Statistics attributes per TE node.";
 leaf discontinuity-time {
 type yang:date-and-time;
 mandatory true;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
 management subsystem re-initialized itself.";
 }
 container node {
 description
 "Containing TE node level statistics attributes.";
 leaf disables {
 type yang:counter32;
 description
```

```
 "Number of times that node was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that node was enabled.";
 }
 leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that node was put in maintenance.";
 }
 leaf maintenance-clears {
 type yang:counter32;
 description
 "Number of times that node was put out of maintenance.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that node was modified.";
 }
} // node
container connectivity-matrix-entry {
 description
 "Containing connectivity matrix entry level statistics
 attributes.";
 leaf creates {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 created.";
 reference
 "RFC6241. Section 7.2 for 'create' operation. ";
 }
 leaf deletes {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 deleted.";
```

```
 reference
 "RFC6241. Section 7.2 for 'delete' operation. ";
 }
 leaf disables {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 enabled.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 modified.";
 }
 } // connectivity-matrix-entry
 } // statistics-per-node

grouping statistics-per-ttp {
 description
 "Statistics attributes per TE TTP (Tunnel Termination Point).";
 leaf discontinuity-time {
 type yang:date-and-time;
 mandatory true;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
 management subsystem re-initialized itself.";
 }
}
container tunnel-termination-point {
 description
```

```
 "Containing TE TTP (Tunnel Termination Point) level
 statistics attributes.";
/* Administrative attributes */
leaf disables {
 type yang:counter32;
 description
 "Number of times that TTP was disabled.";
}
leaf enables {
 type yang:counter32;
 description
 "Number of times that TTP was enabled.";
}
leaf maintenance-clears {
 type yang:counter32;
 description
 "Number of times that TTP was put out of maintenance.";
}
leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that TTP was put in maintenance.";
}
leaf modifies {
 type yang:counter32;
 description
 "Number of times that TTP was modified.";
}
/* Operational attributes */
leaf downs {
 type yang:counter32;
 description
 "Number of times that TTP was set to operational down.";
}
leaf ups {
 type yang:counter32;
 description
 "Number of times that TTP was set to operational up.";
}
leaf in-service-clears {
```

```
 type yang:counter32;
 description
 "Number of times that TTP was taken out of service
 (TE tunnel was released).";
 }
 leaf in-service-sets {
 type yang:counter32;
 description
 "Number of times that TTP was put in service by a TE
 tunnel (TE tunnel was set up).";
 }
} // tunnel-termination-point

container local-link-connectivity {
 description
 "Containing TE LLCL (Local Link Connectivity List) level
 statistics attributes.";
 leaf creates {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was created.";
 reference
 "RFC6241. Section 7.2 for 'create' operation. ";
 }
 leaf deletes {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was deleted.";
 reference
 "RFC6241. Section 7.2 for 'delete' operation.";
 }
 leaf disables {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was enabled.";
```

```
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was modified.";
 }
 } // local-link-connectivity
} // statistics-per-ttp

grouping te-link-augment {
 description
 "Augmentation for TE link.";

 container te {
 must "count(..nt:supporting-link)<=1" {
 description
 "For a link in a TE topology, there cannot be more
 than 1 supporting link. If one or more link paths are
 abstracted, the underlay is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";

 container config {
 description
 "Configuration data.";
 uses te-link-config;
 } // config
 container state {
 config false;
 description
 "Operational state data.";
 uses te-link-config;
 uses te-link-state-derived;
 } // state
 container statistics {
 config false;
 description
 "Statistics data.";
```



```
 uses statistics-per-link;
 } // statistics
} // te
} // te-link-augment

grouping te-link-config {
 description
 "TE link configuration grouping.";
 choice bundle-stack-level {
 description
 "The TE link can be partitioned into bundled
 links, or component links.";
 case bundle {
 container bundled-links {
 description
 "A set of bundled links.";
 reference
 "RFC4201: Link Bundling in MPLS Traffic Engineering
 (TE).";
 list bundled-link {
 key "sequence";
 description
 "Specify a bundled interface that is
 further partitioned.";
 leaf sequence {
 type uint32;
 description
 "Identify the sequence in the bundle.";
 }
 leaf src-tp-ref {
 type leafref {
 path "../../../nw:node[nw:node-id = "
 + "current()/../../../nt:source/"
 + "nt:source-node]/"
 + "nt:termination-point/nt:tp-id";
 require-instance true;
 }
 }
 description
 "Reference to another TE termination point on the
 same source node.";
 }
 }
 }
 }
}
```

```
 }
 leaf des-tp-ref {
 type leafref {
 path "../../../nw:node[nw:node-id = "
 + "current()/../../../nt:destination/"
 + "nt:dest-node]/"
 + "nt:termination-point/nt:tp-id";
 require-instance true;
 }
 description
 "Reference to another TE termination point on the
 same destination node.";
 }
 } // list bundled-link
}
case component {
 container component-links {
 description
 "A set of component links";
 list component-link {
 key "sequence";
 description
 "Specify a component interface that is
 sufficient to unambiguously identify the
 appropriate resources";

 leaf sequence {
 type uint32;
 description
 "Identify the sequence in the bundle.";
 }
 leaf src-interface-ref {
 type string;
 description
 "Reference to component link interface on the
 source node.";
 }
 leaf des-interface-ref {
 type string;
```

```
 description
 "Reference to component link interface on the
 destination node.";
 }
}
}
} // bundle-stack-level

leaf-list te-link-template {
 if-feature template;
 type leafref {
 path "../../../../../te/templates/link-template/name";
 }
 description
 "The reference to a TE link template.";
}
uses te-link-config-attributes;
} // te-link-config

grouping te-link-config-attributes {
 description
 "Link configuration attributes in a TE topology.";
 container te-link-attributes {
 description "Link attributes in a TE topology.";
 leaf access-type {
 type te-types:te-link-access-type;
 description
 "Link access type, which can be point-to-point or
 multi-access.";
 }
 }
 container external-domain {
 description
 "For an inter-domain link, specify the attributes of
 the remote end of link, to facilitate the signalling at
 local end.";
 uses te-topology-ref;
 leaf remote-te-node-id {
 type te-types:te-node-id;
 description
```

```
 "Remote TE node identifier, used together with
 remote-te-link-id to identify the remote link
 termination point in a different domain.";
 }
 leaf remote-te-link-tp-id {
 type te-types:te-tp-id;
 description
 "Remote TE link termination point identifier, used
 together with remote-te-node-id to identify the remote
 link termination point in a different domain.";
 }
 leaf plug-id {
 type uint32;
 description
 "A topology-wide unique number that identifies on the
 network a connectivity supporting a given inter-domain
 TE link. This is more flexible alternative to specifying
 remote-te-node-id and remote-te-link-tp-id, when the
 provider does not know remote-te-node-id and
 remote-te-link-tp-id or need to give client the
 flexibility to mix-n-match multiple topologies.";
 }
}
leaf is-abstract {
 type empty;
 description "Present if the link is abstract.";
}
leaf name {
 type string;
 description "Link Name.";
}
container underlay {
 if-feature te-topology-hierarchy;
 presence
 "Indicates the underlay exists for this link.";
 description "Attributes of the te-link underlay.";
 reference
 "RFC4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";
```

```
 uses te-link-underlay-attributes;
 } // underlay
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
 }

 uses te-link-info-attributes;
} // te-link-attributes
} // te-link-config-attributes

grouping te-link-connectivity-attributes {
 description
 "Advertised TE connectivity attributes.";
 leaf max-link-bandwidth {
 type te-bandwidth;
 description
 "Maximum bandwidth that can be seen on this link in this
 direction. Units in bytes per second.";
 reference
 "RFC3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC5305: IS-IS Extensions for Traffic Engineering.";
 }
 leaf max-resv-link-bandwidth {
 type te-bandwidth;
 description
 "Maximum amount of bandwidth that can be reserved in this
 direction in this link. Units in bytes per second.";
 reference
 "RFC3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC5305: IS-IS Extensions for Traffic Engineering.";
 }
 list unreserved-bandwidth {
 key "priority";
 max-elements "8";
 description
```

```
 "Unreserved bandwidth for 0-7 priority levels. Units in
 bytes per second.";
 reference
 "RFC3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC5305: IS-IS Extensions for Traffic Engineering.";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "Priority.";
 }
 leaf bandwidth {
 type te-bandwidth;
 description
 "Unreserved bandwidth for this level.";
 }
}
leaf te-default-metric {
 type uint32;
 description
 "Traffic engineering metric.";
}
leaf te-delay-metric {
 type uint32;
 description
 "Traffic engineering delay metric.";
}
container te-srlgs {
 description
 "Containing a list of SLRGs.";
 leaf-list value {
 type te-types:srlg;
 description "SRLG value.";
 reference
 "RFC4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
 }
}
container te-nsrlgs {
```

```
 if-feature nsrlg;
 description
 "Containing a list of NSRLGs (Not Sharing Risk Link
 Groups).
 When an abstract TE link is configured, this list specifies
 the request that underlay TE paths need to be mutually
 disjoint with other TE links in the same groups.";
 leaf-list id {
 type uint32;
 description
 "NSRLG ID, uniquely configured within a topology.";
 reference
 "RFC4872: RSVP-TE Extensions in Support of End-to-End
 Generalized Multi-Protocol Label Switching (GMPLS)
 Recovery";
 }
 }
} // te-link-connectivity-attributes

grouping te-link-info-attributes {
 description
 "Advertised TE information attributes.";
 leaf link-index {
 type uint64;
 description
 "The link identifier. If OSPF is used, this represents an
 ospfLsdbID. If IS-IS is used, this represents an isisLSPID.
 If a locally configured link is used, this object represents
 a unique value, which is locally defined in a router.";
 }
 leaf administrative-group {
 type te-types:admin-groups;
 description
 "Administrative group or color of the link.
 This attribute covers both administrative group (defined in
 RFC3630, RFC5329, and RFC5305), and extended administrative
 group (defined in RFC7308).";
 }
 uses interface-switching-capability-list;
 leaf link-protection-type {
```

```
type enumeration {
 enum "unprotected" {
 description "Unprotected.";
 }
 enum "extra-traffic" {
 description "Extra traffic.";
 }
 enum "shared" {
 description "Shared.";
 }
 enum "1-for-1" {
 description "One for one protection.";
 }
 enum "1-plus-1" {
 description "One plus one protection.";
 }
 enum "enhanced" {
 description "Enhanced protection.";
 }
}
description
 "Link Protection Type desired for this link.";
reference
 "RFC4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
}
uses te-link-connectivity-attributes;
} // te-link-info-attributes

grouping te-link-iscd-attributes {
 description
 "TE link ISCD (Interface Switching Capability Descriptor)
 attributes.";
 reference
 "Sec 1.4, RFC4203: OSPF Extensions in Support of Generalized
 Multi-Protocol Label Switching (GMPLS). Section 1.4.";
 list max-lsp-bandwidth {
 key "priority";
 max-elements "8";
 description
```



```
 "Maximum LSP Bandwidth at priorities 0-7.";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "Priority.";
 }
 leaf bandwidth {
 type te-bandwidth;
 description
 "Max LSP Bandwidth for this level";
 }
}
} // te-link-iscd-attributes

grouping te-link-state-derived {
 description
 "Link state attributes in a TE topology.";
 leaf oper-status {
 type te-types:te-oper-status;
 description
 "The current operational state of the link.";
 }
 leaf is-transitional {
 type empty;
 description
 "Present if the link is transitional, used as an
 alternative approach in lieu of inter-layer-lock-id
 for path computation in a TE topology covering multiple
 layers or multiple regions.";
 reference
 "RFC5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
 }
 uses information-source-per-link-attributes;
 list information-source-entry {
 key "information-source";
 description
```

```
 "A list of information sources learned, including the one
 used.";
 uses information-source-per-link-attributes;
 uses te-link-info-attributes;
}
container recovery {
 description
 "Status of the recovery process.";
 leaf restoration-status {
 type te-types:te-recovery-status;
 description
 "Restoration status.";
 }
 leaf protection-status {
 type te-types:te-recovery-status;
 description
 "Protection status.";
 }
}
container underlay {
 if-feature te-topology-hierarchy;
 description "State attributes for te-link underlay.";
 uses te-link-state-underlay-attributes;
}
} // te-link-state-derived

grouping te-link-state-underlay-attributes {
 description "State attributes for te-link underlay.";
 leaf dynamic {
 type boolean;
 description
 "true if the underlay is dynamically created.";
 }
 leaf committed {
 type boolean;
 description
 "true if the underlay is committed.";
 }
}
} // te-link-state-underlay-attributes
```

```
grouping te-link-underlay-attributes {
 description "Attributes for te-link underlay.";
 reference
 "RFC4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";
 container primary-path {
 description
 "The service path on the underlay topology that
 supports this link.";
 uses te-topology-ref;
 list path-element {
 key "path-element-id";
 description
 "A list of path elements describing the service path.";
 leaf path-element-id {
 type uint32;
 description "To identify the element in a path.";
 }
 uses te-path-element;
 }
 } // primary-path
 list backup-path {
 key "index";
 description
 "A list of backup service paths on the underlay topology that
 protect the underlay primary path. If the primary path is
 not protected, the list contains zero elements. If the
 primary path is protected, the list contains one or more
 elements.";
 leaf index {
 type uint32;
 description
 "A sequence number to identify a backup path.";
 }
 uses te-topology-ref;
 list path-element {
 key "path-element-id";
 description
 "A list of path elements describing the backup service
```

```
 path";
 leaf path-element-id {
 type uint32;
 description "To identify the element in a path.";
 }
 uses te-path-element;
 }
} // underlay-backup-path
leaf protection-type {
 type uint16;
 description
 "Underlay protection type desired for this link";
}
container tunnels {
 description
 "Underlay TE tunnels supporting this TE link.";
 leaf sharing {
 type boolean;
 default true;
 description
 "'true' if the underlay tunnel can be shared with other
 TE links;
 'false' if the underlay tunnel is dedicated to this
 TE link.
 This leaf is the default option for all TE tunnels,
 and may be overridden by the per TE tunnel value.";
 }
}
list tunnel {
 key "tunnel-name";
 description
 "Zero, one or more underlay TE tunnels that support this TE
 link.";
 leaf tunnel-name {
 type string;
 description
 "A tunnel name uniquely identifies an underlay TE tunnel,
 used together with the source-node of this link.
 The detailed information of this tunnel can be retrieved
 from the ietf-te model.";
 reference "RFC3209";
 }
}
```

```
 }
 leaf sharing {
 type boolean;
 description
 "'true' if the underlay tunnel can be shared with other
 TE links;
 'false' if the underlay tunnel is dedicated to this
 TE link.";
 }
 } // tunnel
} // tunnels
} // te-link-underlay-attributes

grouping te-node-augment {
 description
 "Augmentation for TE node.";

 leaf te-node-id {
 type te-types:te-node-id;
 description
 "The identifier of a node in the TE topology.
 A node is specific to a topology to which it belongs.";
 }

 container te {
 must "../te-node-id" {
 description
 "te-node-id is mandatory.";
 }
 must "count(..nw:supporting-node)<=1" {
 description
 "For a node in a TE topology, there cannot be more
 than 1 supporting node. If multiple nodes are abstracted,
 the underlay-topology is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";

 container config {
```

```
 description
 "Configuration data.";
 uses te-node-config;
 } // config
 container state {
 config false;
 description
 "Operational state data.";

 uses te-node-config;
 uses te-node-state-derived;
 } // state
 container statistics {
 config false;
 description
 "Statistics data.";
 uses statistics-per-node;
 } // statistics

 list tunnel-termination-point {
 key "tunnel-tp-id";
 description
 "A termination point can terminate a tunnel.";
 leaf tunnel-tp-id {
 type binary;
 description
 "Tunnel termination point identifier.";
 }

 container config {
 description
 "Configuration data.";
 uses te-node-tunnel-termination-attributes;
 }
 container state {
 config false;
 description
 "Operational state data.";

 uses te-node-tunnel-termination-attributes;
```

```
 uses geolocation-container;
 } // state
container statistics {
 config false;
 description
 "Statistics data.";
 uses statistics-per-ttp;
} // statistics

// Relations to other tunnel termination points
list supporting-tunnel-termination-point {
 key "node-ref tunnel-tp-ref";
 description
 "Identifies the tunnel termination points, that this
 tunnel termination point is depending on.";
 leaf node-ref {
 type inet:uri;
 /* The followings are the intended valications
 * but some Yang validation tools fail on them.
 type union {
 type leafref {
 path "../.../.../nw:supporting-node/nw:node-ref";
 require-instance false;
 }
 type leafref {
 path "/nw:networks/nw:network"+
 "[nw:network-id="+
 "current()/.../.../.../te/config/"+
 "te-node-attributes/underlay-topology/"+
 "network-ref]/nw:node/nw:node-id";
 require-instance false;
 }
 type leafref {
 path "/nw:networks/nw:network"+
 "[nw:network-id="+
 "current()/.../.../.../te/state/"+
 "te-node-attributes/underlay-topology/"+
 "network-ref]/nw:node/nw:node-id";
 require-instance false;
 }
 }
 }
}
```

```
 }
 *****/
 description
 "This leaf identifies in which node the supporting
 tunnel termination point is present.";
 }
 leaf tunnel-tp-ref {
 type binary;
 /* The followings are the intended valications
 * but some Yang validation tools fail on them.
 type union {
 type leafref {
 path "/nw:networks/nw:network"+
 "[nw:network-id="+
 "current()/../../../../nw:supporting-node/"+
 "nw:network-ref]/"+
 "nw:node[nw:node-id=current()/../node-ref]/te/"+
 "tunnel-termination-point/tunnel-tp-id";
 require-instance false;
 }
 type leafref {
 path "/nw:networks/nw:network"+
 "[nw:network-id="+
 "current()/../../../../te/config/"+
 "te-node-attributes/underlay-topology/"+
 "network-ref]/"+
 "nw:node[nw:node-id=current()/../node-ref]/te/"+
 "tunnel-termination-point/tunnel-tp-id";
 require-instance false;
 }
 type leafref {
 path "/nw:networks/nw:network"+
 "[nw:network-id="+
 "current()/../../../../te/state/"+
 "te-node-attributes/underlay-topology/"+
 "network-ref]/"+
 "nw:node[nw:node-id=current()/../node-ref]/te/"+
 "tunnel-termination-point/tunnel-tp-id";
 require-instance false;
 }
 }
 }
```



```
 }
 *****/
 description
 "Reference to a tunnel termination point, which is
 either in the supporting node or a node in an
 underlay topology.";
 }
} // supporting-tunnel-termination-point
} // tunnel-termination-point
} // te
} // te-node-augment

grouping te-node-config {
 description "TE node configuration grouping.";

 leaf-list te-node-template {
 if-feature template;
 type leafref {
 path "../..../te/templates/node-template/name";
 }
 description
 "The reference to a TE node template.";
 }
 uses te-node-config-attributes;
} // te-node-config

grouping te-node-config-attributes {
 description "Configuration node attributes in a TE topology.";
 container te-node-attributes {
 description "Containing node attributes in a TE topology.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
 }
 uses te-node-connectivity-matrix;
 uses te-node-info-attributes;
 } // te-node-attributes
} // te-node-config-attributes
```

```
grouping te-node-config-attributes-template {
 description
 "Configuration node attributes for template in a TE topology.";
 container te-node-attributes {
 description "Containing node attributes in a TE topology.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
 }
 uses te-node-info-attributes;
 } // te-node-attributes
} // te-node-config-attributes-template

grouping te-node-connectivity-matrix {
 description "Connectivity matrix on a TE node.";
 container connectivity-matrices {
 description
 "Containing connectivity matrix on a TE node.";
 leaf number-of-entries {
 type uint16;
 description
 "The number of connectivity matrix entries.
 If this number is specified in the configuration request,
 the number is requested number of entries, which may not
 all be listed in the list;
 if this number is reported in the state data,
 the number is the current number of operational entries.";
 }
 uses connectivity-matrix-entry-attributes;
 list connectivity-matrix {
 key "id";
 description
 "Represents node's switching limitations, i.e. limitations
 in interconnecting network TE links across the node.";
 reference
 "RFC7579: General Network Element Constraint Encoding
 for GMPLS-Controlled Networks.";
 leaf id {
 type uint32;
 }
 }
 }
}
```

```
 description "Identifies the connectivity-matrix entry.";
 }
 container from {
 leaf tp-ref {
 type leafref {
 path "../.../.../.../.../nt:termination-point/"+
 "nt:tp-id";
 }
 description
 "Relative reference to source termination point.";
 }
 description
 "Reference to source NTP.";
 }
 container to {
 leaf tp-ref {
 type leafref {
 path "../.../.../.../.../nt:termination-point/"+
 "nt:tp-id";
 }
 description
 "Relative reference to destination termination point.";
 }
 description
 "Reference to destination NTP.";
 }
 uses connectivity-matrix-entry-attributes;
} // connectivity-matrix
} // connectivity-matrices
} // te-node-connectivity-matrix

grouping te-node-connectivity-matrix-abs {
 description
 "Connectivity matrix on a TE node, using absolute
 paths to reference termination points.";
 list connectivity-matrix {
 key "id";
 description
 "Represents node's switching limitations, i.e. limitations
 in interconnecting network TE links across the node.";
 }
}
```

```
reference
 "RFC7579: General Network Element Constraint Encoding
 for GMPLS-Controlled Networks.";
leaf id {
 type uint32;
 description "Identifies the connectivity-matrix entry.";
}
container from {
 uses nt:tp-ref;
 description
 "Reference to source NTP.";
}
container to {
 uses nt:tp-ref;
 description
 "Reference to destination NTP.";
}
leaf is-allowed {
 type boolean;
 description
 "true - switching is allowed,
 false - switching is disallowed.";
}
}
} // te-node-connectivity-matrix-abs

grouping te-node-info-attributes {
 description
 "Advertised TE information attributes.";
 leaf domain-id {
 type uint32;
 description
 "Identifies the domain that this node belongs.
 This attribute is used to support inter-domain links.";
 reference
 "RFC5152: A Per-Domain Path Computation Method for
 Establishing Inter-Domain Traffic Engineering (TE)
 Label Switched Paths (LSPs).
 RFC5392: OSPF Extensions in Support of Inter-Autonomous
 System (AS) MPLS and GMPLS Traffic Engineering.
```

```
 RFC5316: ISIS Extensions in Support of Inter-Autonomous
 System (AS) MPLS and GMPLS Traffic Engineering.";
 }
 leaf is-abstract {
 type empty;
 description
 "Present if the node is abstract, not present if the node
 is actual.";
 }
 leaf name {
 type inet:domain-name;
 description "Node name.";
 }
 leaf-list signaling-address {
 type inet:ip-address;
 description "Node signaling address.";
 }
 container underlay-topology {
 if-feature te-topology-hierarchy;
 description
 "When an abstract node encapsulates a topology,
 the attributes in this container point to said topology.";
 uses te-topology-ref;
 }
} // te-node-info-attributes

grouping te-node-state-derived {
 description "Node state attributes in a TE topology.";
 leaf oper-status {
 type te-types:te-oper-status;
 description
 "The current operational state of the node.";
 }
 uses geolocation-container;
 leaf is-multi-access-dr {
 type empty;
 description
 "The presence of this attribute indicates that this TE node
 is a pseudonode elected as a designated router.";
 reference

```

```
 "RFC3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC1195: Use of OSI IS-IS for Routing in TCP/IP and Dual
 Environments.";
 }
 uses information-source-per-node-attributes;
 list information-source-entry {
 key "information-source";
 description
 "A list of information sources learned, including the one
 used.";
 uses information-source-per-node-attributes;
 uses te-node-connectivity-matrix;
 uses te-node-info-attributes;
 }
} // te-node-state-derived

grouping te-node-state-derived-notification {
 description "Node state attributes in a TE topology.";
 leaf oper-status {
 type te-types:te-oper-status;
 description
 "The current operational state of the node.";
 }
 leaf is-multi-access-dr {
 type empty;
 description
 "The presence of this attribute indicates that this TE node
 is a pseudonode elected as a designated router.";
 reference
 "RFC3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC1195: Use of OSI IS-IS for Routing in TCP/IP and Dual
 Environments.";
 }
 uses information-source-per-node-attributes;
 list information-source-entry {
 key "information-source";
 description
 "A list of information sources learned, including the one
```

```
 used.";
 uses information-source-per-node-attributes;
 uses te-node-connectivity-matrix-abs;
 uses te-node-info-attributes;
 }
} // te-node-state-derived-notification

grouping te-node-tunnel-termination-attributes {
 description
 "Termination capability of a tunnel termination point on a
 TE node.";

 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for this interface.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by this interface.";
 }
 leaf inter-layer-lock-id {
 type uint32;
 description
 "Inter layer lock ID, used for path computation in a TE
 topology covering multiple layers or multiple regions.";
 reference
 "RFC5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
 }
 leaf protection-type {
 type identityref {
 base te-types:lsp-prot-type;
 }
 }
}
```

```
 }
 description
 "The protection type that this tunnel termination point
 is capable of.";
 }

 container client-layer-adaptation {
 description
 "Containing capability information to support a client layer
 adaption in multi-layer topology.";
 list switching-capability {
 key "switching-capability encoding";
 description
 "List of supported switching capabilities";
 reference
 "RFC6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).
 RFC4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for the client layer adaption.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by the client layer adaption.";
 }
 leaf bandwidth {
 type te-bandwidth;
 description
 "Bandwidth available for the client layer adaption.";
 }
 }
 }
}
```



```
container local-link-connectivities {
 description
 "Containing local link connectivity list for
 a tunnel termination point on a TE node.";
 leaf number-of-entries {
 type uint16;
 description
 "The number of local link connectivity list entries.
 If this number is specified in the configuration request,
 the number is requested number of entries, which may not
 all be listed in the list;
 if this number is reported in the state data,
 the number is the current number of operational entries.";
 }
 uses connectivity-matrix-entry-attributes;
 list local-link-connectivity {
 key "link-tp-ref";
 description
 "The termination capabilities between
 tunnel-termination-point and link termination-point.
 The capability information can be used to compute
 the tunnel path.
 The Interface Adjustment Capability Descriptors (IACD)
 [RFC6001] on each link-tp can be derived from this
 local-link-connectivity list.";
 reference
 "RFC6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
 leaf link-tp-ref {
 type leafref {
 path "../..//../..//../..//nt:termination-point/nt:tp-id";
 }
 description
 "Link termination point.";
 }

 uses connectivity-matrix-entry-attributes;
 } // local-link-connectivity
} // local-link-connectivities
```

```
 } // te-node-tunnel-termination-attributes

 grouping te-path-element {
 description
 "A group of attributes defining an element in a TE path
 such as TE node, TE link, TE atomic resource or label.";
 uses te-types:explicit-route-hop_config;
 } // te-path-element

 grouping te-termination-point-augment {
 description
 "Augmentation for TE termination point.";

 leaf te-tp-id {
 type te-types:te-tp-id;
 description
 "An identifier to uniquely identify a TE termination
 point.";
 }

 container te {
 must "../te-tp-id";
 presence "TE support.";
 description
 "Indicates TE support.";

 container config {
 description
 "Configuration data.";
 uses te-termination-point-config;
 } // config
 container state {
 config false;
 description
 "Operational state data.";
 uses te-termination-point-config;
 uses geolocation-container;
 } // state
 } // te
 } // te-termination-point-augment
```

```
grouping te-termination-point-config {
 description
 "TE termination point configuration grouping.";
 uses interface-switching-capability-list;
 leaf inter-layer-lock-id {
 type uint32;
 description
 "Inter layer lock ID, used for path computation in a TE
 topology covering multiple layers or multiple regions.";
 reference
 "RFC5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
 }
} // te-termination-point-config

grouping te-topologies-augment {
 description
 "Augmentation for TE topologies.";

 container te {
 presence "TE support.";
 description
 "Indicates TE support.";

 container templates {
 description
 "Configuration parameters for templates used for TE
 topology.";

 list node-template {
 if-feature template;
 key "name";
 leaf name {
 type te-types:te-template-name;
 description
 "The name to identify a TE node template.";
 }
 }
 }
 }
}
```

```
 description
 "The list of TE node templates used to define sharable
 and reusable TE node attributes.";
 uses template-attributes;
 uses te-node-config-attributes-template;
 } // node-template

 list link-template {
 if-feature template;
 key "name";
 leaf name {
 type te-types:te-template-name;
 description
 "The name to identify a TE link template.";
 }
 description
 "The list of TE link templates used to define sharable
 and reusable TE link attributes.";
 uses template-attributes;
 uses te-link-config-attributes;
 } // link-template
} // templates
} // te
} // te-topologies-augment

grouping te-topology-augment {
 description
 "Augmentation for TE topology.";

 leaf provider-id {
 type te-types:te-global-id;
 description
 "An identifier to uniquely identify a provider.";
 }
 leaf client-id {
 type te-types:te-global-id;
 description
 "An identifier to uniquely identify a client.";
 }
 leaf te-topology-id {
```

```
type te-types:te-topology-id;
description
 "It is presumed that a datastore will contain many
 topologies. To distinguish between topologies it is
 vital to have UNIQUE topology identifiers.";
}

container te {
 must "../provider-id and ../client-id and ../te-topology-id";
 presence "TE support.";
 description
 "Indicates TE support.";

 container config {
 description
 "Configuration data.";
 uses te-topology-config;
 } // config
 container state {
 config false;
 description
 "Operational state data.";
 uses te-topology-config;
 uses geolocation-container;
 } // state
} // te
} // te-topology-augment

grouping te-topology-config {
 description
 "TE topology configuration grouping.";
 leaf preference {
 type uint8 {
 range "1..255";
 }
 description
 "Specifies a preference for this topology. A lower number
 indicates a higher preference.";
 }
 leaf optimization-criterion {
```

```
 type identityref {
 base te-types:te-optimization-criterion;
 }
 description
 "Optimization criterion applied to this topology.";
 reference
 "RFC3272: Overview and Principles of Internet Traffic
 Engineering.";
 }
 list nsrlg {
 if-feature nsrlg;
 key "id";
 description
 "List of NSRLGs (Not Sharing Risk Link Groups).";
 reference
 "RFC4872: RSVP-TE Extensions in Support of End-to-End
 Generalized Multi-Protocol Label Switching (GMPLS)
 Recovery";
 leaf id {
 type uint32;
 description
 "Identify the NSRLG entry.";
 }
 leaf disjointness {
 type te-path-disjointness;
 description
 "The type of resource disjointness.";
 }
 } // nsrlg
} // te-topology-config

grouping te-topology-ref {
 description
 "References a TE topology.";
 leaf network-ref {
 type leafref {
 path "/nw:networks/nw:network/nw:network-id";
 require-instance false;
 }
 description

```

```
 "A reference to a network-id in base ietf-network module.";
 }
} // te-topology-ref

grouping te-topology-type {
 description
 "Identifies the TE topology type.";
 container te-topology {
 presence "Indicates TE topology.";
 description
 "Its presence identifies the TE topology type.";
 }
} // te-topology-type

grouping template-attributes {
 description
 "Common attributes for all templates.";

 leaf priority {
 type uint16;
 description
 "The preference value to resolve conflicts between different
 templates. When two or more templates specify values for
 one configuration attribute, the value from the template
 with the highest priority is used.";
 }
 leaf reference-change-policy {
 type enumeration {
 enum no-action {
 description
 "When an attribute changes in this template, the
 configuration node referring to this template does
 not take any action.";
 }
 enum not-allowed {
 description
 "When any configuration object has a reference to this
 template, changing this template is not allowed.";
 }
 enum cascade {
```

```
 description
 "When an attribute changes in this template, the
 configuration object referring to this template applies
 the new attribute value to the corresponding
 configuration.";
 }
}
description
 "This attribute specifies the action taken to a configuration
 node that has a reference to this template.";
}
} // template-attributes

/*
 * Configuration data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
 description
 "Introduce new network type for TE topology.";
 uses te-topology-type;
}

augment "/nw:networks" {
 description
 "Augmentation parameters for TE topologies.";
 uses te-topologies-augment;
}

augment "/nw:networks/nw:network" {
 when "nw:network-types/te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE topology.";
 uses te-topology-augment;
}

augment "/nw:networks/nw:network/nw:node" {
```



```
when "../nw:network-types/te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
}
description
 "Configuration parameters for TE at node level.";
uses te-node-augment;
}

augment "/nw:networks/nw:network/nt:link" {
 when "../nw:network-types/te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at link level";
 uses te-link-augment;
}

augment "/nw:networks/nw:network/nw:node/"
 + "nt:termination-point" {
 when "../nw:network-types/te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at termination point level";
 uses te-termination-point-augment;
}
}
<CODE ENDS>
```

## 8. Security Considerations

The transport protocol used for retrieving/manipulating the TE topology data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

## 9. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology  
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-te-topology  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology  
prefix: tet

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.
- [RFC3945] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", October 2004.
- [YANG-NET-TOPO] Clemm, A., "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo (Work in Progress).
- [YANG-PUSH] Clemm, A., "Subscribing to YANG datastore push updates", draft-clemm-netconf-yang-push (Work in Progress).
- [RFC5277bis] Clemm, A., "Subscribing to Event Notifications", draft-ietf-netconf-rfc5277bis (Work in Progress).

[YANG-SCHEDULE] Liu, X., " A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule-00 (Work in Progress).

## 10.2. Informative References

[RFC2702] Awduche, D., "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

## 11. Acknowledgments

The authors would like to thank Lou Berger, Sue Hares, Mazen Khaddam, Cyril Margaria and Zafar Ali for participating in design discussions and providing valuable insights.

## Contributors

Sergio Belotti  
Nokia  
Email: sergio.belotti@nokia.com

Dieter Beller  
Nokia  
Email: Dieter.Beller@nokia.com

## Authors' Addresses

Xufeng Liu  
Jabil  
Email: Xufeng\_Liu@jabil.com

Igor Bryskin  
Huawei Technologies  
Email: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram  
Juniper Networks  
Email: vbeeram@juniper.net

Tarek Saad  
Cisco Systems Inc  
Email: tsaad@cisco.com

Himanshu Shah  
Ciena  
Email: hshah@ciena.com

Oscar Gonzalez De Dios  
Telefonica  
Email: oscar.gonzalezdedios@telefonica.com



TEAS Working Group  
Internet Draft  
Intended status: Standards Track

Xufeng Liu  
Volta Networks  
Igor Bryskin  
Huawei Technologies  
Vishnu Pavan Beeram  
Tarek Saad  
Juniper Networks  
Himanshu Shah  
Ciena  
Oscar Gonzalez De Dios  
Telefonica

Expires: December 19, 2019

June 19, 2019

YANG Data Model for Traffic Engineering (TE) Topologies  
draft-ietf-teas-yang-te-topo-22

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on December 19, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Abstract

This document defines a YANG data model for representing, retrieving and manipulating Traffic Engineering (TE) Topologies. The model serves as a base model that other technology specific TE Topology models can augment.

## Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## Table of Contents

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. Introduction.....                                      | 3  |
| 1.1. Terminology.....                                     | 4  |
| 1.2. Tree Structure.....                                  | 4  |
| 1.3. Prefixes in Data Node Names.....                     | 5  |
| 2. Characterizing TE Topologies.....                      | 5  |
| 3. Modeling Abstractions and Transformations.....         | 7  |
| 3.1. TE Topology.....                                     | 7  |
| 3.2. TE Node.....                                         | 7  |
| 3.3. TE Link.....                                         | 8  |
| 3.4. Transitional TE Link for Multi-Layer Topologies..... | 8  |
| 3.5. TE Link Termination Point (LTP).....                 | 10 |
| 3.6. TE Tunnel Termination Point (TTP).....               | 10 |
| 3.7. TE Node Connectivity Matrix.....                     | 11 |
| 3.8. TTP Local Link Connectivity List (LLCL).....         | 11 |
| 3.9. TE Path.....                                         | 11 |
| 3.10. TE Inter-Layer Lock.....                            | 12 |
| 3.11. Underlay TE topology.....                           | 13 |
| 3.12. Overlay TE topology.....                            | 13 |
| 3.13. Abstract TE topology.....                           | 13 |
| 4. Model Applicability.....                               | 14 |
| 4.1. Native TE Topologies.....                            | 14 |

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
| 4.2. Customized TE Topologies.....                                                   | 16  |
| 4.3. Merging TE Topologies Provided by Multiple Providers.....                       | 19  |
| 4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same Provider..... | 22  |
| 5. Modeling Considerations.....                                                      | 25  |
| 5.1. Network topology building blocks.....                                           | 25  |
| 5.2. Technology agnostic TE Topology model.....                                      | 25  |
| 5.3. Model Structure.....                                                            | 26  |
| 5.4. Topology Identifiers.....                                                       | 27  |
| 5.5. Generic TE Link Attributes.....                                                 | 27  |
| 5.6. Generic TE Node Attributes.....                                                 | 28  |
| 5.7. TED Information Sources.....                                                    | 29  |
| 5.8. Overlay/Underlay Relationship.....                                              | 30  |
| 5.9. Templates.....                                                                  | 31  |
| 5.10. Scheduling Parameters.....                                                     | 32  |
| 5.11. Notifications.....                                                             | 33  |
| 6. Guidance for Writing Technology Specific TE Topology Augmentations .....          | 33  |
| 7. TE Topology YANG Module.....                                                      | 46  |
| 8. Security Considerations.....                                                      | 92  |
| 9. IANA Considerations.....                                                          | 94  |
| 10. References.....                                                                  | 95  |
| 10.1. Normative References.....                                                      | 95  |
| 10.2. Informative References.....                                                    | 96  |
| 11. Acknowledgments.....                                                             | 100 |
| Appendix A. Complete Model Tree Structure.....                                       | 101 |
| Appendix B. Companion YANG Model for Non-NMDA Compliant Implementations.....         | 163 |
| Appendix C. Example: YANG Model for Technology Specific Augmentations .....          | 172 |
| Contributors.....                                                                    | 210 |
| Authors' Addresses.....                                                              | 210 |

## 1. Introduction

The Traffic Engineering Database (TED) is an essential component of Traffic Engineered (TE) systems that are based on MPLS-TE [RFC2702] and GMPLS [RFC3945]. The TED is a collection of all TE information about all TE nodes and TE links in the network. The TE Topology is a schematic arrangement of TE nodes and TE links present in a given TED. There could be one or more TE Topologies present in a given Traffic Engineered system. A TE Topology is the topology on which path computational algorithms are run to compute Traffic Engineered Paths (TE Paths).

This document defines a YANG [RFC7950] data model for representing and manipulating TE Topologies. This model contains technology



agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

### 1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The reader is assumed to be familiar with general body of work captured in currently available TE related RFCs. [RFC7926] serves as a good starting point for those who may be less familiar with Traffic Engineering related RFCs.

Some of the key terms used in this document are:

**TED:** The Traffic Engineering Database is a collection of all TE information about all TE nodes and TE links in a given network.

**TE-Topology:** The TE Topology is a schematic arrangement of TE nodes and TE links in a given TED. It forms the basis for a graph suitable for TE path computations.

**Native TE Topology:** Native TE Topology is a topology that is native to a given provider network. Native TE topology could be discovered via various routing protocols and/or subscribe/publish techniques. This is the topology on which path computational algorithms are run to compute TE Paths.

**Customized TE Topology:** Customized TE Topology is a custom topology that is produced by a provider for a given client. This topology typically makes abstractions on the provider's Native TE Topology, and is provided to the client. The client receives the Customized TE Topology, and merges it into the client's Native TE Topology. The client's path computational algorithms aren't typically run on the Customized TE Topology; they are run on the client's Native TE Topology after the merge.

### 1.2. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A. of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

### 1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

| Prefix   | YANG module           | Reference                     |
|----------|-----------------------|-------------------------------|
| yang     | ietf-yang-types       | [RFC6991]                     |
| inet     | ietf-inet-types       | [RFC6991]                     |
| nw       | ietf-network          | [RFC6991]                     |
| nt       | ietf-network-topology | [RFC8345]                     |
| te-types | ietf-te-types         | [I-D.ietf-teas-yang-te-types] |

Table 1: Prefixes and corresponding YANG modules

## 2. Characterizing TE Topologies

The data model proposed by this document takes the following characteristics of TE Topologies into account:

- TE Topology is an abstract control-plane representation of the data-plane topology. Hence attributes specific to the data-plane must make their way into the corresponding TE Topology modeling. The TE Topology comprises of dynamic auto-discovered data as well as fairly static data associated with data-plane nodes and links. The dynamic data may change frequently, such as unreserved bandwidth available on data-plane links. The static data rarely changes, such as layer network identification, switching and adaptation capabilities and limitations, fate sharing, and administrative colors. It is possible for a single TE Topology to encompass TE information at multiple switching layers.
- TE Topologies are protocol independent. Information about topological elements may be learnt via link-state protocols, but the topology can exist without being dependent on any particular protocol.
- TE Topology may not be congruent to the routing topology in a given TE System. The routing topology is constructed based on routing adjacencies. There isn't always a one-to-one association between a TE-link and a routing adjacency. For example, the presence of a TE link between a pair of nodes doesn't necessarily imply the existence of a routing-adjacency between these nodes. To

learn more, see [I-D.ietf-teas-te-topo-and-tunnel-modeling] and [I-D.ietf-teas-yang-l3-te-topo].

- Each TE Topological element has at least one information source associated with it. In some scenarios, there could be more than one information source associated with any given topological element.
- TE Topologies can be hierarchical. Each node and link of a given TE Topology can be associated with respective underlay topology. This means that each node and link of a given TE Topology can be associated with an independent stack of supporting TE Topologies.
- TE Topologies can be customized. TE topologies of a given network presented by the network provider to its client could be customized on per-client request basis. This customization could be performed by provider, by client or by provider/client negotiation. The relationship between a customized topology and provider's native topology could be captured as hierarchical (overlay-underlay), but otherwise the two topologies are decoupled from each other. A customized topology is presented to the client, while provider's native topology is known in its entirety to the provider itself.

### 3. Modeling Abstractions and Transformations

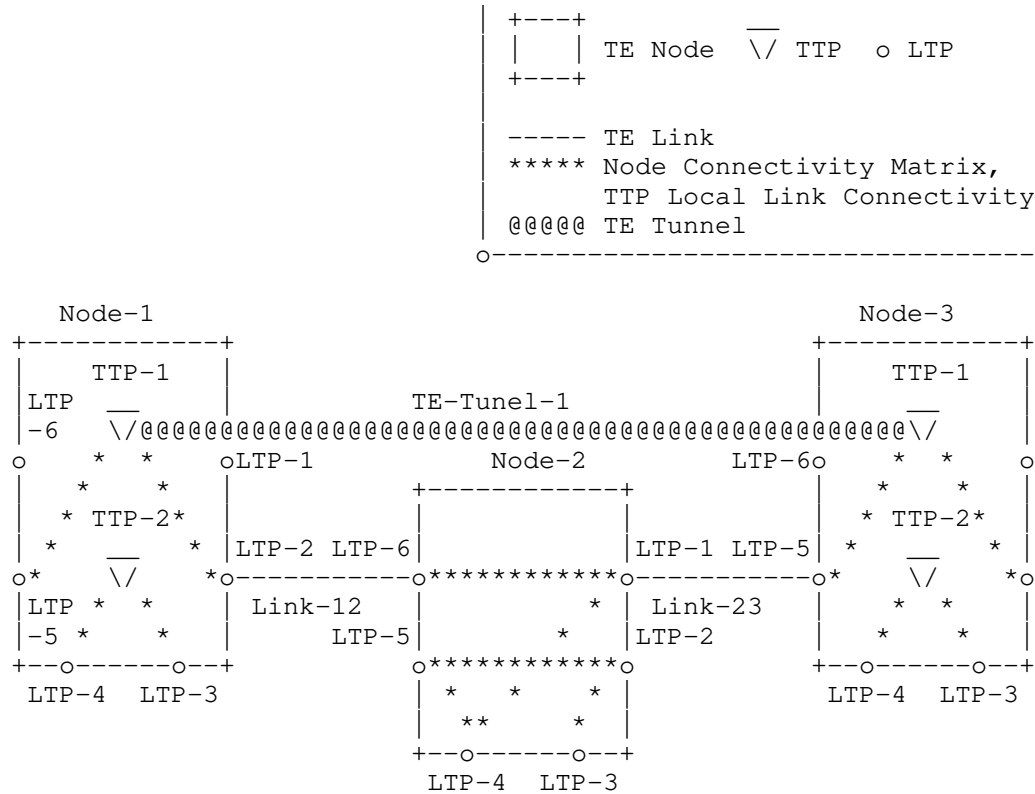


Figure 1: TE Topology Modeling Abstractions

### 3.1. TE Topology

TE topology is a traffic engineering representation of one or more layers of network topologies. TE topology is comprised of TE nodes (TE graph vertices) interconnected via TE links (TE graph edges). A TE topology is mapped to a TE graph.

### 3.2. TE Node

TE node is an element of a TE topology, presented as a vertex on TE graph. TE node represents one or several nodes, or a fraction of a node, which can be a switch or router that is physical or virtual. TE node belongs to and is fully defined in exactly one TE topology. TE node is assigned a unique ID within the TE topology scope. TE node attributes include information related to the data plane aspects of

the associated node(s) (e.g. connectivity matrix), as well as configuration data (such as TE node name). A given TE node can be reached on the TE graph over one of TE links terminated by the TE node.

Multi-layer TE nodes providing switching functions at multiple network layers are an example where a physical node can be decomposed into multiple logical TE nodes, which are fractions of the physical node. Some of these (logical) TE nodes may reside in the client layer TE topology while the remaining TE nodes belong to the server layer TE topology.

In Figure 1, Node-1, Node-2, and Node-3 are TE nodes.

### 3.3. TE Link

TE link is an element of a TE topology, presented as an edge on TE graph. The arrows on an edge indicate one or both directions of the TE link. When there are a pair of parallel links of opposite directions, an edge without arrows is also used. TE link represents one or several (physical) links or a fraction of a link. TE link belongs to and is fully defined in exactly one TE topology. TE link is assigned a unique ID within the TE topology scope. TE link attributes include parameters related to the data plane aspects of the associated link(s) (e.g. unreserved bandwidth, resource maps/pools, etc.), as well as the configuration data (such as remote node/link IDs, SRLGs, administrative colors, etc.). TE link is connected to TE node, terminating the TE link via exactly one TE link termination point (LTP).

In Figure 1, Link-12 and Link-23 are TE links.

### 3.4. Transitional TE Link for Multi-Layer Topologies

Networks are typically composed of multiple network layers where one or multiple signals in the client layer network can be multiplexed and encapsulated into a server layer signal [RFC5212] [G.805]. The server layer signal can be carried in the server layer network across multiple nodes until the server layer signal is terminated and the client layer signals reappear in the node that terminates the server layer signal. Examples of multi-layer networks are: IP over MPLS over Ethernet, low order Optical Data Unit-k (ODUk) signals multiplexed into a high order ODUL (l>k) carried over an Optical Channel (OCh) signal in an optical transport network as defined in [G.872] and [G.709].

TE links as defined in Section 3.3. can be used to represent links within a network layer. In case of a multi-layer network, TE nodes and TE links only allow representation of each network layer as a separate TE topology. Each of these single layer TE topologies would be isolated from their client and their server layer TE topology, if present. The highest and the lowest network layer in the hierarchy only have a single adjacent layer below or above, respectively. Multiplexing of client layer signals and encapsulating them into a server layer signal requires a function that is provided inside a node (typically realized in hardware). This function is also called layer transition.

One of the key requirements for path computation is to be able to calculate a path between two endpoints across a multi-layer network based on the TE topology representing this multi-layer network. This means that an additional TE construct is needed that represents potential layer transitions in the multi-layer TE-topology that connects the TE-topologies representing each separate network layer. The so-called transitional TE link is such a construct and it represents the layer transition function residing inside a node that is decomposed into multiple logical nodes that are represented as TE nodes (see also the transitional link definition in [G.8080] for the optical transport network). Hence, a transitional TE link connects a client layer node with a server layer node. A TE link as defined in 3.3. has LTPs of exactly the same kind on each link end whereas the transitional TE link has client layer LTPs on the client side of the transitional link and in most cases a single server layer LTP on the server side. It should be noted that transitional links are a helper construct in the multi-layer TE topology and they only exist as long as they are not in use, as they represent potential connectivity. When the server layer trail has been established between the server layer LTP of two transitional links in the server layer network, the resulting client layer link in the data plane will be represented as a normal TE link in the client layer topology. The transitional TE links will re-appear when the server layer trail has been torn down.

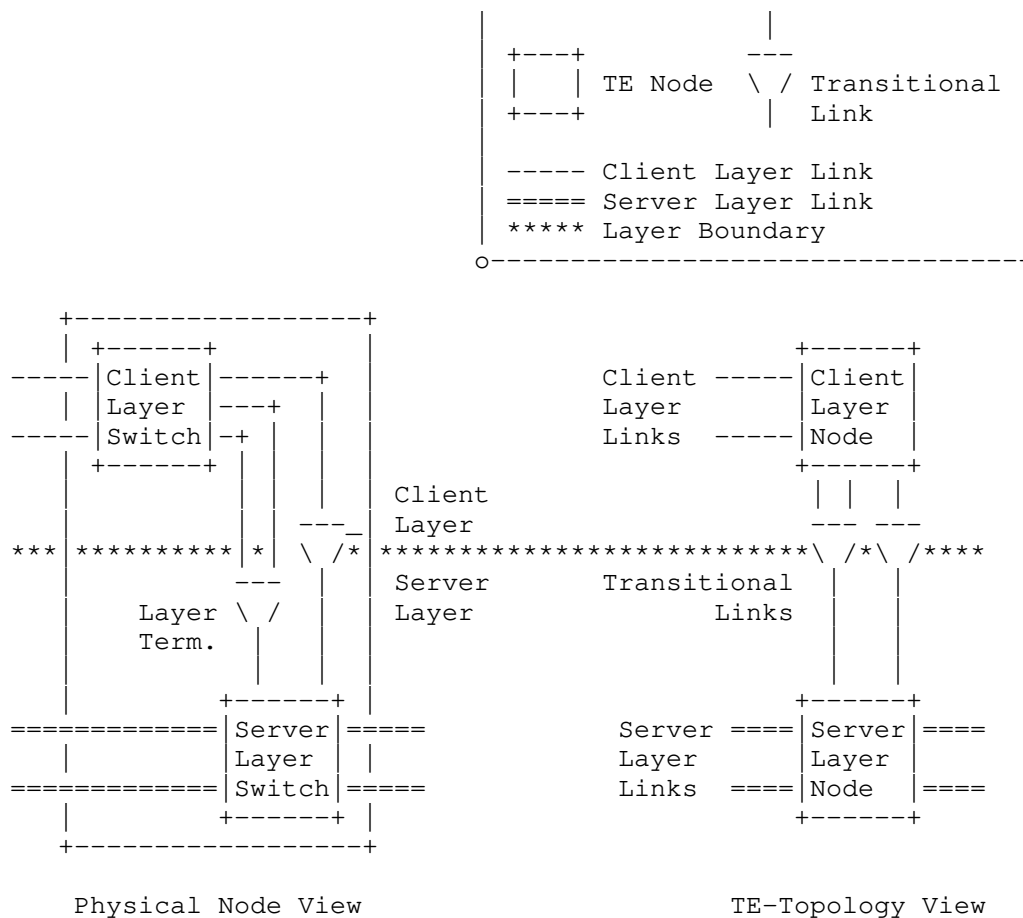


Figure 2: Modeling a Multi-Layer Node (Dual-Layer Example)

### 3.5. TE Link Termination Point (LTP)

TE link termination point (LTP) is a conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1.

In Figure 1, Node-2 has six LTPs: LTP-1 to LTP-6.

### 3.6. TE Tunnel Termination Point (TTP)

TE tunnel termination point (TTP) is an element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as

WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned a unique ID within the TE node scope. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node.

In Figure 1, Node-1 has two TTPs: TTP-1 and TTP-2.

### 3.7. TE Node Connectivity Matrix

TE node connectivity matrix is a TE node's attribute describing the TE node's switching limitations in a form of valid switching combinations of the TE node's LTPs (see below). From the point of view of a potential TE path arriving at the TE node at a given inbound LTP, the node's connectivity matrix describes valid (permissible) outbound LTPs for the TE path to leave the TE node from.

In Figure 1, the connectivity matrix on Node-2 is:  
{<LTP-6, LTP-1>, <LTP-5, LTP-2>, <LTP-5, LTP-4>, <LTP-4, LTP-1>, <LTP-3, LTP-2>}

### 3.8. TTP Local Link Connectivity List (LLCL)

TTP Local Link Connectivity List (LLCL) is a List of TE links terminated by the TTP hosting TE node (i.e. list of the TE link LTPs), which the TTP could be connected to. From the point of view of a potential TE path, LLCL provides a list of valid TE links the TE path needs to start/stop on for the connection, taking the TE path, to be successfully terminated on the TTP in question.

In Figure 1, the LLCL on Node-1 is:  
{<TTP-1, LTP-5>, <TTP-1, LTP-2>, <TTP-2, LTP-3>, <TTP-2, LTP4>}

### 3.9. TE Path

TE path is an ordered list of TE links and/or TE nodes on the TE topology graph, inter-connecting a pair of TTPs to be taken by a potential connection. TE paths, for example, could be a product of successful path computation performed for a given transport service.

In Figure 1, the TE Path for TE-Tunnel-1 is:  
{Node-1:TTP-1, Link-12, Node-2, Link-23, Node-3:TTP1}



## 3.10. TE Inter-Layer Lock

TE inter-layer lock is a modeling concept describing client-server layer adaptation relationships and hence important for the multi-layer traffic engineering. It is an association of M client layer LTPs and N server layer TTPs, within which data arriving at any of the client layer LTPs could be adopted onto any of the server layer TTPs. TE inter-layer lock is identified by inter-layer lock ID, which is unique across all TE topologies provided by the same provider. The client layer LTPs and the server layer TTPs associated within a given TE inter-layer lock are annotated with the same inter-layer lock ID attribute.

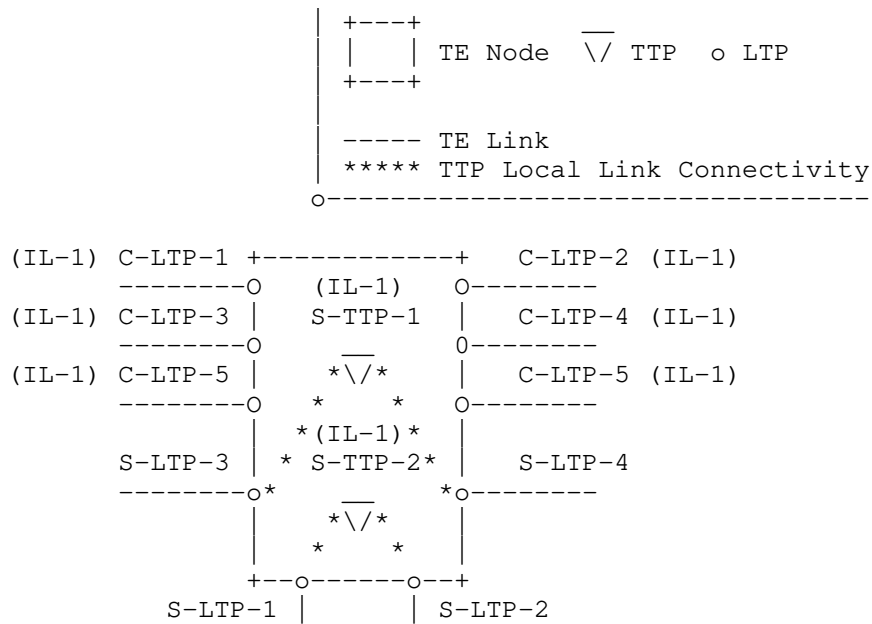


Figure 3: TE Inter-Layer Lock ID Associations

On the picture above a TE inter-layer lock with IL\_1 ID associates 6 client layer LTPs (C-LTP-1 - C-LTP-6) with two server layer TTPs (S-TTP-1 and S-TTP-2). They all have the same attribute - TE inter-layer lock ID: IL-1, which is the only thing that indicates the association. A given LTP may have 0, 1 or more inter-layer lock IDs. In the latter case this means that the data arriving at the LTP may be adopted onto any of TTPs associated with all specified inter-layer locks. For example, C-LTP-1 could have two inter-layer lock IDs - IL-1 and IL-2. This would mean that C-LTP-1 for adaptation purposes could use not just TTPs associated with inter-layer lock IL-1 (i.e.

S-TTP-1 and S-TTP-2 on the picture), but any of TTPs associated with inter-layer lock IL-2 as well. Likewise, a given TTP may have one or more inter-layer lock IDs, meaning that it can offer the adaptation service to any of client layer LTPs with inter-layer lock ID matching one of its own. Additionally, each TTP has an attribute - Unreserved Adaptation Bandwidth, which announces its remaining adaptation resources sharable between all potential client LTPs.

LTPs and TTPs associated within the same TE inter-layer lock may be hosted by the same (hybrid, multi-layer) TE node or multiple TE nodes located in the same or separate TE topologies. The latter is especially important since TE topologies of different layer networks could be modeled by separate augmentations of the basic (common to all layers) TE topology model.

### 3.11. Underlay TE topology

Underlay TE topology is a TE topology that serves as a base for constructing of overlay TE topologies

### 3.12. Overlay TE topology

Overlay TE topology is a TE topology constructed based on one or more underlay TE topologies. Each TE node of the overlay TE topology represents an arbitrary segment of an underlay TE topology; each TE link of the overlay TE topology represents an arbitrary TE path in one of the underlay TE topologies. The overlay TE topology and the supporting underlay TE topologies may represent distinct layer networks (e.g. OTN/ODUk and WDM/OCh respectively) or the same layer network.

### 3.13. Abstract TE topology

Abstract TE topology is a topology that contains abstract topological elements (nodes, links, tunnel termination points). Abstract TE topology is an overlay TE topology created by a topology provider and customized for a topology provider's client based on one or more of the provider's native TE topologies (underlay TE topologies), the provider's policies and the client's preferences. For example, a first level topology provider (such as Domain Controller) can create an abstract TE topology for its client (e.g. Multi-Domain Service Coordinator) based on the provider's one or more native TE topologies, local policies/profiles and the client's TE topology configuration requests

Figure 4 shows an example of abstract TE topology.

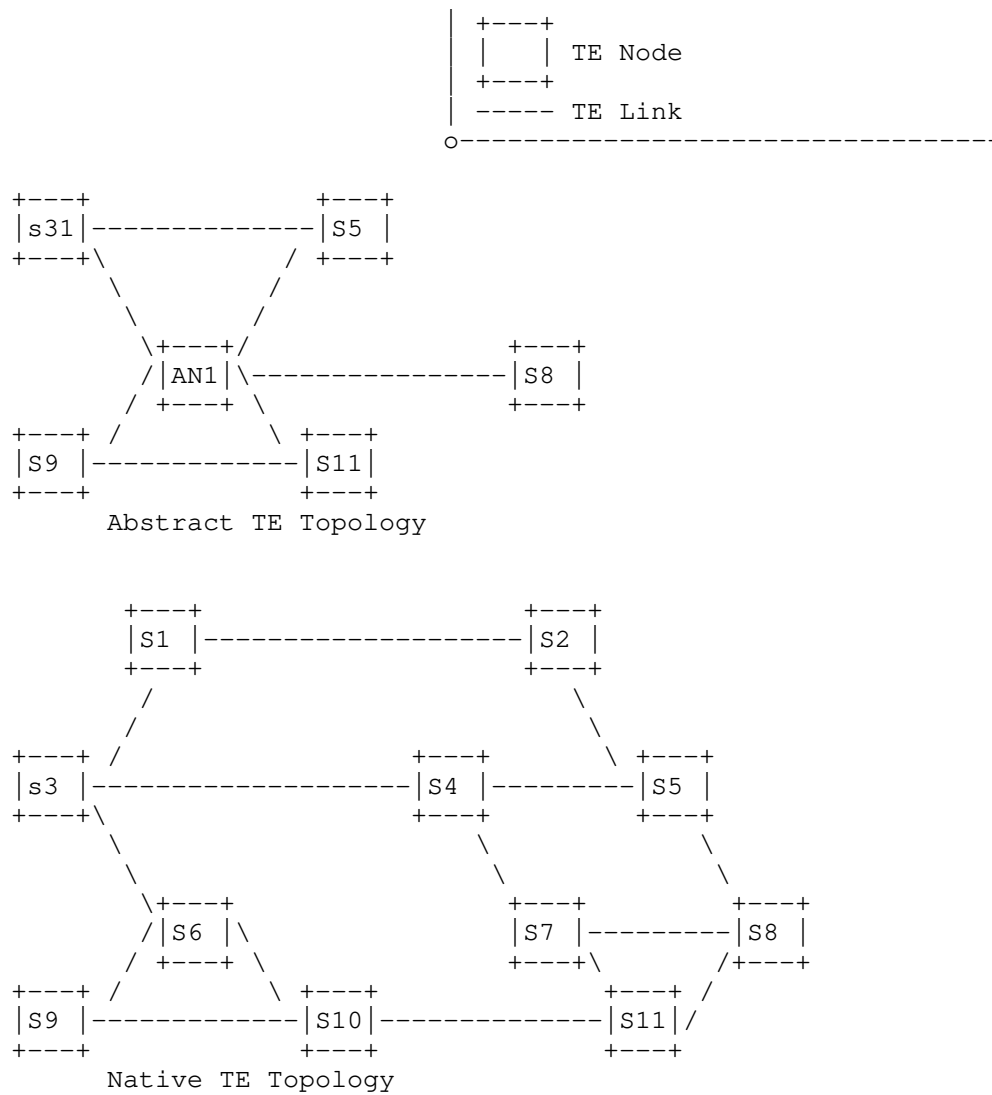


Figure 4: Abstract TE Topology

#### 4. Model Applicability

##### 4.1. Native TE Topologies

The model discussed in this draft can be used to represent and retrieve native TE topologies on a given TE system.

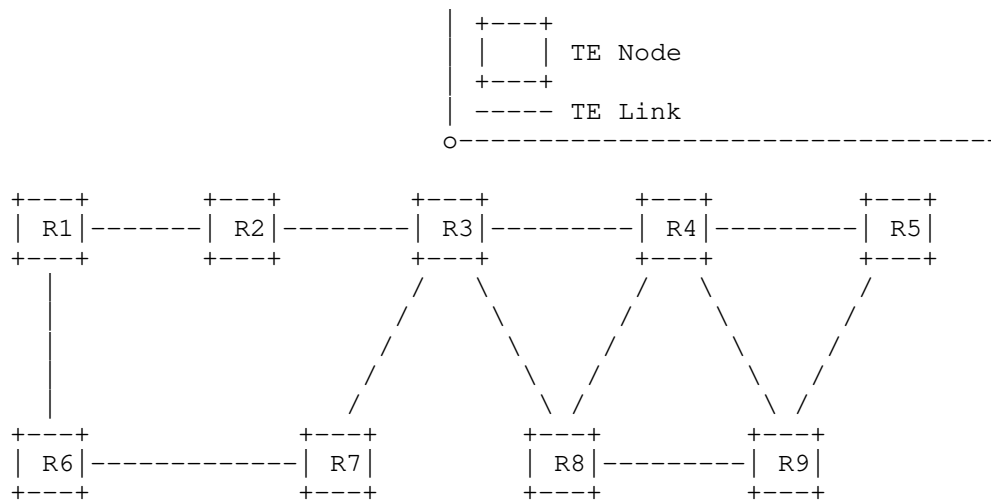


Figure 5a: Example Network Topology

Consider the network topology depicted in Figure 5a. R1 .. R9 are nodes representing routers. An implementation MAY choose to construct a native TE Topology using all nodes and links present in the given TED as depicted in Figure 5b. The data model proposed in this document can be used to retrieve/represent this TE topology.

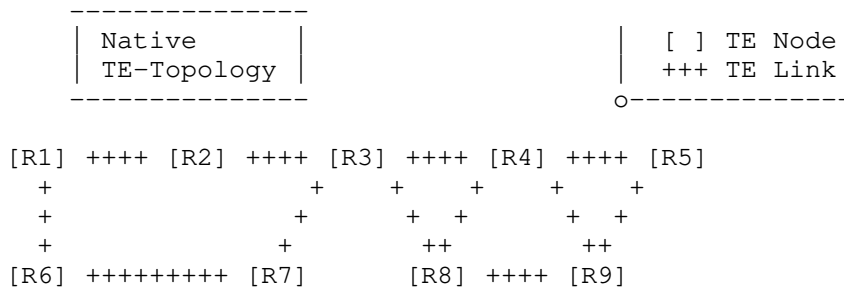


Figure 5b: Native TE Topology as seen on Node R3

Consider the case of the topology being split in a way that some nodes participate in OSPF-TE while others participate in ISIS-TE (Figure 6a). An implementation MAY choose to construct separate TE Topologies based on the information source. The native TE Topologies constructed using only nodes and links that were learnt via a specific information source are depicted in Figure 6b. The data model proposed in this document can be used to retrieve/represent these TE topologies.

Similarly, the data model can be used to represent/retrieve a TE Topology that is constructed using only nodes and links that belong to a particular technology layer. The data model is flexible enough to retrieve and represent many such native TE Topologies.

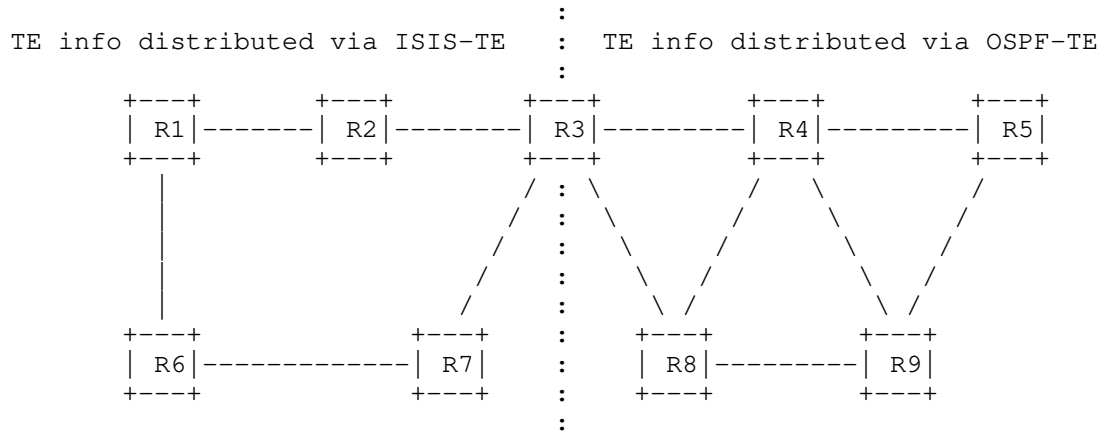


Figure 6a: Example Network Topology

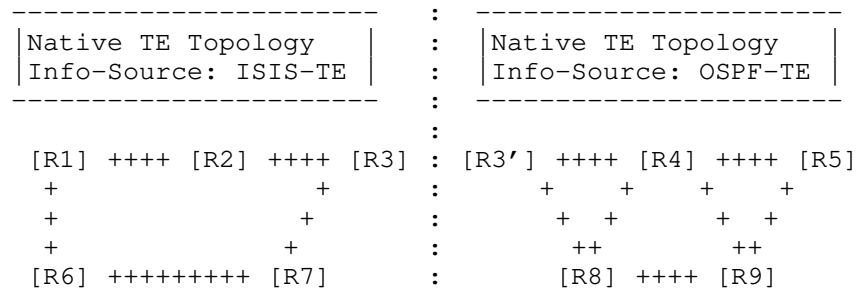


Figure 6b: Native TE Topologies as seen on Node R3

## 4.2. Customized TE Topologies

Customized TE topology is a topology that was modified by the provider to honor a particular client's requirements or preferences. The model discussed in this draft can be used to represent, retrieve and manipulate customized TE Topologies. The model allows the provider to present the network in abstract TE Terms on a per client

basis. These customized topologies contain sufficient information for the path computing client to select paths according to its policies.

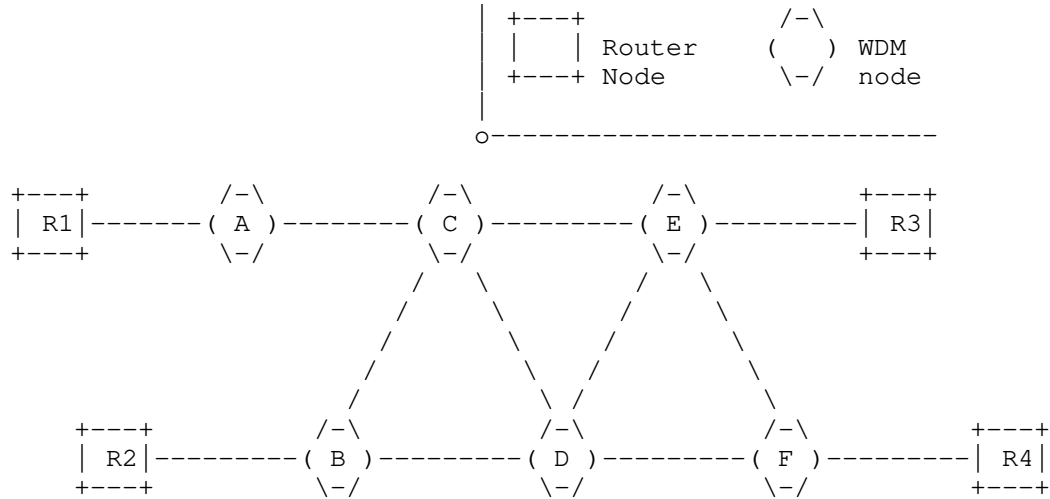


Figure 7: Example packet optical topology

Consider the network topology depicted in Figure 7. This is a typical packet optical transport deployment scenario where the WDM layer network domain serves as a Server Network Domain providing transport connectivity to the packet layer network Domain (Client Network Domain). Nodes R1, R2, R3 and R4 are IP routers that are connected to an Optical WDM transport network. A, B, C, D, E and F are WDM nodes that constitute the Server Network Domain.

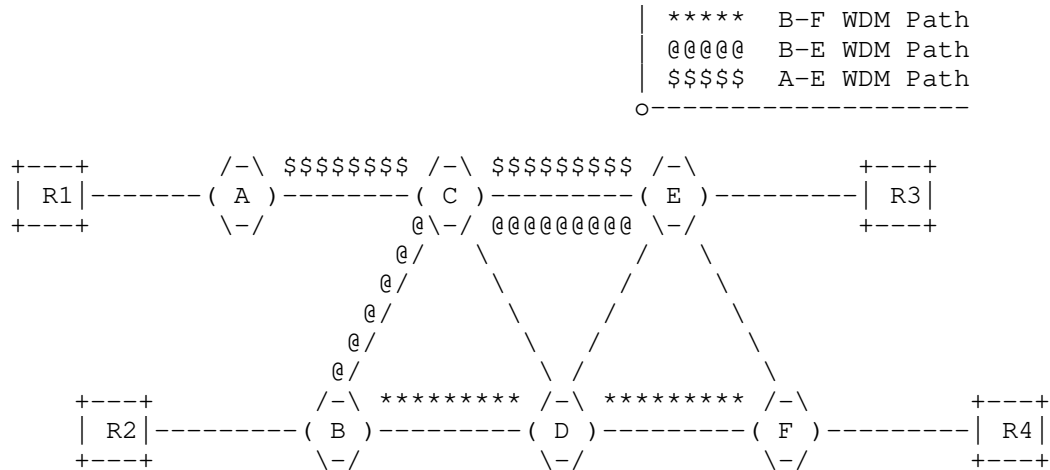


Figure 8a: Paths within the provider domain

```

+++++++ [A] ++++++ [E] ++++++
 ++++++
 ++++++
 ++++++
 ++++++
 ++++++
+++++++ [B] ++++++ [F] ++++++

```

Figure 8b: Customized TE Topology provided to the Client

The goal here is to augment the Client TE Topology with a customized TE Topology provided by the WDM network. Given the availability of the paths A-E, B-F and B-E (Figure 8a), a customized TE Topology as depicted in Figure 8b is provided to the Client. This customized TE Topology is merged with the Client's Native TE Topology and the resulting topology is depicted in Figure 8c.

```

[R1] ++++++ [A] ++++++ [E] ++++++ [R3]
 ++++++
 ++++++
 ++++++
 ++++++
 ++++++
[R2] ++++++ [B] ++++++ [F] ++++++ [R4]

```

Figure 8c: Customized TE Topology merged with the Client's Native TE Topology

The data model proposed in this document can be used to retrieve/represent/manipulate the customized TE Topology depicted in Figure 8b.

A customized TE topology is not necessarily an abstract TE topology. The provider may produce, for example, an abstract TE topology of certain type (e.g. single-abstract-node-with-connectivity-matrix topology, a border-nodes-connected-via-mesh-of-abstract-links topology, etc.) and expose it to all/some clients in expectation that the clients will use it without customization. On the other hand, a client may request a customized version of the provider's native TE topology (e.g. by requesting removal of TE links

which belong to certain layers, are too slow, not protected and/or have a certain affinity). Note that the resulting TE topology will not be abstract (because it will not contain abstract elements), but customized (modified upon client's instructions).

The client ID field in the TE topology identifier (Section 5.4. ) indicates which client the TE topology is customized for. Although an authorized client MAY receive a TE topology with the client ID field matching some other client, the client can customize only TE topologies with the client ID field either 0 or matching the ID of the client in question. If the client starts reconfiguration of a topology its client ID will be automatically set in the topology ID field for all future configurations and updates wrt. the topology in question.

The provider MAY tell the client that a given TE topology cannot be re-negotiated, by setting its own (provider's) ID in the client ID field of the topology ID.

Even though this data model allows to access TE topology information across clients, implementations MAY restrict access for particular clients to particular data fields. The Network Configuration Access Control Model (NACM) [RFC8341] provides such a mechanism.

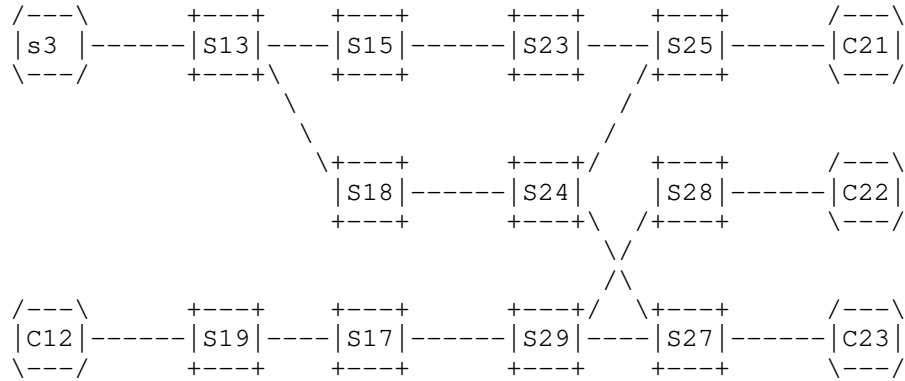
#### 4.3. Merging TE Topologies Provided by Multiple Providers

A client may receive TE topologies provided by multiple providers, each of which managing a separate domain of multi-domain network. In order to make use of said topologies, the client is expected to merge the provided TE topologies into one or more client's native TE topologies, each of which homogeneously representing the multi-domain network. This makes it possible for the client to select end-to-end TE paths for its services traversing multiple domains.

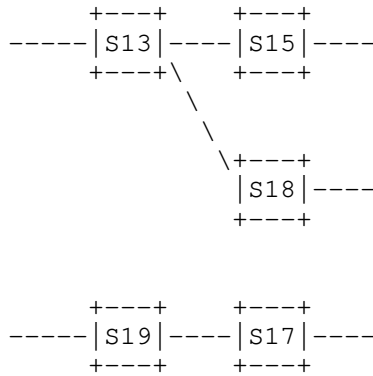
In particular, the process of merging TE topologies includes:

- Identifying neighboring domains and locking their topologies horizontally by connecting their inter-domain open-ended TE links;
- Renaming TE node, link, and SRLG IDs to ones allocated from a separate name space; this is necessary because all TE topologies are considered to be, generally speaking, independent with a possibility of clashes among TE node, link or SRLG IDs;
- Locking, vertically, TE topologies associated with different layer networks, according to provided topology inter-layer locks; this is to facilitate inter-layer path computations across multiple TE topologies provided by the same topology provider.





Domain 1 TE Topology



Domain 2 TE Topology

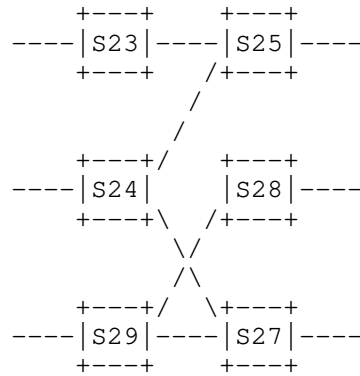


Figure 9: Merging Domain TE Topologies

Figure 9 illustrates the process of merging, by the client, of TE topologies provided by the client's providers. In the Figure, each of the two providers caters to the client (abstract or native) TE topology, describing the network domain under the respective provider's control. The client, by consulting the attributes of the inter-domain TE links - such as inter-domain plug IDs or remote TE node/link IDs (as defined by the TE Topology model) - is able to determine that:

- a) the two domains are adjacent and are inter-connected via three inter-domain TE links, and;

- b) each domain is connected to a separate customer site, connecting the left domain in the Figure to customer devices C-11 and C-12, and the right domain to customer devices C-21, C-22 and C-23.

Therefore, the client inter-connects the open-ended TE links, as shown on the upper part of the Figure.

As mentioned, one way to inter-connect the open-ended inter-domain TE links of neighboring domains is to mandate the providers to specify remote nodeID/linkID attribute in the provided inter-domain TE links. This, however, may prove to be not flexible. For example, the providers may not know the respective remote nodeIDs/ linkIDs. More importantly, this option does not allow for the client to mix-n-match multiple (more than one) topologies catered by the same providers (see below). Another, more flexible, option to resolve the open-ended inter-domain TE links is by annotating them with the inter-domain plug ID attribute. Inter-domain plug ID is a network-wide unique number that identifies on the network a connectivity supporting a given inter-domain TE link. Instead of specifying remote node ID/link ID, an inter-domain TE link may provide a non-zero inter-domain plug ID. It is expected that two neighboring domain TE topologies (provided by separate providers) will have each at least one open-ended inter-domain TE link with an inter-domain plug ID matching to one provided by its neighbor. For example, the inter-domain TE link originating from node S15 of the Domain 1 TE topology (Figure 9) and the inter-domain TE link coming from node S23 of Domain 2 TE topology may specify matching inter-domain plug ID (e.g. 175344). This allows for the client to identify adjacent nodes in the separate neighboring TE topologies and resolve the inter-domain TE links connecting them regardless of their respective nodeIDs/linkIDs (which, as mentioned, could be allocated from independent name spaces). Inter-domain plug IDs may be assigned and managed by a central network authority. Alternatively, inter-domain plug IDs could be dynamically auto-discovered (e.g. via LMP protocol).

Furthermore, the client renames the TE nodes, links and SRLGs offered in the abstract TE topologies by assigning to them IDs allocated from a separate name space managed by the client. Such renaming is necessary, because the two abstract TE topologies may have their own name spaces, generally speaking, independent one from another; hence, ID overlaps/clashes are possible. For example, both TE topologies have TE nodes named S7, which, after renaming, appear in the merged TE topology as S17 and S27, respectively.

Once the merging process is complete, the client can use the merged TE topology for path computations across both domains, for example, to compute a TE path connecting C-11 to C-23.

#### 4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same Provider

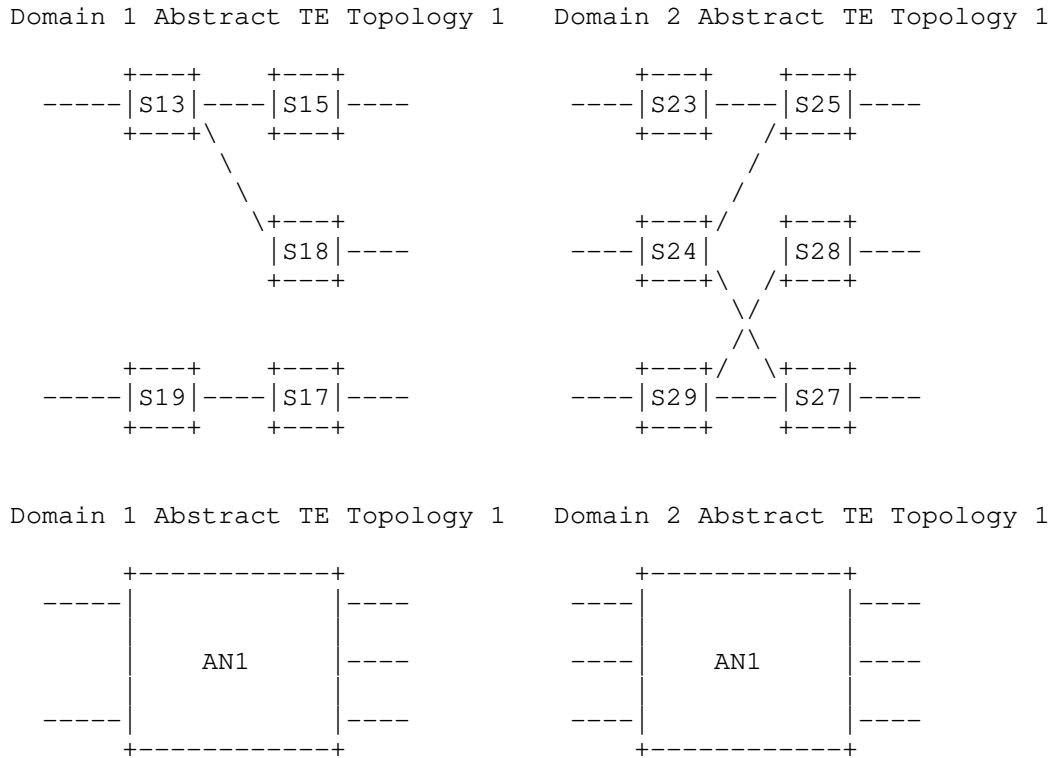


Figure 10: Merging Domain TE Topologies

Based on local configuration, templates and/or policies pushed by the client, a given provider may expose more than one abstract TE topology to the client. For example, one abstract TE topology could be optimized based on a lowest-cost criterion, while another one could be based on best possible delay metrics, while yet another one could be based on maximum bandwidth availability for the client services. Furthermore, the client may request all or some providers to expose additional abstract TE topologies, possibly of a different type and/or optimized differently, as compared to already-provided TE topologies. In any case, the client should be prepared for a provider to offer to the client more than one abstract TE topology.

It should be up to the client (based on the client's local configuration and/or policies conveyed to the client by the client's

clients) to decide how to mix-and-match multiple abstract TE topologies provided by each or some of the providers, as well as how to merge them into the client's native TE topologies. The client also decides how many such merged TE topologies it needs to produce and maintain. For example, in addition to the merged TE topology depicted in the upper part of Figure 9, the client may merge the abstract TE topologies received from the two providers, as shown in Figure 10, into the client's additional native TE topologies, as shown in Figure 11.

Note that allowing for the client mix-n-matching of multiple TE topologies assumes that inter-domain plug IDs (rather than remote nodeID/linkID) option is used for identifying neighboring domains and inter-domain TE link resolution.

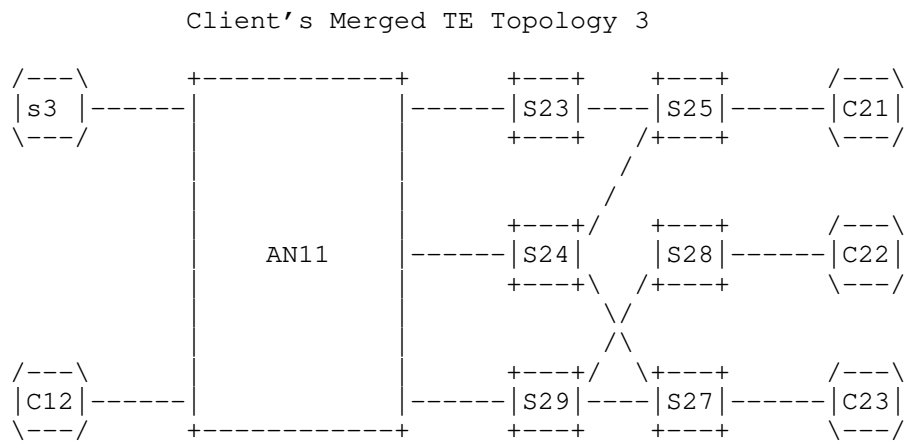
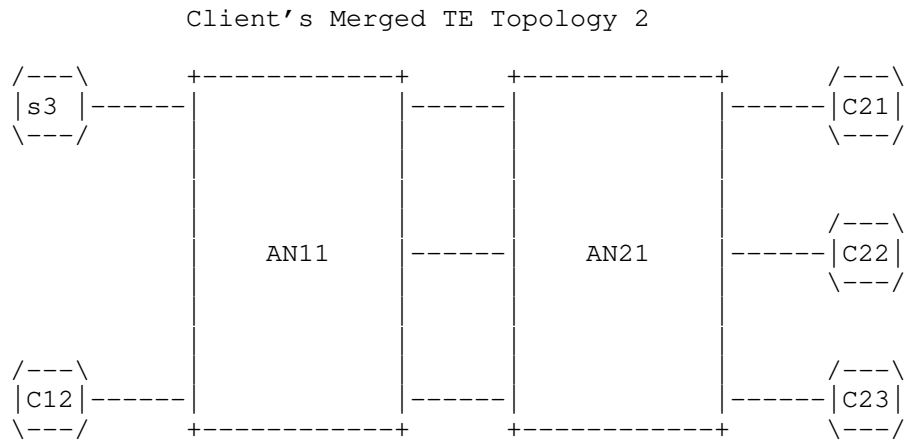


Figure 11: Multiple Native (Merged) Client's TE Topologies

It is important to note that each of the three native (merged) TE topologies could be used by the client for computing TE paths for any of the multi-domain services. The choice as to which topology to use for a given service depends on the service parameters/requirements and the topology's style, optimization criteria and the level of details.

## 5. Modeling Considerations

### 5.1. Network topology building blocks

The network topology building blocks are discussed in [RFC8345]. The TE Topology model proposed in this document augments and uses the `ietf-network-topology` module defined in [RFC8345].

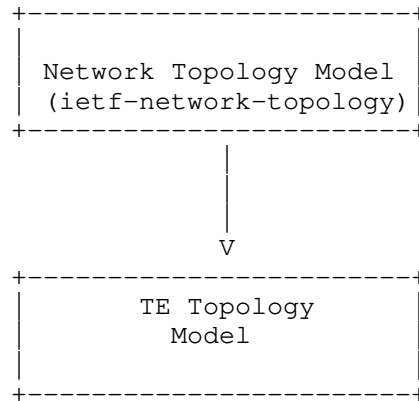


Figure 12: Augmenting the Network Topology Model

### 5.2. Technology agnostic TE Topology model

The TE Topology model proposed in this document is meant to be network technology agnostic. Other technology specific TE Topology models can augment and use the building blocks provided by the proposed model.

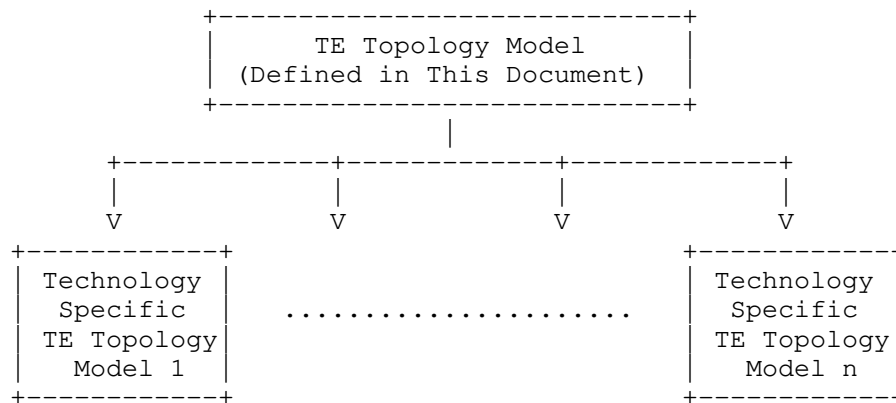


Figure 13: Augmenting the Technology agnostic TE Topology model

### 5.3. Model Structure

The high-level model structure proposed by this document is as shown below:

```

module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
 +--rw te-topology!

augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?
 |
 +--rw link-template* [name] {template}?

augment /nw:networks/nw:network:
 +--rw te-topology-identifier
 | +--rw provider-id? te-global-id
 | +--rw client-id? te-global-id
 | +--rw topology-id? te-topology-id
 +--rw te!
 |

augment /nw:networks/nw:network/nw:node:
 +--rw te-node-id? te-types:te-node-id
 +--rw te!
 |
 +--rw tunnel-termination-point* [tunnel-tp-id]

```

```

 +--rw tunnel-tp-id binary
 |
 +--rw supporting-tunnel-termination-point* [node-ref tunnel-
tp-ref]
 |

```

```

augment /nw:networks/nw:network/nt:link:
 +--rw te!
 |

```

```

augment /nw:networks/nw:network/nw:node/nt:termination-point:
 +--rw te-tp-id? te-types:te-tp-id
 +--rw te!
 |

```

#### 5.4. Topology Identifiers

The TE-Topology is uniquely identified by a key that has 3 constituents - topology-id, provider-id and client-id. The combination of provider-id and topology-id uniquely identifies a native TE Topology on a given provider. The client-id is used only when Customized TE Topologies come into play; a value of "0" is used as the client-id for native TE Topologies.

```

augment /nw:networks/nw:network:
 +--rw te-topology-identifier
 | +--rw provider-id? te-global-id
 | +--rw client-id? te-global-id
 | +--rw topology-id? te-topology-id
 +--rw te!
 |

```

#### 5.5. Generic TE Link Attributes

The model covers the definitions for generic TE Link attributes - bandwidth, admin groups, SRLGs, switching capabilities, TE metric extensions etc.

```

+--rw te-link-attributes

 +--rw admin-status? te-admin-status
 |
 +--rw link-index? uint64
 +--rw administrative-group? te-types:admin-groups
 +--rw link-protection-type? enumeration
 +--rw max-link-bandwidth? te-bandwidth

```



```

+--rw max-resv-link-bandwidth? te-bandwidth
+--rw unreserved-bandwidth* [priority]
|
+--rw te-default-metric? uint32
|
+--rw te-srlgs
+--rw te-nsrlgs {nsrlg}?

```

## 5.6. Generic TE Node Attributes

The model covers the definitions for generic TE Node attributes.

The definition of a generic connectivity matrix is shown below:

```

+--rw te-node-attributes
|
| +--rw connectivity-matrices
| |
| | +--rw connectivity-matrix* [id]
| | | +--rw id uint32
| | | +--rw from
| | | | +--rw tp-ref? leafref
| | | | +--rw label-restrictions
| | | +--rw to
| | | | +--rw tp-ref? leafref
| | | | +--rw label-restrictions
| | | +--rw is-allowed? boolean
| |
| | +--rw underlay! {te-topology-hierarchy}?
| |
| | +--rw path-constraints
| |
| | +--rw optimizations
| |
| | +--rw path-properties
| |
|

```

The definition of a TTP Local Link Connectivity List is shown below:

```

+--rw tunnel-termination-point* [tunnel-tp-id]
| +--rw tunnel-tp-id binary
| +--rw admin-status? te-types:te-admin-status
| +--rw name? string
| +--rw switching-capability? identityref
| +--rw encoding? identityref
| +--rw inter-layer-lock-id* uint32

```

```

 +--rw protection-type? Identityref
 +--rw client-layer-adaptation

 +--rw local-link-connectivities

 | +--rw local-link-connectivity* [link-tp-ref]
 | | +--rw link-tp-ref leafref
 | | +--rw label-restrictions
 |
 | | +--rw is-allowed? boolean
 | | +--rw underlay {te-topology-hierarchy}?
 |
 | | +--rw path-constraints
 |
 | | +--rw optimizations
 |
 | | +--ro path-properties
 |
 +--rw supporting-tunnel-termination-point* [node-ref tunnel-tp-
ref]
 +--rw node-ref inet:uri
 +--rw tunnel-tp-ref binary

```

The attributes directly under container connectivity-matrices are the default attributes for all connectivity-matrix entries when the per entry corresponding attribute is not specified. When a per entry attribute is specified, it overrides the corresponding attribute directly under the container connectivity-matrices. The same rule applies to the attributes directly under container local-link-connectivities.

Each TTP (Tunnel Termination Point) MAY be supported by one or more supporting TTPs. If the TE node hosting the TTP in question refers to a supporting TE node, then the supporting TTPs are hosted by the supporting TE node. If the TE node refers to an underlay TE topology, the supporting TTPs are hosted by one or more specified TE nodes of the underlay TE topology.

### 5.7. TED Information Sources

The model allows each TE topological element to have multiple TE information sources (OSPF-TE, ISIS-TE, BGP-LS, User-Configured, System-Processed, Other). Each information source is associated with a credibility preference to indicate precedence. In scenarios where a customized TE Topology is merged into a Client's native TE Topology, the merged topological elements would point to the corresponding customized TE Topology as its information source.

```

augment /nw:networks/nw:network/nw:node:
 +--rw te!

 +--ro information-source? te-info-source
 +--ro information-source-instance? string
 +--ro information-source-state
 |
 +--ro credibility-preference? uint16
 +--ro logical-network-element? string
 +--ro network-instance? string
 +--ro topology
 |
 +--ro node-ref? leafref
 +--ro network-ref? leafref
 +--ro information-source-entry*
 |
 [information-source information-source-instance]
 +--ro information-source te-info-source
 +--ro information-source-instance string

augment /nw:networks/nw:network/nt:link:
 +--rw te!

 +--ro information-source? te-info-source
 +--ro information-source-instance? string
 +--ro information-source-state
 |
 +--ro credibility-preference? uint16
 +--ro logical-network-element? string
 +--ro network-instance? string
 +--ro topology
 |
 +--ro link-ref? leafref
 +--ro network-ref? leafref
 +--ro information-source-entry*
 |
 [information-source information-source-instance]
 +--ro information-source te-info-source
 +--ro information-source-instance string


```

## 5.8. Overlay/Underlay Relationship

The model captures overlay and underlay relationship for TE nodes/links. For example - in networks where multiple TE Topologies are built hierarchically, this model allows the user to start from a specific topological element in the top most topology and traverse all the way down to the supporting topological elements in the bottom most topology.

This relationship is captured via the "underlay-topology" field for the node and via the "underlay" field for the link. The use of these

fields is optional and this functionality is tagged as a "feature" ("te-topology-hierarchy").

```
augment /nw:networks/nw:network/nw:node:
 +--rw te-node-id? te-types:te-node-id
 +--rw te!
 +--rw te-node-template* leafref {template}?
 +--rw te-node-attributes
 |
 | +--rw underlay-topology {te-topology-hierarchy}?
 | +--rw network-ref? leafref
 |
 +--rw te-link!
 +--rw te-link-attributes
 |
 | +--rw underlay {te-topology-hierarchy}?
 | +--rw enabled? boolean
 | +--rw primary-path
 | | +--rw network-ref? leafref
 | |
 | +--rw backup-path* [index]
 | | +--rw index uint32
 | | +--rw network-ref? leafref
 | |
 | +--rw protection-type? identityref
 | +--rw tunnel-termination-points
 | | +--rw source? binary
 | | +--rw destination? binary
 | +--rw tunnels
 | |
 | |
```

## 5.9. Templates

The data model provides the users with the ability to define templates and apply them to link and node configurations. The use of "template" configuration is optional and this functionality is tagged as a "feature" ("template").

```
augment /nw:networks/nw:network/nw:node:
 +--rw te-node-id? te-types:te-node-id
 +--rw te!
 +--rw te-node-template*
 | -> ../../../../te/templates/node-template/name
 | {template}?
```

```

augment /nw:networks/nw:network/nt:link:
 +--rw te!
 +--rw te-link-template*
 | -> ../../../../te/templates/link-template/name
 | {template}?

augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?
 +--rw name
 | te-types:te-template-name
 +--rw priority? uint16
 +--rw reference-change-policy? enumeration
 +--rw te-node-attributes

 +--rw link-template* [name] {template}?
 +--rw name
 | te-types:te-template-name
 +--rw priority? uint16
 +--rw reference-change-policy? enumeration
 +--rw te-link-attributes


```

Multiple templates can be specified to a configuration element. When two or more templates specify values for the same configuration field, the value from the template with the highest priority is used. The range of the priority is from 0 to 65535, with a lower number indicating a higher priority. The reference-change-policy specifies the action that needs to be taken when the template changes on a configuration element that has a reference to this template. The choices of action include taking no action, rejecting the change to the template and applying the change to the corresponding configuration.

#### 5.10. Scheduling Parameters

The model allows time scheduling parameters to be specified for each topological element or for the topology as a whole. These parameters allow the provider to present different topological views to the client at different time slots. The use of "scheduling parameters" is optional.

The YANG data model for configuration scheduling is defined in [I-D.liu-netmod-yang-schedule], which allows specifying configuration schedules without altering this data model.

### 5.11. Notifications

Notifications are a key component of any topology data model.

[I-D.ietf-netconf-subscribed-notifications] and [I-D.ietf-netconf-yang-push] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

- Subscribe notifications on a per client basis
- Specify subtree filters or xpath filters so that only interested contents will be sent.
- Specify either periodic or on-demand notifications.

### 6. Guidance for Writing Technology Specific TE Topology Augmentations

The TE topology model defined in this document is technology agnostic as it defines concepts, abstractions and attributes that are common across multiple network technologies. It is envisioned that this base model will be widely used when defining technology specific TE topology models for various layer networks.

[I-D.ietf-ccamp-wson-yang], [I-D.ietf-ccamp-otn-topo-yang], and [I-D.ietf-teas-yang-l3-te-topo] are some examples of technology specific TE Topology models. Writers of such models are encouraged to augment the basic TE topology model's containers, such as TE Topology, TE Node, TE Link, Link Termination Point (LTP), Tunnel Termination Point (TTP), Bandwidth and Label with the layer specific attributes instead of defining new containers.

Consider the following technology specific example-topology model:

```
module: example-topology
 augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
 +--rw example-topology!
 augment /nw:networks/nw:network/tet:te:
 +--rw attributes
 +--rw attribute-1? uint8
 augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes:
 +--rw attributes
 +--rw attribute-2? uint8
 augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices:
 +--rw attributes
 +--rw attribute-3? uint8
 augment /nw:networks/nw:network/nw:node/tet:te
```

```

 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix:
 +--rw attributes
 +--rw attribute-3? uint8
 augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point:
 +--rw attributes
 +--rw attribute-4? uint8
 augment /nw:networks/nw:network/nw:node/nt:termination-point
 /tet:te:
 +--rw attributes
 +--rw attribute-5? uint8
 augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes:
 +--rw attributes
 +--rw attribute-6? uint8

```

The technology specific TE bandwidth for this example topology can be specified using the following augment statements:

```

 augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes
 /tet:interface-switching-capability/tet:max-lsp-bandwidth
 /tet:te-bandwidth/tet:technology:
 +--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
 augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes/tet:max-link-bandwidth
 /tet:te-bandwidth/tet:technology:
 +--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
 augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes/tet:max-resv-link-bandwidth
 /tet:te-bandwidth/tet:technology:
 +--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
 augment /nw:networks/tet:te/tet:templates/tet:link-template

```

```

 /tet:te-link-attributes/tet:unreserved-bandwidth
 /tet:te-bandwidth/tet:technology:
+--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:path-constraints/tet:te-bandwidth/tet:technology:
+--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:path-constraints
 /tet:te-bandwidth/tet:technology:
+--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:path-constraints/tet:te-bandwidth/tet:technology:
+--:(example)
 +--ro example
 +--ro bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:path-constraints
 /tet:te-bandwidth/tet:technology:
+--:(example)
 +--ro example
 +--ro bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point/tet:client-layer-adaptation
 /tet:switching-capability/tet:te-bandwidth
 /tet:technology:
+--:(example)
 +--rw example
 +--rw bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:path-constraints

```



```

 /tet:te-bandwidth/tet:technology:
+---: (example)
+---rw example
+---rw bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:path-constraints
 /tet:te-bandwidth/tet:technology:
+---: (example)
+---rw example
+---rw bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes
 /tet:interface-switching-capability/tet:max-lsp-bandwidth
 /tet:te-bandwidth/tet:technology:
+---: (example)
+---rw example
+---rw bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:max-link-bandwidth
 /tet:te-bandwidth/tet:technology:
+---: (example)
+---rw example
+---rw bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:max-resv-link-bandwidth
 /tet:te-bandwidth/tet:technology:
+---: (example)
+---rw example
+---rw bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry
 /tet:interface-switching-capability/tet:max-lsp-bandwidth
 /tet:te-bandwidth/tet:technology:
+---: (example)
+---ro example
+---ro bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry/tet:max-link-bandwidth
 /tet:te-bandwidth/tet:technology:

```

```

 +---:(example)
 +---ro example
 +---ro bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry/tet:max-resv-link-bandwidth
 /tet:te-bandwidth/tet:technology:
 +---:(example)
 +---ro example
 +---ro bandwidth-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry/tet:unreserved-bandwidth
 /tet:te-bandwidth/tet:technology:
 +---:(example)
 +---ro example
 +---ro bandwidth-1? uint32
augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te
 /tet:interface-switching-capability/tet:max-lsp-bandwidth
 /tet:te-bandwidth/tet:technology:
 +---:(example)
 +---rw example
 +---rw bandwidth-1? uint32

```

The technology specific TE label for this example topology can be specified using the following augment statements:

```

augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes/tet:underlay/tet:primary-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
 +---:(example)
 +---rw example
 +---rw label-1? uint32
augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes/tet:underlay/tet:backup-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
 +---:(example)
 +---rw example
 +---rw label-1? uint32
augment /nw:networks/tet:te/tet:templates/tet:link-template

```

```

 /tet:te-link-attributes/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/tet:te/tet:templates/tet:link-template
 /tet:te-link-attributes/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:label-restrictions/tet:label-restriction
 /tet:label-start/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:label-restrictions/tet:label-restriction
 /tet:label-end/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:underlay/tet:primary-path/tet:path-element/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:underlay/tet:backup-path/tet:path-element/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32

```

```

augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:path-properties/tet:path-route-objects
 /tet:path-route-object/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:from/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:from/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:to/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:to/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te

```

```

 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:underlay/tet:primary-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+---:(example)
+---rw example
+---rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:underlay/tet:backup-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+---:(example)
+---rw example
+---rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:path-properties
 /tet:path-route-objects/tet:path-route-object/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(example)
+---ro example
+---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:label-restrictions/tet:label-restriction
 /tet:label-start/tet:te-label/tet:technology:
+---:(example)
+---ro example
+---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:label-restrictions/tet:label-restriction
 /tet:label-end/tet:te-label/tet:technology:
+---:(example)
+---ro example
+---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:underlay/tet:primary-path/tet:path-element/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:

```

```

 +--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:underlay/tet:backup-path/tet:path-element/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:path-properties/tet:path-route-objects
 /tet:path-route-object/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:from/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:from/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:to/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--ro example

```

```

 +---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:to/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+---:(example)
 +---ro example
 +---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:underlay/tet:primary-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+---:(example)
 +---ro example
 +---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:underlay/tet:backup-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+---:(example)
 +---ro example
 +---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:information-source-entry/tet:connectivity-matrices
 /tet:connectivity-matrix/tet:path-properties
 /tet:path-route-objects/tet:path-route-object/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+---:(example)
 +---ro example
 +---ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+---:(example)
 +---rw example
 +---rw label-1? uint32

```

```
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:underlay
 /tet:primary-path/tet:path-element/tet:type/tet:label
 /tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:underlay
 /tet:backup-path/tet:path-element/tet:type/tet:label
 /tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities/tet:path-properties
 /tet:path-route-objects/tet:path-route-object/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
```



```

augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
 +--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:underlay
 /tet:primary-path/tet:path-element/tet:type/tet:label
 /tet:label-hop/tet:te-label/tet:technology:
 +--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:underlay/tet:backup-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
 +--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nw:node/tet:te
 /tet:tunnel-termination-point
 /tet:local-link-connectivities
 /tet:local-link-connectivity/tet:path-properties
 /tet:path-route-objects/tet:path-route-object/tet:type
 /tet:label/tet:label-hop/tet:te-label/tet:technology:
 +--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
 +--:(example)

```

```

 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:underlay/tet:primary-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:te-link-attributes/tet:underlay/tet:backup-path
 /tet:path-element/tet:type/tet:label/tet:label-hop
 /tet:te-label/tet:technology:
+--:(example)
 +--rw example
 +--rw label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry/tet:label-restrictions
 /tet:label-restriction/tet:label-start/tet:te-label
 /tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32
augment /nw:networks/nw:network/nt:link/tet:te
 /tet:information-source-entry/tet:label-restrictions
 /tet:label-restriction/tet:label-end/tet:te-label
 /tet:technology:
+--:(example)
 +--ro example
 +--ro label-1? uint32

```

The YANG module to implement the above example topology can be seen in Appendix C.

## 7. TE Topology YANG Module

This module references [RFC1195], [RFC3209], [RFC3272], [RFC3471], [RFC3630], [RFC3785], [RFC4201], [RFC4202], [RFC4203], [RFC4206], [RFC4872], [RFC5152], [RFC5212], [RFC5305], [RFC5316], [RFC5329], [RFC5392], [RFC6001], [RFC6241], [RFC6991], [RFC7308], [RFC7471], [RFC7579], [RFC7752], [RFC8345], and [I-D.ietf-teas-yang-te-types].

```
<CODE BEGINS> file "ietf-te-topology@2019-02-07.yang"
module ietf-te-topology {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology";

 prefix "tet";

 import ietf-yang-types {
 prefix "yang";
 reference "RFC 6991: Common YANG Data Types";
 }

 import ietf-inet-types {
 prefix "inet";
 reference "RFC 6991: Common YANG Data Types";
 }

 import ietf-te-types {
 prefix "te-types";
 reference
 "I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
 Types";
 }

 import ietf-network {
 prefix "nw";
 reference "RFC 8345: A YANG Data Model for Network Topologies";
 }

 import ietf-network-topology {
 prefix "nt";
 reference "RFC 8345: A YANG Data Model for Network Topologies";
 }
}
```

## organization

"IETF Traffic Engineering Architecture and Signaling (TEAS)  
Working Group";

## contact

"WG Web: <<http://tools.ietf.org/wg/teas/>>

WG List: <<mailto:teas@ietf.org>>

Editor: Xufeng Liu  
<<mailto:xufeng.liu.ietf@gmail.com>>

Editor: Igor Bryskin  
<<mailto:Igor.Bryskin@huawei.com>>

Editor: Vishnu Pavan Beeram  
<<mailto:vbeeram@juniper.net>>

Editor: Tarek Saad  
<<mailto:tsaad@juniper.net>>

Editor: Himanshu Shah  
<<mailto:hshah@ciena.com>>

Editor: Oscar Gonzalez De Dios  
<<mailto:oscar.gonzalezdedios@telefonica.com>>;

## description

"TE topology model for representing and manipulating technology  
agnostic TE Topologies.

Copyright (c) 2019 IETF Trust and the persons identified as  
authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject to  
the license terms contained in, the Simplified BSD License set  
forth in Section 4.c of the IETF Trust's Legal Provisions  
Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the

```
 RFC itself for full legal notices.";

revision "2019-02-07" {
 description "Initial revision";
 reference "RFC XXXX: YANG Data Model for TE Topologies";
 // RFC Ed.: replace XXXX with actual RFC number and remove
 // this note
}

/*
 * Features
 */
feature nsrlg {
 description
 "This feature indicates that the system supports NSRLG
 (Not Sharing Risk Link Group).";
}

feature te-topology-hierarchy {
 description
 "This feature indicates that the system allows underlay
 and/or overlay TE topology hierarchy.";
}

feature template {
 description
 "This feature indicates that the system supports
 template configuration.";
}

/*
 * Typedefs
 */
typedef geographic-coordinate-degree {
 type decimal64 {
 fraction-digits 8;
 }
 description
 "Decimal degree (DD) used to express latitude and longitude
 geographic coordinates.";
} // geographic-coordinate-degree
```

```
typedef te-info-source {
 type enumeration {
 enum "unknown" {
 description "The source is unknown.";
 }
 enum "locally-configured" {
 description "Configured entity.";
 }
 enum "ospfv2" {
 description "OSPFv2.";
 }
 enum "ospfv3" {
 description "OSPFv3.";
 }
 enum "isis" {
 description "ISIS.";
 }
 enum "bgp-ls" {
 description "BGP-LS.";
 reference
 "RFC 7752: North-Bound Distribution of Link-State and
 Traffic Engineering (TE) Information Using BGP";
 }
 enum "system-processed" {
 description "System processed entity.";
 }
 enum "other" {
 description "Other source.";
 }
 }
 description
 "Describing the type of source that has provided the
 related information, and the source credibility.";
} // te-info-source

/*
 * Groupings
 */
grouping connectivity-matrix-entry-path-attributes {
 description
```

```
 "Attributes of connectivity matrix entry.";
 leaf is-allowed {
 type boolean;
 description
 "true - switching is allowed,
 false - switching is disallowed.";
 }
 container underlay {
 if-feature te-topology-hierarchy;
 description "Attributes of the te-link underlay.";
 reference
 "RFC 4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";

 uses te-link-underlay-attributes;
 } // underlay

 uses te-types:generic-path-constraints;
 uses te-types:generic-path-optimization;
 uses te-types:generic-path-properties;
} // connectivity-matrix-entry-path-attributes

grouping geolocation-container {
 description
 "A container containing a GPS location.";
 container geolocation{
 config false;
 description
 "A container containing a GPS location.";
 leaf altitude {
 type int64;
 units millimeter;
 description
 "Distance above the sea level.";
 }
 leaf latitude {
 type geographic-coordinate-degree {
 range "-90..90";
 }
 description

```

```
 "Relative position north or south on the Earth's surface.";
 }
 leaf longitude {
 type geographic-coordinate-degree {
 range "-180..180";
 }
 description
 "Angular distance east or west on the Earth's surface.";
 }
} // gps-location
} // geolocation-container

grouping information-source-state-attributes {
 description
 "The attributes identifying source that has provided the
 related information, and the source credibility.";
 leaf credibility-preference {
 type uint16;
 description
 "The preference value to calculate the traffic
 engineering database credibility value used for
 tie-break selection between different
 information-source values.
 Higher value is more preferable.";
 }
 leaf logical-network-element {
 type string;
 description
 "When applicable, this is the name of a logical network
 element from which the information is learned.";
 } // logical-network-element
 leaf network-instance {
 type string;
 description
 "When applicable, this is the name of a network-instance
 from which the information is learned.";
 } // network-instance
} // information-source-state-attributes

grouping information-source-per-link-attributes {
 description
```



```
 "Per node container of the attributes identifying source that
 has provided the related information, and the source
 credibility.";
 leaf information-source {
 type te-info-source;
 config false;
 description
 "Indicates the type of the information source.";
 }
 leaf information-source-instance {
 type string;
 config false;
 description
 "The name indicating the instance of the information
 source.";
 }
 container information-source-state {
 config false;
 description
 "The container contains state attributes related to
 the information source.";
 uses information-source-state-attributes;
 container topology {
 description
 "When the information is processed by the system,
 the attributes in this container indicate which topology
 is used to process to generate the result information.";
 uses nt:link-ref;
 } // topology
 } // information-source-state
} // information-source-per-link-attributes

grouping information-source-per-node-attributes {
 description
 "Per node container of the attributes identifying source that
 has provided the related information, and the source
 credibility.";
 leaf information-source {
 type te-info-source;
 config false;
 description
```

```
 "Indicates the type of the information source.";
 }
 leaf information-source-instance {
 type string;
 config false;
 description
 "The name indicating the instance of the information
 source.";
 }
 container information-source-state {
 config false;
 description
 "The container contains state attributes related to
 the information source.";
 uses information-source-state-attributes;
 container topology {
 description
 "When the information is processed by the system,
 the attributes in this container indicate which topology
 is used to process to generate the result information.";
 uses nw:node-ref;
 } // topology
 } // information-source-state
} // information-source-per-node-attributes

grouping interface-switching-capability-list {
 description
 "List of Interface Switching Capabilities Descriptors (ISCD)";
 list interface-switching-capability {
 key "switching-capability encoding";
 description
 "List of Interface Switching Capabilities Descriptors (ISCD)
 for this link.";
 reference
 "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
 Signaling Functional Description.
 RFC 4203: OSPF Extensions in Support of Generalized
 Multi-Protocol Label Switching (GMPLS).";
 }
 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 }
}
```

```
 }
 description
 "Switching Capability for this interface.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by this interface.";
 }
 uses te-link-iscd-attributes;
} // interface-switching-capability
} // interface-switching-capability-list

grouping statistics-per-link {
 description
 "Statistics attributes per TE link.";
 leaf discontinuity-time {
 type yang:date-and-time;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
 management subsystem re-initialized itself.";
 }
 /* Administrative attributes */
 leaf disables {
 type yang:counter32;
 description
 "Number of times that link was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that link was enabled.";
 }
 leaf maintenance-clears {
 type yang:counter32;
```

```
 description
 "Number of times that link was put out of maintenance.";
 }
 leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that link was put in maintenance.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that link was modified.";
 }
 /* Operational attributes */
 leaf downs {
 type yang:counter32;
 description
 "Number of times that link was set to operational down.";
 }
 leaf ups {
 type yang:counter32;
 description
 "Number of times that link was set to operational up.";
 }
 /* Recovery attributes */
 leaf fault-clears {
 type yang:counter32;
 description
 "Number of times that link experienced fault clear event.";
 }
 leaf fault-detects {
 type yang:counter32;
 description
 "Number of times that link experienced fault detection.";
 }
 leaf protection-switches {
 type yang:counter32;
 description
 "Number of times that link experienced protection
 switchover.";
 }
}
```

```
leaf protection-reverts {
 type yang:counter32;
 description
 "Number of times that link experienced protection
 reversion.";
}
leaf restoration-failures {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 failure.";
}
leaf restoration-starts {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 start.";
}
leaf restoration-successes {
 type yang:counter32;
 description
 "Number of times that link experienced restoration
 success.";
}
leaf restoration-reversion-failures {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 failure.";
}
leaf restoration-reversion-starts {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 start.";
}
leaf restoration-reversion-successes {
 type yang:counter32;
 description
 "Number of times that link experienced restoration reversion
 success.";
```

```
 }
 } // statistics-per-link

 grouping statistics-per-node {
 description
 "Statistics attributes per TE node.";
 leaf discontinuity-time {
 type yang:date-and-time;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
 management subsystem re-initialized itself.";
 }
 container node {
 description
 "Containing TE node level statistics attributes.";
 leaf disables {
 type yang:counter32;
 description
 "Number of times that node was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that node was enabled.";
 }
 leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that node was put in maintenance.";
 }
 leaf maintenance-clears {
 type yang:counter32;
 description
 "Number of times that node was put out of maintenance.";
 }
 leaf modifies {
 type yang:counter32;
```

```
 description
 "Number of times that node was modified.";
 }
} // node
container connectivity-matrix-entry {
 description
 "Containing connectivity matrix entry level statistics
 attributes.";
 leaf creates {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 created.";
 reference
 "RFC 6241. Section 7.2 for 'create' operation. ";
 }
 leaf deletes {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 deleted.";
 reference
 "RFC 6241. Section 7.2 for 'delete' operation. ";
 }
 leaf disables {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 enabled.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that a connectivity matrix entry was
 modified.";
```

```
 }
 } // connectivity-matrix-entry
} // statistics-per-node

grouping statistics-per-ttp {
 description
 "Statistics attributes per TE TTP (Tunnel Termination Point).";
 leaf discontinuity-time {
 type yang:date-and-time;
 description
 "The time on the most recent occasion at which any one or
 more of this interface's counters suffered a
 discontinuity. If no such discontinuities have occurred
 since the last re-initialization of the local management
 subsystem, then this node contains the time the local
 management subsystem re-initialized itself.";
 }
 container tunnel-termination-point {
 description
 "Containing TE TTP (Tunnel Termination Point) level
 statistics attributes.";
 /* Administrative attributes */
 leaf disables {
 type yang:counter32;
 description
 "Number of times that TTP was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that TTP was enabled.";
 }
 leaf maintenance-clears {
 type yang:counter32;
 description
 "Number of times that TTP was put out of maintenance.";
 }
 leaf maintenance-sets {
 type yang:counter32;
 description
 "Number of times that TTP was put in maintenance.";
 }
 }
}
```



```
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that TTP was modified.";
 }
 /* Operational attributes */
 leaf downs {
 type yang:counter32;
 description
 "Number of times that TTP was set to operational down.";
 }
 leaf ups {
 type yang:counter32;
 description
 "Number of times that TTP was set to operational up.";
 }
 leaf in-service-clears {
 type yang:counter32;
 description
 "Number of times that TTP was taken out of service
 (TE tunnel was released).";
 }
 leaf in-service-sets {
 type yang:counter32;
 description
 "Number of times that TTP was put in service by a TE
 tunnel (TE tunnel was set up).";
 }
} // tunnel-termination-point

container local-link-connectivity {
 description
 "Containing TE LLCL (Local Link Connectivity List) level
 statistics attributes.";
 leaf creates {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was created.";
 reference
 "RFC 6241. Section 7.2 for 'create' operation.";
 }
}
```

```
 }
 leaf deletes {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was deleted.";
 reference
 "RFC 6241. Section 7.2 for 'delete' operation.";
 }
 leaf disables {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was disabled.";
 }
 leaf enables {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was enabled.";
 }
 leaf modifies {
 type yang:counter32;
 description
 "Number of times that an LLCL entry was modified.";
 }
 } // local-link-connectivity
} // statistics-per-ttp

grouping te-link-augment {
 description
 "Augmentation for TE link.";
 uses te-link-config;
 uses te-link-state-derived;
 container statistics {
 config false;
 description
 "Statistics data.";
 uses statistics-per-link;
 } // statistics
} // te-link-augment

grouping te-link-config {
 description
```

```
"TE link configuration grouping.";
choice bundle-stack-level {
 description
 "The TE link can be partitioned into bundled
 links, or component links.";
 case bundle {
 container bundled-links {
 description
 "A set of bundled links.";
 reference
 "RFC 4201: Link Bundling in MPLS Traffic Engineering
 (TE).";
 list bundled-link {
 key "sequence";
 description
 "Specify a bundled interface that is
 further partitioned.";
 leaf sequence {
 type uint32;
 description
 "Identify the sequence in the bundle.";
 }
 } // list bundled-link
 }
 }
 case component {
 container component-links {
 description
 "A set of component links";
 list component-link {
 key "sequence";
 description
 "Specify a component interface that is
 sufficient to unambiguously identify the
 appropriate resources";

 leaf sequence {
 type uint32;
 description
 "Identify the sequence in the bundle.";
 }
 }
 }
 }
}
```

```
 leaf src-interface-ref {
 type string;
 description
 "Reference to component link interface on the
 source node.";
 }
 leaf des-interface-ref {
 type string;
 description
 "Reference to component link interface on the
 destination node.";
 }
 }
}
} // bundle-stack-level

leaf-list te-link-template {
 if-feature template;
 type leafref {
 path "../.../te/templates/link-template/name";
 }
 description
 "The reference to a TE link template.";
}
uses te-link-config-attributes;
} // te-link-config

grouping te-link-config-attributes {
 description
 "Link configuration attributes in a TE topology.";
 container te-link-attributes {
 description "Link attributes in a TE topology.";
 leaf access-type {
 type te-types:te-link-access-type;
 description
 "Link access type, which can be point-to-point or
 multi-access.";
 }
 container external-domain {
 description

```

```
 "For an inter-domain link, specify the attributes of
 the remote end of link, to facilitate the signalling at
 local end.";
 uses nw:network-ref;
 leaf remote-te-node-id {
 type te-types:te-node-id;
 description
 "Remote TE node identifier, used together with
 remote-te-link-id to identify the remote link
 termination point in a different domain.";
 }
 leaf remote-te-link-tp-id {
 type te-types:te-tp-id;
 description
 "Remote TE link termination point identifier, used
 together with remote-te-node-id to identify the remote
 link termination point in a different domain.";
 }
}
leaf is-abstract {
 type empty;
 description "Present if the link is abstract.";
}
leaf name {
 type string;
 description "Link Name.";
}
container underlay {
 if-feature te-topology-hierarchy;
 description "Attributes of the te-link underlay.";
 reference
 "RFC 4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";

 uses te-link-underlay-attributes;
} // underlay
leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
```

```
 }

 uses te-link-info-attributes;
 } // te-link-attributes
} // te-link-config-attributes

grouping te-link-info-attributes {
 description
 "Advertised TE information attributes.";
 leaf link-index {
 type uint64;
 description
 "The link identifier. If OSPF is used, this represents an
 ospfLsdbID. If IS-IS is used, this represents an isisLSPID.
 If a locally configured link is used, this object represents
 a unique value, which is locally defined in a router.";
 }
 leaf administrative-group {
 type te-types:admin-groups;
 description
 "Administrative group or color of the link.
 This attribute covers both administrative group (defined in
 RFC 3630, RFC 5305 and RFC 5329), and extended
 administrative group (defined in RFC 7308).";
 }
}

uses interface-switching-capability-list;
uses te-types:label-set-info;

leaf link-protection-type {
 type identityref {
 base te-types:link-protection-type;
 }
 description
 "Link Protection Type desired for this link.";
 reference
 "RFC 4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
}

container max-link-bandwidth {
```

```
 uses te-types:te-bandwidth;
 description
 "Maximum bandwidth that can be seen on this link in this
 direction. Units in bytes per second.";
 reference
 "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC 5305: IS-IS Extensions for Traffic Engineering.";
}
container max-resv-link-bandwidth {
 uses te-types:te-bandwidth;
 description
 "Maximum amount of bandwidth that can be reserved in this
 direction in this link. Units in bytes per second.";
 reference
 "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC 5305: IS-IS Extensions for Traffic Engineering.";
}
list unreserved-bandwidth {
 key "priority";
 max-elements "8";
 description
 "Unreserved bandwidth for 0-7 priority levels. Units in
 bytes per second.";
 reference
 "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC 5305: IS-IS Extensions for Traffic Engineering.";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "Priority.";
 }
 uses te-types:te-bandwidth;
}
leaf te-default-metric {
 type uint32;
 description
 "Traffic engineering metric.";
```

```
 reference
 "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC 5305: IS-IS Extensions for Traffic Engineering.";
 }
 leaf te-delay-metric {
 type uint32;
 description
 "Traffic engineering delay metric.";
 reference
 "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.";
 }
 leaf te-igp-metric {
 type uint32;
 description
 "IGP metric used for traffic engineering.";
 reference
 "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
 Second MPLS Traffic Engineering (TE) Metric.";
 }
 container te-srlgs {
 description
 "Containing a list of SLRGs.";
 leaf-list value {
 type te-types:srlg;
 description "SRLG value.";
 reference
 "RFC 4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
 }
 }
 container te-nsrlgs {
 if-feature nsrlg;
 description
 "Containing a list of NSRLGs (Not Sharing Risk Link
 Groups).
 When an abstract TE link is configured, this list specifies
 the request that underlay TE paths need to be mutually
 disjoint with other TE links in the same groups.";
 leaf-list id {
 type uint32;
```



```
 description
 "NSRLG ID, uniquely configured within a topology.";
 reference
 "RFC 4872: RSVP-TE Extensions in Support of End-to-End
 Generalized Multi-Protocol Label Switching (GMPLS)
 Recovery";
 }
}
} // te-link-info-attributes

grouping te-link-iscd-attributes {
 description
 "TE link ISCD (Interface Switching Capability Descriptor)
 attributes.";
 reference
 "Sec 1.4, RFC 4203: OSPF Extensions in Support of Generalized
 Multi-Protocol Label Switching (GMPLS). Section 1.4.";
 list max-lsp-bandwidth {
 key "priority";
 max-elements "8";
 description
 "Maximum LSP Bandwidth at priorities 0-7.";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "Priority.";
 }
 uses te-types:te-bandwidth;
 }
} // te-link-iscd-attributes

grouping te-link-state-derived {
 description
 "Link state attributes in a TE topology.";
 leaf oper-status {
 type te-types:te-oper-status;
 config false;
 description
 "The current operational state of the link.";
 }
}
```

```
leaf is-transitional {
 type empty;
 config false;
 description
 "Present if the link is transitional, used as an
 alternative approach in lieu of inter-layer-lock-id
 for path computation in a TE topology covering multiple
 layers or multiple regions.";
 reference
 "RFC 5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
}
uses information-source-per-link-attributes;
list information-source-entry {
 key "information-source information-source-instance";
 config false;
 description
 "A list of information sources learned, including the one
 used.";
 uses information-source-per-link-attributes;
 uses te-link-info-attributes;
}
container recovery {
 config false;
 description
 "Status of the recovery process.";
 leaf restoration-status {
 type te-types:te-recovery-status;
 description
 "Restoration status.";
 }
 leaf protection-status {
 type te-types:te-recovery-status;
 description
 "Protection status.";
 }
}
container underlay {
 if-feature te-topology-hierarchy;
```

```
 config false;
 description "State attributes for te-link underlay.";
 leaf dynamic {
 type boolean;
 description
 "true if the underlay is dynamically created.";
 }
 leaf committed {
 type boolean;
 description
 "true if the underlay is committed.";
 }
}
} // te-link-state-derived

grouping te-link-underlay-attributes {
 description "Attributes for te-link underlay.";
 reference
 "RFC 4206: Label Switched Paths (LSP) Hierarchy with
 Generalized Multi-Protocol Label Switching (GMPLS)
 Traffic Engineering (TE)";
 leaf enabled {
 type boolean;
 description
 "'true' if the underlay is enabled.
 'false' if the underlay is disabled.";
 }
}
container primary-path {
 description
 "The service path on the underlay topology that
 supports this link.";
 uses nw:network-ref;
 list path-element {
 key "path-element-id";
 description
 "A list of path elements describing the service path.";
 leaf path-element-id {
 type uint32;
 description "To identify the element in a path.";
 }
 }
 uses te-path-element;
```

```
 }
 } // primary-path
 list backup-path {
 key "index";
 description
 "A list of backup service paths on the underlay topology that
 protect the underlay primary path. If the primary path is
 not protected, the list contains zero elements. If the
 primary path is protected, the list contains one or more
 elements.";
 leaf index {
 type uint32;
 description
 "A sequence number to identify a backup path.";
 }
 uses nw:network-ref;
 list path-element {
 key "path-element-id";
 description
 "A list of path elements describing the backup service
 path";
 leaf path-element-id {
 type uint32;
 description "To identify the element in a path.";
 }
 uses te-path-element;
 }
 } // underlay-backup-path
 leaf protection-type {
 type identityref {
 base te-types:lsp-protection-type;
 }
 description
 "Underlay protection type desired for this link.";
 }
 container tunnel-termination-points {
 description
 "Underlay TTP (Tunnel Termination Points) desired for this
 link.";
 leaf source {
 type binary;
 }
 }
}
```

```
 description
 "Source tunnel termination point identifier.";
 }
 leaf destination {
 type binary;
 description
 "Destination tunnel termination point identifier.";
 }
}
container tunnels {
 description
 "Underlay TE tunnels supporting this TE link.";
 leaf sharing {
 type boolean;
 default true;
 description
 "'true' if the underlay tunnel can be shared with other
 TE links;
 'false' if the underlay tunnel is dedicated to this
 TE link.
 This leaf is the default option for all TE tunnels,
 and may be overridden by the per TE tunnel value.";
 }
 list tunnel {
 key "tunnel-name";
 description
 "Zero, one or more underlay TE tunnels that support this TE
 link.";
 leaf tunnel-name {
 type string;
 description
 "A tunnel name uniquely identifies an underlay TE tunnel,
 used together with the source-node of this link.
 The detailed information of this tunnel can be retrieved
 from the ietf-te model.";
 reference "RFC 3209";
 }
 }
 leaf sharing {
 type boolean;
 description
 "'true' if the underlay tunnel can be shared with other
```

```
 TE links;
 'false' if the underlay tunnel is dedicated to this
 TE link.";
 }
} // tunnel
} // tunnels
} // te-link-underlay-attributes

grouping te-node-augment {
 description
 "Augmentation for TE node.";
 uses te-node-config;
 uses te-node-state-derived;
 container statistics {
 config false;
 description
 "Statistics data.";
 uses statistics-per-node;
 } // statistics

 list tunnel-termination-point {
 key "tunnel-tp-id";
 description
 "A termination point can terminate a tunnel.";
 leaf tunnel-tp-id {
 type binary;
 description
 "Tunnel termination point identifier.";
 }

 uses te-node-tunnel-termination-point-config;
 leaf oper-status {
 type te-types:te-oper-status;
 config false;
 description
 "The current operational state of the tunnel
 termination point.";
 }
 uses geolocation-container;
 container statistics {
 config false;
```

```
 description
 "Statistics data.";
 uses statistics-per-ttp;
 } // statistics

 // Relations to other tunnel termination points
 list supporting-tunnel-termination-point {
 key "node-ref tunnel-tp-ref";
 description
 "Identifies the tunnel termination points, that this
 tunnel termination point is depending on.";
 leaf node-ref {
 type inet:uri;
 description
 "This leaf identifies the node in which the supporting
 tunnel termination point is present.
 This node is either the supporting node or a node in an
 underlay topology.";
 }
 leaf tunnel-tp-ref {
 type binary;
 description
 "Reference to a tunnel termination point, which is
 either in the supporting node or a node in an
 underlay topology.";
 }
 } // supporting-tunnel-termination-point
} // tunnel-termination-point
} // te-node-augment

grouping te-node-config {
 description "TE node configuration grouping.";
 leaf-list te-node-template {
 if-feature template;
 type leafref {
 path "../.../.../te/templates/node-template/name";
 }
 description
 "The reference to a TE node template.";
 }
 uses te-node-config-attributes;
```

```
 } // te-node-config

 grouping te-node-config-attributes {
 description "Configuration node attributes in a TE topology.";
 container te-node-attributes {
 description "Containing node attributes in a TE topology.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
 }
 uses te-node-connectivity-matrices;
 uses te-node-info-attributes;
 } // te-node-attributes
 } // te-node-config-attributes

 grouping te-node-config-attributes-template {
 description
 "Configuration node attributes for template in a TE topology.";
 container te-node-attributes {
 description "Containing node attributes in a TE topology.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link.";
 }
 uses te-node-info-attributes;
 } // te-node-attributes
 } // te-node-config-attributes-template

 grouping te-node-connectivity-matrices {
 description "Connectivity matrix on a TE node.";
 container connectivity-matrices {
 description
 "Containing connectivity matrix on a TE node.";
 leaf number-of-entries {
 type uint16;
 description
 "The number of connectivity matrix entries.
 If this number is specified in the configuration request,
 the number is requested number of entries, which may not
```



```
 all be listed in the list;
 if this number is reported in the state data,
 the number is the current number of operational entries.";
 }
 uses te-types:label-set-info;
 uses connectivity-matrix-entry-path-attributes;
 list connectivity-matrix {
 key "id";
 description
 "Represents node's switching limitations, i.e. limitations
 in interconnecting network TE links across the node.";
 reference
 "RFC 7579: General Network Element Constraint Encoding
 for GMPLS-Controlled Networks.";
 leaf id {
 type uint32;
 description "Identifies the connectivity-matrix entry.";
 }
 } // connectivity-matrix
} // connectivity-matrices
} // te-node-connectivity-matrices

grouping te-node-connectivity-matrix-attributes {
 description
 "Termination point references of a connectivity matrix entry.";
 container from {
 description
 "Reference to source link termination point.";
 leaf tp-ref {
 type leafref {
 path "../..//../..//../nt:termination-point/nt:tp-id";
 }
 description
 "Relative reference to a termination point.";
 }
 uses te-types:label-set-info;
 }
 container to {
 description
 "Reference to destination link termination point.";
 leaf tp-ref {
```

```
 type leafref {
 path "../.../.../.../nt:termination-point/nt:tp-id";
 }
 description
 "Relative reference to a termination point.";
 }
 uses te-types:label-set-info;
}
uses connectivity-matrix-entry-path-attributes;
} // te-node-connectivity-matrix-attributes

grouping te-node-info-attributes {
 description
 "Advertised TE information attributes.";
 leaf domain-id {
 type uint32;
 description
 "Identifies the domain that this node belongs.
 This attribute is used to support inter-domain links.";
 reference
 "RFC 5152: A Per-Domain Path Computation Method for
 Establishing Inter-Domain Traffic Engineering (TE)
 Label Switched Paths (LSPs).
 RFC 5392: OSPF Extensions in Support of Inter-Autonomous
 System (AS) MPLS and GMPLS Traffic Engineering.
 RFC 5316: ISIS Extensions in Support of Inter-Autonomous
 System (AS) MPLS and GMPLS Traffic Engineering.";
 }
 leaf is-abstract {
 type empty;
 description
 "Present if the node is abstract, not present if the node
 is actual.";
 }
 leaf name {
 type string;
 description "Node name.";
 }
 leaf-list signaling-address {
 type inet:ip-address;
 description "Node signaling address.";
```

```
 }
 container underlay-topology {
 if-feature te-topology-hierarchy;
 description
 "When an abstract node encapsulates a topology,
 the attributes in this container point to said topology.";
 uses nw:network-ref;
 }
 } // te-node-info-attributes

grouping te-node-state-derived {
 description "Node state attributes in a TE topology.";
 leaf oper-status {
 type te-types:te-oper-status;
 config false;
 description
 "The current operational state of the node.";
 }
 uses geolocation-container;
 leaf is-multi-access-dr {
 type empty;
 config false;
 description
 "The presence of this attribute indicates that this TE node
 is a pseudonode elected as a designated router.";
 reference
 "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
 Version 2.
 RFC 1195: Use of OSI IS-IS for Routing in TCP/IP and Dual
 Environments.";
 }
 uses information-source-per-node-attributes;
 list information-source-entry {
 key "information-source information-source-instance";
 config false;
 description
 "A list of information sources learned, including the one
 used.";
 uses information-source-per-node-attributes;
 uses te-node-connectivity-matrices;
 uses te-node-info-attributes;
 }
}
```

```
 }
 } // te-node-state-derived

 grouping te-node-tunnel-termination-point-config {
 description
 "Termination capability of a tunnel termination point on a
 TE node.";
 uses te-node-tunnel-termination-point-config-attributes;
 container local-link-connectivities {
 description
 "Containing local link connectivity list for
 a tunnel termination point on a TE node.";
 leaf number-of-entries {
 type uint16;
 description
 "The number of local link connectivity list entries.
 If this number is specified in the configuration request,
 the number is requested number of entries, which may not
 all be listed in the list;
 if this number is reported in the state data,
 the number is the current number of operational entries.";
 }
 uses te-types:label-set-info;
 uses connectivity-matrix-entry-path-attributes;
 } // local-link-connectivities
 } // te-node-tunnel-termination-point-config

 grouping te-node-tunnel-termination-point-config-attributes {
 description
 "Configuration attributes of a tunnel termination point on a
 TE node.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the tunnel termination point.";
 }
 leaf name {
 type string;
 description
 "A descriptive name for the tunnel termination point.";
 }
 }
```

```
leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for this interface.";
}
leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by this interface.";
}
leaf-list inter-layer-lock-id {
 type uint32;
 description
 "Inter layer lock ID, used for path computation in a TE
 topology covering multiple layers or multiple regions.";
 reference
 "RFC 5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
}
leaf protection-type {
 type identityref {
 base te-types:lsp-protection-type;
 }
 description
 "The protection type that this tunnel termination point
 is capable of.";
}

container client-layer-adaptation {
 description
 "Containing capability information to support a client layer
 adaption in multi-layer topology.";
 list switching-capability {
 key "switching-capability encoding";
 description
```

```
 "List of supported switching capabilities";
 reference
 "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).
 RFC 4202: Routing Extensions in Support of
 Generalized Multi-Protocol Label Switching (GMPLS).";
 leaf switching-capability {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "Switching Capability for the client layer adaption.";
 }
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "Encoding supported by the client layer adaption.";
 }
 uses te-types:te-bandwidth;
}
}
} // te-node-tunnel-termination-point-config-attributes

grouping te-node-tunnel-termination-point-llc-list {
 description
 "Local link connectivity list of a tunnel termination
 point on a TE node.";
 list local-link-connectivity {
 key "link-tp-ref";
 description
 "The termination capabilities between
 tunnel-termination-point and link termination-point.
 The capability information can be used to compute
 the tunnel path.
 The Interface Adjustment Capability Descriptors (IACD)
 (defined in RFC 6001) on each link-tp can be derived from
 this local-link-connectivity list.";
 reference
 "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
```

```
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";

 leaf link-tp-ref {
 type leafref {
 path "../..../nt:termination-point/nt:tp-id";
 }
 description
 "Link termination point.";
 }
 uses te-types:label-set-info;
 uses connectivity-matrix-entry-path-attributes;
} // local-link-connectivity
} // te-node-tunnel-termination-point-config

grouping te-path-element {
 description
 "A group of attributes defining an element in a TE path
 such as TE node, TE link, TE atomic resource or label.";
 uses te-types:explicit-route-hop;
} // te-path-element

grouping te-termination-point-augment {
 description
 "Augmentation for TE termination point.";
 leaf te-tp-id {
 type te-types:te-tp-id;
 description
 "An identifier to uniquely identify a TE termination
 point.";
 }
 container te {
 must "../te-tp-id";
 presence "TE support.";
 description
 "Indicates TE support.";

 uses te-termination-point-config;
 leaf oper-status {
 type te-types:te-oper-status;
 config false;
 description
```

```
 "The current operational state of the link termination
 point.";
 }
 uses geolocation-container;
} // te
} // te-termination-point-augment

grouping te-termination-point-config {
 description
 "TE termination point configuration grouping.";
 leaf admin-status {
 type te-types:te-admin-status;
 description
 "The administrative state of the link termination point.";
 }
 leaf name {
 type string;
 description
 "A descriptive name for the link termination point.";
 }
 uses interface-switching-capability-list;
 leaf inter-domain-plug-id {
 type binary;
 description
 "A topology-wide unique number that identifies on the
 network a connectivity supporting a given inter-domain
 TE link. This is more flexible alternative to specifying
 remote-te-node-id and remote-te-link-tp-id on a TE link,
 when the provider does not know remote-te-node-id and
 remote-te-link-tp-id or need to give client the
 flexibility to mix-n-match multiple topologies.";
 }
 leaf-list inter-layer-lock-id {
 type uint32;
 description
 "Inter layer lock ID, used for path computation in a TE
 topology covering multiple layers or multiple regions.";
 reference
 "RFC 5212: Requirements for GMPLS-Based Multi-Region and
 Multi-Layer Networks (MRN/MLN).
 RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
```



```
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";
 }
} // te-termination-point-config

grouping te-topologies-augment {
 description
 "Augmentation for TE topologies.";
 container te {
 presence "TE support.";
 description
 "Indicates TE support.";

 container templates {
 description
 "Configuration parameters for templates used for TE
 topology.";

 list node-template {
 if-feature template;
 key "name";
 leaf name {
 type te-types:te-template-name;
 description
 "The name to identify a TE node template.";
 }
 description
 "The list of TE node templates used to define sharable
 and reusable TE node attributes.";
 uses template-attributes;
 uses te-node-config-attributes-template;
 } // node-template

 list link-template {
 if-feature template;
 key "name";
 leaf name {
 type te-types:te-template-name;
 description
 "The name to identify a TE link template.";
 }
 description

```

```
 "The list of TE link templates used to define sharable
 and reusable TE link attributes.";
 uses template-attributes;
 uses te-link-config-attributes;
 } // link-template
} // templates
} // te
} // te-topologies-augment

grouping te-topology-augment {
 description
 "Augmentation for TE topology.";
 uses te-types:te-topology-identifier;

 container te {
 must "../te-topology-identifier/provider-id"
 + " and ../te-topology-identifier/client-id"
 + " and ../te-topology-identifier/topology-id";
 presence "TE support.";
 description
 "Indicates TE support.";

 uses te-topology-config;
 uses geolocation-container;
 } // te
} // te-topology-augment

grouping te-topology-config {
 description
 "TE topology configuration grouping.";
 leaf name {
 type string;
 description
 "Name of the TE topology. This attribute is optional and can
 be specified by the operator to describe the TE topology,
 which can be useful when network-id is not descriptive
 and not modifiable because of being generated by the
 system.";
 }
 leaf preference {
 type uint8 {
```

```
 range "1..255";
 }
 description
 "Specifies a preference for this topology. A lower number
 indicates a higher preference.";
}
leaf optimization-criterion {
 type identityref {
 base te-types:objective-function-type;
 }
 description
 "Optimization criterion applied to this topology.";
 reference
 "RFC 3272: Overview and Principles of Internet Traffic
 Engineering.";
}
list nsrlg {
 if-feature nsrlg;
 key "id";
 description
 "List of NSRLGs (Not Sharing Risk Link Groups).";
 reference
 "RFC 4872: RSVP-TE Extensions in Support of End-to-End
 Generalized Multi-Protocol Label Switching (GMPLS)
 Recovery";
 leaf id {
 type uint32;
 description
 "Identify the NSRLG entry.";
 }
 leaf disjointness {
 type te-types:te-path-disjointness;
 description
 "The type of resource disjointness.";
 }
} // nsrlg
} // te-topology-config

grouping template-attributes {
 description
 "Common attributes for all templates.";
```

```
leaf priority {
 type uint16;
 description
 "The preference value to resolve conflicts between different
 templates. When two or more templates specify values for
 one configuration attribute, the value from the template
 with the highest priority is used.
 A lower number indicates a higher priority. The highest
 priority is 0.";
}
leaf reference-change-policy {
 type enumeration {
 enum no-action {
 description
 "When an attribute changes in this template, the
 configuration node referring to this template does
 not take any action.";
 }
 enum not-allowed {
 description
 "When any configuration object has a reference to this
 template, changing this template is not allowed.";
 }
 enum cascade {
 description
 "When an attribute changes in this template, the
 configuration object referring to this template applies
 the new attribute value to the corresponding
 configuration.";
 }
 }
 description
 "This attribute specifies the action taken to a configuration
 node that has a reference to this template.";
}
} // template-attributes

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
```

```
 description
 "Introduce new network type for TE topology.";
 container te-topology {
 presence "Indicates TE topology.";
 description
 "Its presence identifies the TE topology type.";
 }
 }

 augment "/nw:networks" {
 description
 "Augmentation parameters for TE topologies.";
 uses te-topologies-augment;
 }

 augment "/nw:networks/nw:network" {
 when "nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE topology.";
 uses te-topology-augment;
 }

 augment "/nw:networks/nw:network/nw:node" {
 when "../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at node level.";
 leaf te-node-id {
 type te-types:te-node-id;
 description
 "The identifier of a node in the TE topology.
 A node is specific to a topology to which it belongs.";
 }
 }
 container te {
```

```
 must "../te-node-id" {
 description
 "te-node-id is mandatory.";
 }
 must "count(..../nw:supporting-node)<=1" {
 description
 "For a node in a TE topology, there cannot be more
 than 1 supporting node. If multiple nodes are abstracted,
 the underlay-topology is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";
 uses te-node-augment;
 } // te
}

augment "/nw:networks/nw:network/nt:link" {
 when "../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at link level.";
 container te {
 must "count(..../nt:supporting-link)<=1" {
 description
 "For a link in a TE topology, there cannot be more
 than 1 supporting link. If one or more link paths are
 abstracted, the underlay is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";
 uses te-link-augment;
 } // te
}

augment "/nw:networks/nw:network/nw:node/"
 + "nt:termination-point" {
```

```
when "../../../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
}
description
 "Configuration parameters for TE at termination point level.";
uses te-termination-point-augment;
}

augment
 "/nw:networks/nw:network/nt:link/te/bundle-stack-level/"
+ "bundle/bundled-links/bundled-link" {
 when "../../../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE link bundled link.";
 leaf src-tp-ref {
 type leafref {
 path "../../../nw:node[nw:node-id = "
 + "current()/../../../../nt:source/"
 + "nt:source-node]/"
 + "nt:termination-point/nt:tp-id";
 require-instance true;
 }
 description
 "Reference to another TE termination point on the
 same source node.";
 }
 leaf des-tp-ref {
 type leafref {
 path "../../../nw:node[nw:node-id = "
 + "current()/../../../../nt:destination/"
 + "nt:dest-node]/"
 + "nt:termination-point/nt:tp-id";
 require-instance true;
 }
 description

```

```
 "Reference to another TE termination point on the
 same destination node.";
 }
}

augment
 "/nw:networks/nw:network/nw:node/te/"
+ "information-source-entry/connectivity-matrices/"
+ "connectivity-matrix" {
 when "../..../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE node connectivity-matrix.";
 uses te-node-connectivity-matrix-attributes;
}

augment
 "/nw:networks/nw:network/nw:node/te/te-node-attributes/"
+ "connectivity-matrices/connectivity-matrix" {
 when "../..../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE node connectivity-matrix.";
 uses te-node-connectivity-matrix-attributes;
}

augment
 "/nw:networks/nw:network/nw:node/te/"
+ "tunnel-termination-point/local-link-connectivities" {
 when "../..../nw:network-types/tet:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
```



```
 "Augment TE node tunnel termination point LLCs
 (Local Link Connectivities).";
 uses te-node-tunnel-termination-point-llc-list;
 }
}
<CODE ENDS>
```

## 8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /nw:networks/nw:network/nw:network-types/tet:te-topology  
This subtree specifies the TE topology type. Modifying the configurations can make TE topology type invalid. By such modifications, a malicious attacker may disable the TE capabilities on the related networks and cause traffic disrupted or misrouted.
- o /nw:networks/tet:te  
This subtree specifies the TE node templates and TE link templates. Modifying the configurations in this subtree will change the related future TE configurations. By such modifications, a malicious attacker may change the TE capabilities scheduled at a future time, to cause traffic disrupted or misrouted.

- o /nw:networks/nw:network  
This subtree specifies the topology-wide configurations, including the TE topology ID and topology-wide policies. Modifying the configurations in this subtree can add, remove, or modify TE topologies. By adding a TE topology, a malicious attacker may create an unauthorized traffic network. By removing or modifying a TE topology, a malicious attacker may cause traffic disabled or misrouted in the specified TE topology. Such traffic changes may also affect the traffic in the connected TE topologies.
- o /nw:networks/nw:network/nw:node  
This subtree specifies the configurations for TE nodes. Modifying the configurations in this subtree can add, remove, or modify TE nodes. By adding a TE node, a malicious attacker may create an unauthorized traffic path. By removing or modifying a TE node, a malicious attacker may cause traffic disabled or misrouted in the specified TE node. Such traffic changes may also affect the traffic on the surrounding TE nodes and TE links in this TE topology and the connected TE topologies.
- o /nw:networks/nw:network/nt:link/tet:te  
This subtree specifies the configurations for TE links. Modifying the configurations in this subtree can add, remove, or modify TE links. By adding a TE link, a malicious attacker may create an unauthorized traffic path. By removing or modifying a TE link, a malicious attacker may cause traffic disabled or misrouted on the specified TE link. Such traffic changes may also affect the traffic on the surrounding TE nodes and TE links in this TE topology and the connected TE topologies.
- o /nw:networks/nw:network/nw:node/nt:termination-point  
This subtree specifies the configurations of TE link termination points. Modifying the configurations in this subtree can add, remove, or modify TE link termination points. By adding a TE link termination point, a malicious attacker may create an unauthorized traffic path. By removing or modifying a TE link termination point, a malicious attacker may cause traffic disabled or misrouted on the specified TE link termination point. Such traffic changes may also affect the traffic on the surrounding TE nodes and TE links in this TE topology and the connected TE topologies.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

- o /nw:networks/nw:network/nw:network-types/tet:te-topology  
Unauthorized access to this subtree can disclose the TE topology type.
- o /nw:networks/tet:te  
Unauthorized access to this subtree can disclose the TE node templates and TE link templates.
- o /nw:networks/nw:network  
Unauthorized access to this subtree can disclose the topology-wide configurations, including the TE topology ID, the topology-wide policies, and the topology geolocation.
- o /nw:networks/nw:network/nw:node  
Unauthorized access to this subtree can disclose the operational state information of TE nodes.
- o /nw:networks/nw:network/nt:link/tet:te  
Unauthorized access to this subtree can disclose the operational state information of TE links.
- o /nw:networks/nw:network/nw:node/nt:termination-point  
Unauthorized access to this subtree can disclose the operational state information of TE link termination points.

## 9. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-state  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC7950].

name: ietf-te-topology  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology  
prefix: tet  
reference: RFC XXXX

name: ietf-te-topology-state  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology-state  
prefix: tet-s  
reference: RFC XXXX

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <<https://www.rfc-editor.org/info/rfc7926>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [I-D.ietf-teas-yang-te-types]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and  
I. Bryskin, "Traffic Engineering Common YANG Types",  
draft-ietf-teas-yang-te-types-08 (work in progress),  
April 2019.

## 10.2. Informative References

- [G.709] ITU-T, "Interfaces for the optical transport network", ITU-T Recommendation G.709, June 2016.
- [G.805] ITU-T, "Generic functional architecture of transport networks", ITU-T Recommendation G.805, March 2000.
- [G.872] ITU-T, "Architecture of optical transport networks", ITU-T Recommendation G.872, January 2017.
- [G.8080] ITU-T, "Architecture for the automatically switched optical network", ITU-T Recommendation G.8080, February 2012.

- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <<https://www.rfc-editor.org/info/rfc1195>>.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <<https://www.rfc-editor.org/info/rfc2702>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <<https://www.rfc-editor.org/info/rfc3272>>.
- [RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, DOI 10.17487/RFC3471, January 2003, <<https://www.rfc-editor.org/info/rfc3471>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC3785] Le Faucheur, F., Uppili, R., Vedrenne, A., Merckx, P., and T. Telkamp, "Use of Interior Gateway Protocol (IGP) Metric as a second MPLS Traffic Engineering (TE) Metric", BCP 87, RFC 3785, DOI 10.17487/RFC3785, May 2004, <<https://www.rfc-editor.org/info/rfc3785>>.
- [RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling in MPLS Traffic Engineering (TE)", RFC 4201, DOI 10.17487/RFC4201, October 2005, <<https://www.rfc-editor.org/info/rfc4201>>.
- [RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005, <<https://www.rfc-editor.org/info/rfc4202>>.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching

- (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <<https://www.rfc-editor.org/info/rfc4203>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<https://www.rfc-editor.org/info/rfc4872>>.
- [RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <<https://www.rfc-editor.org/info/rfc5152>>.
- [RFC5212] Shiimoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, DOI 10.17487/RFC5212, July 2008, <<https://www.rfc-editor.org/info/rfc5212>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316, December 2008, <<https://www.rfc-editor.org/info/rfc5316>>.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed., "Traffic Engineering Extensions to OSPF Version 3", RFC 5329, DOI 10.17487/RFC5329, September 2008, <<https://www.rfc-editor.org/info/rfc5329>>.
- [RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392, January 2009, <<https://www.rfc-editor.org/info/rfc5392>>.
- [RFC6001] Papadimitriou, D., Vigoureux, M., Shiimoto, K., Brungard,

- D., and JL. Le Roux, "Generalized MPLS (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)", RFC 6001, DOI 10.17487/RFC6001, October 2010, <<https://www.rfc-editor.org/info/rfc6001>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC7579] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and J. Han, "General Network Element Constraint Encoding for GMPLS-Controlled Networks", RFC 7579, DOI 10.17487/RFC7579, June 2015, <<https://www.rfc-editor.org/info/rfc7579>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [I-D.ietf-netconf-subscribed-notifications]  
Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Customized Subscriptions to a Publisher's Event Streams", draft-ietf-netconf-subscribed-notifications-23 (work in progress), February 2019.
- [I-D.ietf-netconf-yang-push]  
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-22 (work in progress), February 2019.
- [I-D.liu-netmod-yang-schedule]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule-05 (work in progress),



March 2018.

[I-D.ietf-ccamp-wson-yang]

Lee, Y., Dhody, D., Zhang, X., Guo, A., Lopezalvarez, V., King, D., Yoon, B., and R. Vilata, "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang-20 (work in progress), March 2019.

[I-D.ietf-ccamp-otn-topo-yang]

zhenghaomian@huawei.com, z., Guo, A., Busi, I., Sharma, A., Liu, X., Belotti, S., Xu, Y., Wang, L., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-06 (work in progress), February 2019.

[I-D.ietf-teas-yang-l3-te-topo]

Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-04 (work in progress), March 2019.

[I-D.ietf-teas-te-topo-and-tunnel-modeling]

Bryskin, I., Beeram, V., Saad, T., and X. Liu, "TE Topology and Tunnel Modeling for Transport Networks", draft-ietf-teas-te-topo-and-tunnel-modeling-03 (work in progress), October 2018.

## 11. Acknowledgments

The authors would like to thank Lou Berger, Sue Hares, Mazen Khaddam, Cyril Margaria and Zafar Ali for participating in design discussions and providing valuable insights.

## Appendix A. Complete Model Tree Structure

```

module: ietf-te-topology
 augment /nw:networks/nw:network/nw:network-types:
 +--rw te-topology!
 augment /nw:networks:
 +--rw te!
 +--rw templates
 +--rw node-template* [name] {template}?
 +--rw name
 | te-types:te-template-name
 +--rw priority? uint16
 +--rw reference-change-policy? enumeration
 +--rw te-node-attributes
 +--rw admin-status? te-types:te-admin-status
 +--rw domain-id? uint32
 +--rw is-abstract? empty
 +--rw name? string
 +--rw signaling-address* inet:ip-address
 +--rw underlay-topology {te-topology-hierarchy}?
 +--rw network-ref?
 | -> /nw:networks/network/network-id
 +--rw link-template* [name] {template}?
 +--rw name
 | te-types:te-template-name
 +--rw priority? uint16
 +--rw reference-change-policy? enumeration
 +--rw te-link-attributes
 +--rw access-type?
 | te-types:te-link-access-type
 +--rw external-domain
 | +--rw network-ref?
 | | -> /nw:networks/network/network-id
 | +--rw remote-te-node-id? te-types:te-node-id
 | +--rw remote-te-link-tp-id? te-types:te-tp-id
 +--rw is-abstract? empty
 +--rw name? string
 +--rw underlay {te-topology-hierarchy}?
 +--rw enabled? boolean
 +--rw primary-path
 | +--rw network-ref?

```

```

| -> /nw:networks/network/network-id
+--rw path-element* [path-element-id]
| +--rw path-element-id uint32
| +--rw (type)?
| +--:(numbered-node-hop)
| +--rw numbered-node-hop
| +--rw node-id te-node-id
| +--rw hop-type? te-hop-type
| +--:(numbered-link-hop)
| +--rw numbered-link-hop
| +--rw link-tp-id te-tp-id
| +--rw hop-type? te-hop-type
| +--rw direction?
| te-link-direction
| +--:(unnumbered-link-hop)
| +--rw unnumbered-link-hop
| +--rw link-tp-id te-tp-id
| +--rw node-id te-node-id
| +--rw hop-type? te-hop-type
| +--rw direction?
| te-link-direction
| +--:(as-number)
| +--rw as-number-hop
| +--rw as-number inet:as-number
| +--rw hop-type? te-hop-type
| +--:(label)
| +--rw label-hop
| +--rw te-label
| +--rw (technology)?
| +--:(generic)
| +--rw generic?
| rt-
types:generalized-label
| +--rw direction?
| te-label-direction
+--rw backup-path* [index]
| +--rw index uint32
| +--rw network-ref?
| -> /nw:networks/network/network-id
+--rw path-element* [path-element-id]
| +--rw path-element-id uint32

```

```

+--rw (type)?
+--:(numbered-node-hop)
| +--rw numbered-node-hop
| +--rw node-id te-node-id
| +--rw hop-type? te-hop-type
+--:(numbered-link-hop)
| +--rw numbered-link-hop
| +--rw link-tp-id te-tp-id
| +--rw hop-type? te-hop-type
| +--rw direction?
| te-link-direction
+--:(unnumbered-link-hop)
| +--rw unnumbered-link-hop
| +--rw link-tp-id te-tp-id
| +--rw node-id te-node-id
| +--rw hop-type? te-hop-type
| +--rw direction?
| te-link-direction
+--:(as-number)
| +--rw as-number-hop
| +--rw as-number inet:as-number
| +--rw hop-type? te-hop-type
+--:(label)
| +--rw label-hop
| +--rw te-label
| +--rw (technology)?
| +--:(generic)
| +--rw generic?
| rt-
types:generalized-label
| +--rw direction?
| te-label-direction
+--rw protection-type? identityref
+--rw tunnel-termination-points
| +--rw source? binary
| +--rw destination? binary
+--rw tunnels
| +--rw sharing? boolean
| +--rw tunnel* [tunnel-name]
| +--rw tunnel-name string
| +--rw sharing? boolean

```

```

+--rw admin-status?
| te-types:te-admin-status
+--rw link-index? uint64
+--rw administrative-group?
| te-types:admin-groups
+--rw interface-switching-capability*
| [switching-capability encoding]
| +--rw switching-capability identityref
| +--rw encoding identityref
| +--rw max-lsp-bandwidth* [priority]
| +--rw priority uint8
| +--rw te-bandwidth
| +--rw (technology)?
| +--:(generic)
| +--rw generic? te-bandwidth
+--rw label-restrictions
| +--rw label-restriction* [index]
| +--rw restriction? enumeration
| +--rw index uint32
| +--rw label-start
| +--rw te-label
| +--rw (technology)?
| +--:(generic)
| +--rw generic?
| rt-types:generalized-label
| +--rw direction? te-label-direction
+--rw label-end
| +--rw te-label
| +--rw (technology)?
| +--:(generic)
| +--rw generic?
| rt-types:generalized-label
| +--rw direction? te-label-direction
+--rw label-step
| +--rw (technology)?
| +--:(generic)
| +--rw generic? int32
| +--rw range-bitmap? yang:hex-string
+--rw link-protection-type? identityref
+--rw max-link-bandwidth
| +--rw te-bandwidth

```

```

 | +---rw (technology)?
 | +---:(generic)
 | +---rw generic? te-bandwidth
+---rw max-resv-link-bandwidth
 | +---rw te-bandwidth
 | +---rw (technology)?
 | +---:(generic)
 | +---rw generic? te-bandwidth
+---rw unreserved-bandwidth* [priority]
 | +---rw priority uint8
 | +---rw te-bandwidth
 | +---rw (technology)?
 | +---:(generic)
 | +---rw generic? te-bandwidth
+---rw te-default-metric? uint32
+---rw te-delay-metric? uint32
+---rw te-igp-metric? uint32
+---rw te-srlgs
 | +---rw value* te-types:srlg
+---rw te-nsrlgs {nsrlg}?
 +---rw id* uint32
augment /nw:networks/nw:network:
+---rw te-topology-identifier
 | +---rw provider-id? te-global-id
 | +---rw client-id? te-global-id
 | +---rw topology-id? te-topology-id
+---rw te!
 +---rw name? string
 +---rw preference? uint8
 +---rw optimization-criterion? identityref
 +---rw nsrlg* [id] {nsrlg}?
 | +---rw id uint32
 | +---rw disjointness? te-types:te-path-disjointness
+---ro geolocation
 +---ro altitude? int64
 +---ro latitude? geographic-coordinate-degree
 +---ro longitude? geographic-coordinate-degree
augment /nw:networks/nw:network/nw:node:
+---rw te-node-id? te-types:te-node-id
+---rw te!
 +---rw te-node-template*

```

```

| -> ../../../../te/templates/node-template/name
| {template}?
+--rw te-node-attributes
| +--rw admin-status? te-types:te-admin-status
| +--rw connectivity-matrices
| | +--rw number-of-entries? uint16
| | +--rw label-restrictions
| | | +--rw label-restriction* [index]
| | | | +--rw restriction? enumeration
| | | | +--rw index uint32
| | | | +--rw label-start
| | | | | +--rw te-label
| | | | | | +--rw (technology)?
| | | | | | | +--:(generic)
| | | | | | | | +--rw generic?
| | | | | | | | | rt-types:generalized-label
| | | | | | | | | +--rw direction? te-label-direction
| | | | +--rw label-end
| | | | | +--rw te-label
| | | | | | +--rw (technology)?
| | | | | | | +--:(generic)
| | | | | | | | +--rw generic?
| | | | | | | | | rt-types:generalized-label
| | | | | | | | | +--rw direction? te-label-direction
| | | | +--rw label-step
| | | | | +--rw (technology)?
| | | | | | +--:(generic)
| | | | | | | +--rw generic? int32
| | | | +--rw range-bitmap? yang:hex-string
| | +--rw is-allowed? boolean
| +--rw underlay {te-topology-hierarchy}?
| | +--rw enabled? boolean
| | +--rw primary-path
| | | +--rw network-ref?
| | | | -> /nw:networks/network/network-id
| | | +--rw path-element* [path-element-id]
| | | | +--rw path-element-id uint32
| | | | +--rw (type)?
| | | | | +--:(numbered-node-hop)
| | | | | | +--rw numbered-node-hop
| | | | | | | +--rw node-id te-node-id

```







```

| +---rw metric-type identityref
| +---rw upper-bound? uint64
+---rw path-affinities-values
| +---rw path-affinities-value* [usage]
| +---rw usage identityref
| +---rw value? admin-groups
+---rw path-affinity-names
| +---rw path-affinity-name* [usage]
| +---rw usage identityref
| +---rw affinity-name* [name]
| +---rw name string
+---rw path-srlgs-lists
| +---rw path-srlgs-list* [usage]
| +---rw usage identityref
| +---rw values* srlg
+---rw path-srlgs-names
| +---rw path-srlgs-name* [usage]
| +---rw usage identityref
| +---rw names* string
+---rw disjointness? te-path-disjointness
+---rw optimizations
+---rw (algorithm)?
| +---:(metric) {path-optimization-metric}?
| | +---rw optimization-metric* [metric-type]
| | | +---rw metric-type
| | | | identityref
| | +---rw weight?
| | | uint8
| | +---rw explicit-route-exclude-objects
| | | +---rw route-object-exclude-object*
| | | | [index]
| | | +---rw index
| | | | uint32
| | +---rw (type)?
| | | +---:(numbered-node-hop)
| | | | +---rw numbered-node-hop
| | | | | +---rw node-id te-node-id
| | | | | +---rw hop-type? te-hop-type
| | | +---:(numbered-link-hop)
| | | | +---rw numbered-link-hop
| | | | | +---rw link-tp-id te-tp-id

```

```

+---rw hop-type?
| te-hop-type
+---rw direction?
| te-link-direction
+---:(unnumbered-link-hop)
+---rw unnumbered-link-hop
+---rw link-tp-id te-tp-id
+---rw node-id
| te-node-id
+---rw hop-type?
| te-hop-type
+---rw direction?
| te-link-direction
+---:(as-number)
+---rw as-number-hop
+---rw as-number
| inet:as-number
+---rw hop-type?
| te-hop-type
+---:(label)
+---rw label-hop
+---rw te-label
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic?
| | rt-
types:generalized-label
| +---rw direction?
| | te-label-direction
+---:(srlg)
+---rw srlg
| +---rw srlg? uint32
+---rw explicit-route-include-objects
+---rw route-object-include-object*
| [index]
+---rw index
| uint32
+---rw (type)?
| +---:(numbered-node-hop)
| | +---rw numbered-node-hop
| | +---rw node-id te-node-id

```

```

+---rw hop-type? te-hop-type
+---:(numbered-link-hop)
| +---rw numbered-link-hop
| +---rw link-tp-id te-tp-id
| +---rw hop-type?
| |
| te-hop-type
| +---rw direction?
| te-link-direction
+---:(unnumbered-link-hop)
| +---rw unnumbered-link-hop
| +---rw link-tp-id te-tp-id
| +---rw node-id
| |
| te-node-id
| +---rw hop-type?
| |
| te-hop-type
| +---rw direction?
| te-link-direction
+---:(as-number)
| +---rw as-number-hop
| +---rw as-number
| |
| inet:as-number
| +---rw hop-type?
| te-hop-type
+---:(label)
| +---rw label-hop
| +---rw te-label
| +---rw (technology)?
| +---:(generic)
| +---rw generic?
| rt-
types:generalized-label
|
| +---rw direction?
| te-label-direction
+---rw tiebreakers
+---rw tiebreaker* [tiebreaker-type]
+---rw tiebreaker-type identityref
+---:(objective-function)
| {path-optimization-objective-function}?
+---rw objective-function
+---rw objective-function-type? identityref
+---ro path-properties

```

```

+--ro path-metric* [metric-type]
| +--ro metric-type identityref
| +--ro accumulative-value? uint64
+--ro path-affinities-values
| +--ro path-affinities-value* [usage]
| +--ro usage identityref
| +--ro value? admin-groups
+--ro path-affinity-names
| +--ro path-affinity-name* [usage]
| +--ro usage identityref
| +--ro affinity-name* [name]
| +--ro name string
+--ro path-srlgs-lists
| +--ro path-srlgs-list* [usage]
| +--ro usage identityref
| +--ro values* srlg
+--ro path-srlgs-names
| +--ro path-srlgs-name* [usage]
| +--ro usage identityref
| +--ro names* string
+--ro path-route-objects
| +--ro path-route-object* [index]
| +--ro index uint32
| +--ro (type)?
| +--:(numbered-node-hop)
| +--ro numbered-node-hop
| +--ro node-id te-node-id
| +--ro hop-type? te-hop-type
| +--:(numbered-link-hop)
| +--ro numbered-link-hop
| +--ro link-tp-id te-tp-id
| +--ro hop-type? te-hop-type
| +--ro direction? te-link-direction
| +--:(unnumbered-link-hop)
| +--ro unnumbered-link-hop
| +--ro link-tp-id te-tp-id
| +--ro node-id te-node-id
| +--ro hop-type? te-hop-type
| +--ro direction? te-link-direction
| +--:(as-number)
| +--ro as-number-hop

```

|       |  |  |  |                            |                          |                       |
|-------|--|--|--|----------------------------|--------------------------|-----------------------|
|       |  |  |  |                            | +--ro as-number          | inet:as-number        |
|       |  |  |  |                            | +--ro hop-type?          | te-hop-type           |
|       |  |  |  | +---:(label)               |                          |                       |
|       |  |  |  |                            | +--ro label-hop          |                       |
|       |  |  |  |                            | +--ro te-label           |                       |
|       |  |  |  |                            | +--ro (technology)?      |                       |
|       |  |  |  |                            | +---:(generic)           |                       |
|       |  |  |  |                            | +--ro generic?           |                       |
|       |  |  |  |                            |                          | rt-types:generalized- |
| label |  |  |  |                            |                          |                       |
|       |  |  |  |                            | +--ro direction?         |                       |
|       |  |  |  |                            |                          | te-label-direction    |
|       |  |  |  | +--rw connectivity-matrix* | [id]                     |                       |
|       |  |  |  | +--rw id                   |                          | uint32                |
|       |  |  |  | +--rw from                 |                          |                       |
|       |  |  |  |                            | +--rw tp-ref?            | leafref               |
|       |  |  |  |                            | +--rw label-restrictions |                       |
|       |  |  |  |                            | +--rw label-restriction* | [index]               |
|       |  |  |  |                            | +--rw restriction?       | enumeration           |
|       |  |  |  |                            | +--rw index              | uint32                |
|       |  |  |  |                            | +--rw label-start        |                       |
|       |  |  |  |                            | +--rw te-label           |                       |
|       |  |  |  |                            | +--rw (technology)?      |                       |
|       |  |  |  |                            | +---:(generic)           |                       |
|       |  |  |  |                            | +--rw generic?           |                       |
|       |  |  |  |                            |                          | rt-types:generalized- |
| label |  |  |  |                            |                          |                       |
|       |  |  |  |                            | +--rw direction?         |                       |
|       |  |  |  |                            |                          | te-label-direction    |
|       |  |  |  | +--rw label-end            |                          |                       |
|       |  |  |  |                            | +--rw te-label           |                       |
|       |  |  |  |                            | +--rw (technology)?      |                       |
|       |  |  |  |                            | +---:(generic)           |                       |
|       |  |  |  |                            | +--rw generic?           |                       |
|       |  |  |  |                            |                          | rt-types:generalized- |
| label |  |  |  |                            |                          |                       |
|       |  |  |  |                            | +--rw direction?         |                       |
|       |  |  |  |                            |                          | te-label-direction    |
|       |  |  |  | +--rw label-step           |                          |                       |
|       |  |  |  |                            | +--rw (technology)?      |                       |
|       |  |  |  |                            | +---:(generic)           |                       |



```

+--rw numbered-node-hop
+---rw node-id te-node-id
+---rw hop-type? te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id te-tp-id
+---rw hop-type? te-hop-type
+---rw direction?
+---rw te-link-direction
+---:(unnumbered-link-hop)
+---rw unnumbered-link-hop
+---rw link-tp-id te-tp-id
+---rw node-id te-node-id
+---rw hop-type? te-hop-type
+---rw direction?
+---rw te-link-direction
+---:(as-number)
+---rw as-number-hop
+---rw as-number inet:as-number
+---rw hop-type? te-hop-type
+---:(label)
+---rw label-hop
+---rw te-label
+---rw (technology)?
+---:(generic)
+---rw generic?
+---rw rt-
types:generalized-label
+---rw direction?
+---rw te-label-direction
+---rw backup-path* [index]
+---rw index uint32
+---rw network-ref?
+---rw -> /nw:networks/network/network-id
+---rw path-element* [path-element-id]
+---rw path-element-id uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id te-node-id
+---rw hop-type? te-hop-type

```





```

+--rw link-protection? identityref
+--rw setup-priority? uint8
+--rw hold-priority? uint8
+--rw signaling-type? identityref
+--rw path-metric-bounds
| +--rw path-metric-bound* [metric-type]
| | +--rw metric-type identityref
| | +--rw upper-bound? uint64
+--rw path-affinities-values
| +--rw path-affinities-value* [usage]
| | +--rw usage identityref
| | +--rw value? admin-groups
+--rw path-affinity-names
| +--rw path-affinity-name* [usage]
| | +--rw usage identityref
| | +--rw affinity-name* [name]
| | | +--rw name string
+--rw path-srlgs-lists
| +--rw path-srlgs-list* [usage]
| | +--rw usage identityref
| | +--rw values* srlg
+--rw path-srlgs-names
| +--rw path-srlgs-name* [usage]
| | +--rw usage identityref
| | +--rw names* string
+--rw disjointness?
| te-path-disjointness
+--rw optimizations
| +--rw (algorithm)?
| | +--:(metric) {path-optimization-metric}?
| | | +--rw optimization-metric* [metric-type]
| | | | +--rw metric-type
| | | | | identityref
| | | | +--rw weight?
| | | | | uint8
| | | +--rw explicit-route-exclude-objects
| | | | +--rw route-object-exclude-object*
| | | | | [index]
| | | | | +--rw index
| | | | | | uint32
| | | +--rw (type)?

```

|                         |  |  |  |  |  |                            |
|-------------------------|--|--|--|--|--|----------------------------|
|                         |  |  |  |  |  | +--:(numbered-node-hop)    |
|                         |  |  |  |  |  | +---rw numbered-node-hop   |
|                         |  |  |  |  |  | +---rw node-id             |
|                         |  |  |  |  |  | te-node-id                 |
|                         |  |  |  |  |  | +---rw hop-type?           |
|                         |  |  |  |  |  | te-hop-type                |
|                         |  |  |  |  |  | +--:(numbered-link-hop)    |
|                         |  |  |  |  |  | +---rw numbered-link-hop   |
|                         |  |  |  |  |  | +---rw link-tp-id          |
|                         |  |  |  |  |  | te-tp-id                   |
|                         |  |  |  |  |  | +---rw hop-type?           |
|                         |  |  |  |  |  | te-hop-type                |
|                         |  |  |  |  |  | +---rw direction?          |
|                         |  |  |  |  |  | te-link-direction          |
|                         |  |  |  |  |  | +--:(unnumbered-link-hop)  |
|                         |  |  |  |  |  | +---rw unnumbered-link-hop |
|                         |  |  |  |  |  | +---rw link-tp-id          |
|                         |  |  |  |  |  | te-tp-id                   |
|                         |  |  |  |  |  | +---rw node-id             |
|                         |  |  |  |  |  | te-node-id                 |
|                         |  |  |  |  |  | +---rw hop-type?           |
|                         |  |  |  |  |  | te-hop-type                |
|                         |  |  |  |  |  | +---rw direction?          |
|                         |  |  |  |  |  | te-link-direction          |
|                         |  |  |  |  |  | +--:(as-number)            |
|                         |  |  |  |  |  | +---rw as-number-hop       |
|                         |  |  |  |  |  | +---rw as-number           |
|                         |  |  |  |  |  | inet:as-number             |
|                         |  |  |  |  |  | +---rw hop-type?           |
|                         |  |  |  |  |  | te-hop-type                |
|                         |  |  |  |  |  | +--:(label)                |
|                         |  |  |  |  |  | +---rw label-hop           |
|                         |  |  |  |  |  | +---rw te-label            |
|                         |  |  |  |  |  | +---rw (technology)?       |
|                         |  |  |  |  |  | +--:(generic)              |
|                         |  |  |  |  |  | +---rw generic?            |
|                         |  |  |  |  |  | rt-                        |
| types:generalized-label |  |  |  |  |  |                            |
|                         |  |  |  |  |  | +---rw direction?          |
| direction               |  |  |  |  |  | te-label-                  |

```

+--:(srlg)
 +--rw srlg
 +--rw srlg? uint32
+--rw explicit-route-include-objects
 +--rw route-object-include-object*
 [index]
 +--rw index
 | uint32
 +--rw (type)?
 +--:(numbered-node-hop)
 +--rw numbered-node-hop
 +--rw node-id
 | te-node-id
 +--rw hop-type?
 | te-hop-type
 +--:(numbered-link-hop)
 +--rw numbered-link-hop
 +--rw link-tp-id
 | te-tp-id
 +--rw hop-type?
 | te-hop-type
 +--rw direction?
 | te-link-direction
 +--:(unnumbered-link-hop)
 +--rw unnumbered-link-hop
 +--rw link-tp-id
 | te-tp-id
 +--rw node-id
 | te-node-id
 +--rw hop-type?
 | te-hop-type
 +--rw direction?
 | te-link-direction
 +--:(as-number)
 +--rw as-number-hop
 +--rw as-number
 | inet:as-number
 +--rw hop-type?
 | te-hop-type
 +--:(label)
 +--rw label-hop

```

```

types:generalized-label
direction
function}?

+--rw te-label
+--rw (technology)?
+--:(generic)
+--rw generic?
rt-

+--rw direction?
te-label-

+--rw tiebreakers
+--rw tiebreaker* [tiebreaker-type]
+--rw tiebreaker-type identityref
+--:(objective-function)
{path-optimization-objective-

+--rw objective-function
+--rw objective-function-type?
identityref

+--ro path-properties
+--ro path-metric* [metric-type]
| +--ro metric-type identityref
| +--ro accumulative-value? uint64
+--ro path-affinities-values
+--ro path-affinities-value* [usage]
+--ro usage identityref
+--ro value? admin-groups
+--ro path-affinity-names
+--ro path-affinity-name* [usage]
+--ro usage identityref
+--ro affinity-name* [name]
+--ro name string
+--ro path-srlgs-lists
+--ro path-srlgs-list* [usage]
+--ro usage identityref
+--ro values* srlg
+--ro path-srlgs-names
+--ro path-srlgs-name* [usage]
+--ro usage identityref
+--ro names* string
+--ro path-route-objects
+--ro path-route-object* [index]

```

```

+--ro index uint32
+--ro (type)?
 +--:(numbered-node-hop)
 +--ro numbered-node-hop
 +--ro node-id te-node-id
 +--ro hop-type? te-hop-type
 +--:(numbered-link-hop)
 +--ro numbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro hop-type? te-hop-type
 +--ro direction?
 te-link-direction
 +--:(unnumbered-link-hop)
 +--ro unnumbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro node-id te-node-id
 +--ro hop-type? te-hop-type
 +--ro direction?
 te-link-direction
 +--:(as-number)
 +--ro as-number-hop
 +--ro as-number inet:as-number
 +--ro hop-type? te-hop-type
 +--:(label)
 +--ro label-hop
 +--ro te-label
 +--ro (technology)?
 +--:(generic)
 +--ro generic?
 rt-
 +--ro direction?
 te-label-direction
types:generalized-label
+--rw domain-id? uint32
+--rw is-abstract? empty
+--rw name? string
+--rw signaling-address* inet:ip-address
+--rw underlay-topology {te-topology-hierarchy}?
 +--rw network-ref? -> /nw:networks/network/network-id
+--ro oper-status? te-types:te-oper-status
+--ro geolocation

```

```

| +---ro altitude? int64
| +---ro latitude? geographic-coordinate-degree
| +---ro longitude? geographic-coordinate-degree
+---ro is-multi-access-dr? empty
+---ro information-source? te-info-source
+---ro information-source-instance? string
+---ro information-source-state
| +---ro credibility-preference? uint16
| +---ro logical-network-element? string
| +---ro network-instance? string
| +---ro topology
| +---ro node-ref? leafref
| +---ro network-ref? -> /nw:networks/network/network-id
+---ro information-source-entry*
| [information-source information-source-instance]
+---ro information-source te-info-source
+---ro information-source-instance string
+---ro information-source-state
| +---ro credibility-preference? uint16
| +---ro logical-network-element? string
| +---ro network-instance? string
| +---ro topology
| +---ro node-ref? leafref
| +---ro network-ref?
| -> /nw:networks/network/network-id
+---ro connectivity-matrices
| +---ro number-of-entries? uint16
| +---ro label-restrictions
| +---ro label-restriction* [index]
| +---ro restriction? enumeration
| +---ro index uint32
| +---ro label-start
| +---ro te-label
| +---ro (technology)?
| +---: (generic)
| +---ro generic?
| rt-types:generalized-label
| +---ro direction? te-label-direction
+---ro label-end
| +---ro te-label
| +---ro (technology)?

```

```

+---:(generic)
+---ro generic?
+---ro rt-types:generalized-label
+---ro direction? te-label-direction
+---ro label-step
+---ro (technology)?
+---:(generic)
+---ro generic? int32
+---ro range-bitmap? yang:hex-string
+---ro is-allowed? boolean
+---ro underlay {te-topology-hierarchy}?
+---ro enabled? boolean
+---ro primary-path
+---ro network-ref?
+---ro -> /nw:networks/network/network-id
+---ro path-element* [path-element-id]
+---ro path-element-id uint32
+---ro (type)?
+---:(numbered-node-hop)
+---ro numbered-node-hop
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---:(numbered-link-hop)
+---ro numbered-link-hop
+---ro link-tp-id te-tp-id
+---ro hop-type? te-hop-type
+---ro direction? te-link-direction
+---:(unnumbered-link-hop)
+---ro unnumbered-link-hop
+---ro link-tp-id te-tp-id
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---ro direction? te-link-direction
+---:(as-number)
+---ro as-number-hop
+---ro as-number inet:as-number
+---ro hop-type? te-hop-type
+---:(label)
+---ro label-hop
+---ro te-label
+---ro (technology)?

```



|       |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|--|--|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  |  |  | <pre> +--:(generic)   +--ro generic?     rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| label |  |  |  |  | <pre> +--ro direction?   te-label-direction +--ro backup-path* [index]   +--ro index          uint32   +--ro network-ref?             -&gt; /nw:networks/network/network-id   +--ro path-element* [path-element-id]     +--ro path-element-id          uint32     +--ro (type)?       +--:(numbered-node-hop)         +--ro numbered-node-hop           +--ro node-id      te-node-id           +--ro hop-type?    te-hop-type       +--:(numbered-link-hop)         +--ro numbered-link-hop           +--ro link-tp-id    te-tp-id           +--ro hop-type?    te-hop-type           +--ro direction?   te-link-direction       +--:(unnumbered-link-hop)         +--ro unnumbered-link-hop           +--ro link-tp-id    te-tp-id           +--ro node-id      te-node-id           +--ro hop-type?    te-hop-type           +--ro direction?   te-link-direction       +--:(as-number)         +--ro as-number-hop           +--ro as-number    inet:as-number           +--ro hop-type?    te-hop-type       +--:(label)         +--ro label-hop           +--ro te-label             +--ro (technology)?               +--:(generic)                 +--ro generic?                   rt-types:generalized- </pre> |
| label |  |  |  |  | <pre> +--ro direction? </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

```

| te-label-direction
+--ro protection-type? identityref
+--ro tunnel-termination-points
| +--ro source? binary
| +--ro destination? binary
+--ro tunnels
| +--ro sharing? boolean
| +--ro tunnel* [tunnel-name]
| +--ro tunnel-name string
| +--ro sharing? boolean
+--ro path-constraints
| +--ro te-bandwidth
| +--ro (technology)?
| +--:(generic)
| +--ro generic? te-bandwidth
+--ro link-protection? identityref
+--ro setup-priority? uint8
+--ro hold-priority? uint8
+--ro signaling-type? identityref
+--ro path-metric-bounds
| +--ro path-metric-bound* [metric-type]
| +--ro metric-type identityref
| +--ro upper-bound? uint64
+--ro path-affinities-values
| +--ro path-affinities-value* [usage]
| +--ro usage identityref
| +--ro value? admin-groups
+--ro path-affinity-names
| +--ro path-affinity-name* [usage]
| +--ro usage identityref
| +--ro affinity-name* [name]
| +--ro name string
+--ro path-srlgs-lists
| +--ro path-srlgs-list* [usage]
| +--ro usage identityref
| +--ro values* srlg
+--ro path-srlgs-names
| +--ro path-srlgs-name* [usage]
| +--ro usage identityref
| +--ro names* string
+--ro disjointness? te-path-disjointness

```

```

+--ro optimizations
 +--ro (algorithm)?
 +--:(metric) {path-optimization-metric}?
 +--ro optimization-metric* [metric-type]
 +--ro metric-type
 | identityref
 +--ro weight?
 | uint8
 +--ro explicit-route-exclude-objects
 +--ro route-object-exclude-object*
 [index]
 +--ro index
 | uint32
 +--ro (type)?
 +--:(numbered-node-hop)
 +--ro numbered-node-hop
 +--ro node-id te-node-id
 +--ro hop-type? te-hop-type
 +--:(numbered-link-hop)
 +--ro numbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro hop-type?
 | te-hop-type
 +--ro direction?
 te-link-direction
 +--:(unnumbered-link-hop)
 +--ro unnumbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro node-id
 | te-node-id
 +--ro hop-type?
 | te-hop-type
 +--ro direction?
 te-link-direction
 +--:(as-number)
 +--ro as-number-hop
 +--ro as-number
 | inet:as-number
 +--ro hop-type?
 te-hop-type
 +--:(label)

```

|                         |  |  |                                      |                        |                     |
|-------------------------|--|--|--------------------------------------|------------------------|---------------------|
|                         |  |  |                                      |                        | +--ro label-hop     |
|                         |  |  |                                      |                        | +--ro te-label      |
|                         |  |  |                                      |                        | +--ro (technology)? |
|                         |  |  |                                      |                        | +--:(generic)       |
|                         |  |  |                                      |                        | +--ro generic?      |
|                         |  |  |                                      |                        | rt-                 |
| types:generalized-label |  |  |                                      |                        |                     |
|                         |  |  |                                      |                        | +--ro direction?    |
|                         |  |  |                                      |                        | te-label-direction  |
|                         |  |  |                                      | +--:(srlg)             |                     |
|                         |  |  |                                      | +--ro srlg             |                     |
|                         |  |  |                                      | +--ro srlg?     uint32 |                     |
|                         |  |  | +--ro explicit-route-include-objects |                        |                     |
|                         |  |  | +--ro route-object-include-object*   |                        |                     |
|                         |  |  | [index]                              |                        |                     |
|                         |  |  | +--ro index                          |                        |                     |
|                         |  |  | uint32                               |                        |                     |
|                         |  |  | +--ro (type)?                        |                        |                     |
|                         |  |  | +--:(numbered-node-hop)              |                        |                     |
|                         |  |  | +--ro numbered-node-hop              |                        |                     |
|                         |  |  | +--ro node-id         te-node-id     |                        |                     |
|                         |  |  | +--ro hop-type?     te-hop-type      |                        |                     |
|                         |  |  | +--:(numbered-link-hop)              |                        |                     |
|                         |  |  | +--ro numbered-link-hop              |                        |                     |
|                         |  |  | +--ro link-tp-id     te-tp-id        |                        |                     |
|                         |  |  | +--ro hop-type?                      |                        |                     |
|                         |  |  | te-hop-type                          |                        |                     |
|                         |  |  | +--ro direction?                     |                        |                     |
|                         |  |  | te-link-direction                    |                        |                     |
|                         |  |  | +--:(unnumbered-link-hop)            |                        |                     |
|                         |  |  | +--ro unnumbered-link-hop            |                        |                     |
|                         |  |  | +--ro link-tp-id     te-tp-id        |                        |                     |
|                         |  |  | +--ro node-id                        |                        |                     |
|                         |  |  | te-node-id                           |                        |                     |
|                         |  |  | +--ro hop-type?                      |                        |                     |
|                         |  |  | te-hop-type                          |                        |                     |
|                         |  |  | +--ro direction?                     |                        |                     |
|                         |  |  | te-link-direction                    |                        |                     |
|                         |  |  | +--:(as-number)                      |                        |                     |
|                         |  |  | +--ro as-number-hop                  |                        |                     |
|                         |  |  | +--ro as-number                      |                        |                     |

[illegible]

|       |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  | <pre> +--ro path-route-objects   +--ro path-route-object* [index]     +--ro index                               uint32     +--ro (type)?       +--:(numbered-node-hop)         +--ro numbered-node-hop           +--ro node-id       te-node-id           +--ro hop-type?     te-hop-type       +--:(numbered-link-hop)         +--ro numbered-link-hop           +--ro link-tp-id    te-tp-id           +--ro hop-type?     te-hop-type           +--ro direction?   te-link-direction       +--:(unnumbered-link-hop)         +--ro unnumbered-link-hop           +--ro link-tp-id    te-tp-id           +--ro node-id       te-node-id           +--ro hop-type?     te-hop-type           +--ro direction?   te-link-direction       +--:(as-number)         +--ro as-number-hop           +--ro as-number     inet:as-number           +--ro hop-type?     te-hop-type       +--:(label)         +--ro label-hop           +--ro te-label             +--ro (technology)?               +--:(generic)                 +--ro generic?                   rt-types:generalized- </pre> |
| label |  |  | <pre>           +--ro direction?             te-label-direction +--ro connectivity-matrix* [id]   +--ro id                               uint32   +--ro from     +--ro tp-ref?                        leafref     +--ro label-restrictions       +--ro label-restriction* [index]         +--ro restriction?              enumeration         +--ro index                     uint32 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|       |  |  |  |                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                |
|-------|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  |  |                                                                                                                                                                                                                                                                                                                                                                     | <pre> +--ro label-start     +--ro te-label         +--ro (technology)?             +--:(generic)                 +--ro generic?                     rt-types:generalized- </pre>                                               |
| label |  |  |  |                                                                                                                                                                                                                                                                                                                                                                     | <pre>     +--ro direction?         te-label-direction +--ro label-end     +--ro te-label         +--ro (technology)?             +--:(generic)                 +--ro generic?                     rt-types:generalized- </pre> |
| label |  |  |  |                                                                                                                                                                                                                                                                                                                                                                     | <pre>     +--ro direction?         te-label-direction +--ro label-step     +--ro (technology)?         +--:(generic)             +--ro generic?   int32 +--ro range-bitmap?   yang:hex-string </pre>                           |
|       |  |  |  | <pre> +--ro to +--ro tp-ref?           leafref +--ro label-restrictions     +--ro label-restriction* [index]         +--ro restriction?   enumeration         +--ro index         uint32 +--ro label-start     +--ro te-label         +--ro (technology)?             +--:(generic)                 +--ro generic?                     rt-types:generalized- </pre> |                                                                                                                                                                                                                                |
| label |  |  |  |                                                                                                                                                                                                                                                                                                                                                                     | <pre>     +--ro direction?         te-label-direction +--ro label-end     +--ro te-label         +--ro (technology)? </pre>                                                                                                    |

```

label
+---: (generic)
+---ro generic?
rt-types:generalized-
+---ro direction?
te-label-direction
+---ro label-step
+---ro (technology)?
+---: (generic)
+---ro generic? int32
+---ro range-bitmap? yang:hex-string
+---ro is-allowed? boolean
+---ro underlay {te-topology-hierarchy}?
+---ro enabled? boolean
+---ro primary-path
+---ro network-ref?
|
-> /nw:networks/network/network-id
+---ro path-element* [path-element-id]
+---ro path-element-id uint32
+---ro (type)?
+---: (numbered-node-hop)
+---ro numbered-node-hop
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---: (numbered-link-hop)
+---ro numbered-link-hop
+---ro link-tp-id te-tp-id
+---ro hop-type? te-hop-type
+---ro direction?
te-link-direction
+---: (unnumbered-link-hop)
+---ro unnumbered-link-hop
+---ro link-tp-id te-tp-id
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---ro direction?
te-link-direction
+---: (as-number)
+---ro as-number-hop
+---ro as-number inet:as-number
+---ro hop-type? te-hop-type

```



```

+---:(label)
+---ro label-hop
+---ro te-label
+---ro (technology)?
+---:(generic)
+---ro generic?
rt-

types:generalized-label

+---ro direction?
te-label-direction
+---ro backup-path* [index]
+---ro index uint32
+---ro network-ref?
| -> /nw:networks/network/network-id
+---ro path-element* [path-element-id]
+---ro path-element-id uint32
+---ro (type)?
+---:(numbered-node-hop)
+---ro numbered-node-hop
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---:(numbered-link-hop)
+---ro numbered-link-hop
+---ro link-tp-id te-tp-id
+---ro hop-type? te-hop-type
+---ro direction?
te-link-direction
+---:(unnumbered-link-hop)
+---ro unnumbered-link-hop
+---ro link-tp-id te-tp-id
+---ro node-id te-node-id
+---ro hop-type? te-hop-type
+---ro direction?
te-link-direction
+---:(as-number)
+---ro as-number-hop
+---ro as-number inet:as-number
+---ro hop-type? te-hop-type
+---:(label)
+---ro label-hop
+---ro te-label

```

```

types:generalized-label
 +--ro (technology)?
 +--:(generic)
 +--ro generic?
 rt-
 +--ro direction?
 te-label-direction
 +--ro protection-type?
 identityref
 +--ro tunnel-termination-points
 | +--ro source?
 | binary
 | +--ro destination?
 | binary
 +--ro tunnels
 | +--ro sharing?
 | boolean
 | +--ro tunnel* [tunnel-name]
 | | +--ro tunnel-name
 | | string
 | | +--ro sharing?
 | | boolean
 +--ro path-constraints
 +--ro te-bandwidth
 | +--ro (technology)?
 | +--:(generic)
 | | +--ro generic?
 | | te-bandwidth
 +--ro link-protection?
 identityref
 +--ro setup-priority?
 uint8
 +--ro hold-priority?
 uint8
 +--ro signaling-type?
 identityref
 +--ro path-metric-bounds
 | +--ro path-metric-bound* [metric-type]
 | | +--ro metric-type
 | | identityref
 | | +--ro upper-bound?
 | | uint64
 +--ro path-affinities-values
 | +--ro path-affinities-value* [usage]
 | | +--ro usage
 | | identityref
 | | +--ro value?
 | | admin-groups
 +--ro path-affinity-names
 | +--ro path-affinity-name* [usage]
 | | +--ro usage
 | | identityref
 | | +--ro affinity-name* [name]
 | | | +--ro name
 | | | string
 +--ro path-srlgs-lists
 | +--ro path-srlgs-list* [usage]
 | | +--ro usage
 | | identityref

```

```

| +---ro values* srlg
+---ro path-srlgs-names
| +---ro path-srlgs-name* [usage]
| +---ro usage identityref
| +---ro names* string
+---ro disjointness?
| te-path-disjointness
+---ro optimizations
+---ro (algorithm)?
| +---:(metric) {path-optimization-metric}?
| | +---ro optimization-metric* [metric-type]
| | | +---ro metric-type
| | | | identityref
| | | +---ro weight?
| | | | uint8
| | | +---ro explicit-route-exclude-objects
| | | | +---ro route-object-exclude-object*
| | | | | [index]
| | | | +---ro index
| | | | | uint32
| | | | +---ro (type)?
| | | | | +---:(numbered-node-hop)
| | | | | | +---ro numbered-node-hop
| | | | | | | +---ro node-id
| | | | | | | | te-node-id
| | | | | | | +---ro hop-type?
| | | | | | | | te-hop-type
| | | | | | +---:(numbered-link-hop)
| | | | | | | +---ro numbered-link-hop
| | | | | | | | +---ro link-tp-id
| | | | | | | | | te-tp-id
| | | | | | | | +---ro hop-type?
| | | | | | | | | te-hop-type
| | | | | | | | +---ro direction?
| | | | | | | | | te-link-direction
| | | | | | +---:(unnumbered-link-hop)
| | | | | | | +---ro unnumbered-link-hop
| | | | | | | | +---ro link-tp-id
| | | | | | | | | te-tp-id
| | | | | | | +---ro node-id
| | | | | | | | te-node-id

```

|                         |  |  |  |  |  |                                      |                   |
|-------------------------|--|--|--|--|--|--------------------------------------|-------------------|
|                         |  |  |  |  |  |                                      | +--ro hop-type?   |
|                         |  |  |  |  |  |                                      | te-hop-type       |
|                         |  |  |  |  |  | +--ro direction?                     | te-link-direction |
|                         |  |  |  |  |  | +---:(as-number)                     |                   |
|                         |  |  |  |  |  | +--ro as-number-hop                  |                   |
|                         |  |  |  |  |  | +--ro as-number                      |                   |
|                         |  |  |  |  |  |                                      | inet:as-number    |
|                         |  |  |  |  |  | +--ro hop-type?                      |                   |
|                         |  |  |  |  |  |                                      | te-hop-type       |
|                         |  |  |  |  |  | +---:(label)                         |                   |
|                         |  |  |  |  |  | +--ro label-hop                      |                   |
|                         |  |  |  |  |  | +--ro te-label                       |                   |
|                         |  |  |  |  |  | +--ro (technology)?                  |                   |
|                         |  |  |  |  |  |                                      | +---:(generic)    |
|                         |  |  |  |  |  |                                      | +--ro generic?    |
|                         |  |  |  |  |  |                                      | rt-               |
| types:generalized-label |  |  |  |  |  |                                      |                   |
|                         |  |  |  |  |  |                                      | +--ro direction?  |
|                         |  |  |  |  |  |                                      | te-label-         |
| direction               |  |  |  |  |  |                                      |                   |
|                         |  |  |  |  |  | +---:(srlg)                          |                   |
|                         |  |  |  |  |  | +--ro srlg                           |                   |
|                         |  |  |  |  |  | +--ro srlg? uint32                   |                   |
|                         |  |  |  |  |  | +--ro explicit-route-include-objects |                   |
|                         |  |  |  |  |  | +--ro route-object-include-object*   |                   |
|                         |  |  |  |  |  | [index]                              |                   |
|                         |  |  |  |  |  | +--ro index                          |                   |
|                         |  |  |  |  |  |                                      | uint32            |
|                         |  |  |  |  |  | +--ro (type)?                        |                   |
|                         |  |  |  |  |  | +---:(numbered-node-hop)             |                   |
|                         |  |  |  |  |  | +--ro numbered-node-hop              |                   |
|                         |  |  |  |  |  | +--ro node-id                        |                   |
|                         |  |  |  |  |  |                                      | te-node-id        |
|                         |  |  |  |  |  | +--ro hop-type?                      |                   |
|                         |  |  |  |  |  |                                      | te-hop-type       |
|                         |  |  |  |  |  | +---:(numbered-link-hop)             |                   |
|                         |  |  |  |  |  | +--ro numbered-link-hop              |                   |
|                         |  |  |  |  |  | +--ro link-tp-id                     |                   |
|                         |  |  |  |  |  |                                      | te-tp-id          |
|                         |  |  |  |  |  | +--ro hop-type?                      |                   |



```

| +--ro metric-type identityref
| +--ro accumulative-value? uint64
+--ro path-affinities-values
| +--ro path-affinities-value* [usage]
| +--ro usage identityref
| +--ro value? admin-groups
+--ro path-affinity-names
| +--ro path-affinity-name* [usage]
| +--ro usage identityref
| +--ro affinity-name* [name]
| +--ro name string
+--ro path-srlgs-lists
| +--ro path-srlgs-list* [usage]
| +--ro usage identityref
| +--ro values* srlg
+--ro path-srlgs-names
| +--ro path-srlgs-name* [usage]
| +--ro usage identityref
| +--ro names* string
+--ro path-route-objects
| +--ro path-route-object* [index]
| +--ro index uint32
| +--ro (type)?
| +--:(numbered-node-hop)
| +--ro numbered-node-hop
| +--ro node-id te-node-id
| +--ro hop-type? te-hop-type
| +--:(numbered-link-hop)
| +--ro numbered-link-hop
| +--ro link-tp-id te-tp-id
| +--ro hop-type? te-hop-type
| +--ro direction?
| te-link-direction
| +--:(unnumbered-link-hop)
| +--ro unnumbered-link-hop
| +--ro link-tp-id te-tp-id
| +--ro node-id te-node-id
| +--ro hop-type? te-hop-type
| +--ro direction?
| te-link-direction
| +--:(as-number)

```

```

+--ro as-number-hop
+--ro as-number inet:as-number
+--ro hop-type? te-hop-type
+--:(label)
+--ro label-hop
+--ro te-label
+--ro (technology)?
+--:(generic)
+--ro generic?
rt-

types:generalized-label

+--ro direction?
te-label-direction
uint32
+--ro domain-id?
+--ro is-abstract?
empty
+--ro name?
string
+--ro signaling-address*
inet:ip-address
+--ro underlay-topology {te-topology-hierarchy}?
+--ro network-ref? -> /nw:networks/network/network-id
+--ro statistics
+--ro discontinuity-time? yang:date-and-time
+--ro node
+--ro disables? yang:counter32
+--ro enables? yang:counter32
+--ro maintenance-sets? yang:counter32
+--ro maintenance-clears? yang:counter32
+--ro modifies? yang:counter32
+--ro connectivity-matrix-entry
+--ro creates? yang:counter32
+--ro deletes? yang:counter32
+--ro disables? yang:counter32
+--ro enables? yang:counter32
+--ro modifies? yang:counter32
+--rw tunnel-termination-point* [tunnel-tp-id]
+--rw tunnel-tp-id binary
+--rw admin-status?
|
te-types:te-admin-status
+--rw name? string
+--rw switching-capability? identityref
+--rw encoding? identityref
+--rw inter-layer-lock-id* uint32

```

```

+--rw protection-type? identityref
+--rw client-layer-adaptation
| +--rw switching-capability*
| | [switching-capability encoding]
| | +--rw switching-capability identityref
| | +--rw encoding identityref
| | +--rw te-bandwidth
| | | +--rw (technology)?
| | | +--:(generic)
| | | +--rw generic? te-bandwidth
+--rw local-link-connectivities
| +--rw number-of-entries? uint16
| +--rw label-restrictions
| | +--rw label-restriction* [index]
| | | +--rw restriction? enumeration
| | | +--rw index uint32
| | | +--rw label-start
| | | | +--rw te-label
| | | | | +--rw (technology)?
| | | | | +--:(generic)
| | | | | +--rw generic?
| | | | | | rt-types:generalized-label
| | | | | +--rw direction? te-label-direction
| | | +--rw label-end
| | | | +--rw te-label
| | | | | +--rw (technology)?
| | | | | +--:(generic)
| | | | | +--rw generic?
| | | | | | rt-types:generalized-label
| | | | | +--rw direction? te-label-direction
| | | +--rw label-step
| | | | +--rw (technology)?
| | | | +--:(generic)
| | | | +--rw generic? int32
| | +--rw range-bitmap? yang:hex-string
+--rw is-allowed? boolean
+--rw underlay {te-topology-hierarchy}?
| +--rw enabled? boolean
| +--rw primary-path
| | +--rw network-ref?
| | | -> /nw:networks/network/network-id

```



|       |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  | <pre> +--rw path-element* [path-element-id] +--rw path-element-id          uint32 +--rw (type)? +--: (numbered-node-hop)     +--rw numbered-node-hop         +--rw node-id          te-node-id         +--rw hop-type?       te-hop-type +--: (numbered-link-hop)     +--rw numbered-link-hop         +--rw link-tp-id       te-tp-id         +--rw hop-type?       te-hop-type         +--rw direction?      te-link-direction +--: (unnumbered-link-hop)     +--rw unnumbered-link-hop         +--rw link-tp-id       te-tp-id         +--rw node-id          te-node-id         +--rw hop-type?       te-hop-type         +--rw direction?      te-link-direction +--: (as-number)     +--rw as-number-hop         +--rw as-number        inet:as-number         +--rw hop-type?       te-hop-type +--: (label)     +--rw label-hop         +--rw te-label             +--rw (technology)?                 +--: (generic)                     +--rw generic?                         rt-types:generalized- </pre> |
| label |  |  | <pre> +--rw direction? +--rw te-label-direction +--rw backup-path* [index] +--rw index          uint32 +--rw network-ref?     -&gt; /nw:networks/network/network-id +--rw path-element* [path-element-id] +--rw path-element-id          uint32 +--rw (type)? +--: (numbered-node-hop)     +--rw numbered-node-hop </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|       |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|--|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  |  | <pre>       +--rw node-id      te-node-id       +--rw hop-type?    te-hop-type +--:(numbered-link-hop)   +--rw numbered-link-hop     +--rw link-tp-id     te-tp-id     +--rw hop-type?      te-hop-type     +--rw direction?     te-link-direction +--:(unnumbered-link-hop)   +--rw unnumbered-link-hop     +--rw link-tp-id     te-tp-id     +--rw node-id        te-node-id     +--rw hop-type?      te-hop-type     +--rw direction?     te-link-direction +--:(as-number)   +--rw as-number-hop     +--rw as-number      inet:as-number     +--rw hop-type?      te-hop-type +--:(label)   +--rw label-hop     +--rw te-label       +--rw (technology)?         +--:(generic)           +--rw generic?             rt-types:generalized- </pre> |
| label |  |  |  | <pre>       +--rw direction?         te-label-direction +--rw protection-type?      identityref +--rw tunnel-termination-points   +--rw source?             binary   +--rw destination?        binary +--rw tunnels   +--rw sharing?            boolean   +--rw tunnel* [tunnel-name]     +--rw tunnel-name       string     +--rw sharing?          boolean +--rw path-constraints   +--rw te-bandwidth     +--rw (technology)?       +--:(generic)         +--rw generic?     te-bandwidth </pre>                                                                                                                                                                                                                                                  |

```

+--rw link-protection? identityref
+--rw setup-priority? uint8
+--rw hold-priority? uint8
+--rw signaling-type? identityref
+--rw path-metric-bounds
| +--rw path-metric-bound* [metric-type]
| | +--rw metric-type identityref
| | +--rw upper-bound? uint64
+--rw path-affinities-values
| +--rw path-affinities-value* [usage]
| | +--rw usage identityref
| | +--rw value? admin-groups
+--rw path-affinity-names
| +--rw path-affinity-name* [usage]
| | +--rw usage identityref
| | +--rw affinity-name* [name]
| | | +--rw name string
+--rw path-srlgs-lists
| +--rw path-srlgs-list* [usage]
| | +--rw usage identityref
| | +--rw values* srlg
+--rw path-srlgs-names
| +--rw path-srlgs-name* [usage]
| | +--rw usage identityref
| | +--rw names* string
+--rw disjointness? te-path-disjointness
+--rw optimizations
| +--rw (algorithm)?
| | +--:(metric) {path-optimization-metric}?
| | | +--rw optimization-metric* [metric-type]
| | | | +--rw metric-type
| | | | | identityref
| | | | +--rw weight?
| | | | | uint8
| | | +--rw explicit-route-exclude-objects
| | | | +--rw route-object-exclude-object*
| | | | | [index]
| | | | | +--rw index
| | | | | | uint32
| | | +--rw (type)?
| | | | +--:(numbered-node-hop)

```

|  |  |  |  |                                      |                             |
|--|--|--|--|--------------------------------------|-----------------------------|
|  |  |  |  |                                      | +--rw numbered-node-hop     |
|  |  |  |  |                                      | +--rw node-id te-node-id    |
|  |  |  |  |                                      | +--rw hop-type? te-hop-type |
|  |  |  |  | +---:(numbered-link-hop)             |                             |
|  |  |  |  | +--rw numbered-link-hop              |                             |
|  |  |  |  | +--rw link-tp-id te-tp-id            |                             |
|  |  |  |  | +--rw hop-type?                      |                             |
|  |  |  |  | te-hop-type                          |                             |
|  |  |  |  | +--rw direction?                     |                             |
|  |  |  |  | te-link-direction                    |                             |
|  |  |  |  | +---:(unnumbered-link-hop)           |                             |
|  |  |  |  | +--rw unnumbered-link-hop            |                             |
|  |  |  |  | +--rw link-tp-id te-tp-id            |                             |
|  |  |  |  | +--rw node-id                        |                             |
|  |  |  |  | te-node-id                           |                             |
|  |  |  |  | +--rw hop-type?                      |                             |
|  |  |  |  | te-hop-type                          |                             |
|  |  |  |  | +--rw direction?                     |                             |
|  |  |  |  | te-link-direction                    |                             |
|  |  |  |  | +---:(as-number)                     |                             |
|  |  |  |  | +--rw as-number-hop                  |                             |
|  |  |  |  | +--rw as-number                      |                             |
|  |  |  |  | inet:as-number                       |                             |
|  |  |  |  | +--rw hop-type?                      |                             |
|  |  |  |  | te-hop-type                          |                             |
|  |  |  |  | +---:(label)                         |                             |
|  |  |  |  | +--rw label-hop                      |                             |
|  |  |  |  | +--rw te-label                       |                             |
|  |  |  |  | +--rw (technology)?                  |                             |
|  |  |  |  | +---:(generic)                       |                             |
|  |  |  |  | +--rw generic?                       |                             |
|  |  |  |  | rt-                                  |                             |
|  |  |  |  | +--rw direction?                     |                             |
|  |  |  |  | te-label-direction                   |                             |
|  |  |  |  | +---:(srlg)                          |                             |
|  |  |  |  | +--rw srlg                           |                             |
|  |  |  |  | +--rw srlg? uint32                   |                             |
|  |  |  |  | +--rw explicit-route-include-objects |                             |
|  |  |  |  | +--rw route-object-include-object*   |                             |
|  |  |  |  | [index]                              |                             |

|                         |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|--|--|--|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         |  |  |  |  | <pre> +--rw index     uint32 +--rw (type)?   +--:(numbered-node-hop)       +--rw numbered-node-hop           +--rw node-id      te-node-id           +--rw hop-type?    te-hop-type   +--:(numbered-link-hop)       +--rw numbered-link-hop           +--rw link-tp-id    te-tp-id           +--rw hop-type?                               +--rw direction?                   te-link-direction   +--:(unnumbered-link-hop)       +--rw unnumbered-link-hop           +--rw link-tp-id    te-tp-id           +--rw node-id                               +--rw hop-type?                                       +--rw hop-type                   +--rw direction?                       te-link-direction   +--:(as-number)       +--rw as-number-hop           +--rw as-number                               +--rw inet:as-number           +--rw hop-type?               te-hop-type   +--:(label)       +--rw label-hop           +--rw te-label               +--rw (technology)?                   +--:(generic)                       +--rw generic?                           rt-           +--rw direction?           te-label-direction   +--rw tiebreakers   +--rw tiebreaker* [tiebreaker-type] </pre> |
| types:generalized-label |  |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

```

| +--rw tiebreaker-type identityref
+--:(objective-function)
| {path-optimization-objective-function}?
| +--rw objective-function
| +--rw objective-function-type? identityref
+--ro path-properties
+--ro path-metric* [metric-type]
| +--ro metric-type identityref
| +--ro accumulative-value? uint64
+--ro path-affinities-values
| +--ro path-affinities-value* [usage]
| | +--ro usage identityref
| | +--ro value? admin-groups
+--ro path-affinity-names
| +--ro path-affinity-name* [usage]
| | +--ro usage identityref
| | +--ro affinity-name* [name]
| | | +--ro name string
+--ro path-srlgs-lists
| +--ro path-srlgs-list* [usage]
| | +--ro usage identityref
| | +--ro values* srlg
+--ro path-srlgs-names
| +--ro path-srlgs-name* [usage]
| | +--ro usage identityref
| | +--ro names* string
+--ro path-route-objects
+--ro path-route-object* [index]
| +--ro index uint32
| +--ro (type)?
| | +--:(numbered-node-hop)
| | | +--ro numbered-node-hop
| | | | +--ro node-id te-node-id
| | | | +--ro hop-type? te-hop-type
| | +--:(numbered-link-hop)
| | | +--ro numbered-link-hop
| | | | +--ro link-tp-id te-tp-id
| | | | +--ro hop-type? te-hop-type
| | | | +--ro direction? te-link-direction
| | +--:(unnumbered-link-hop)
| | | +--ro unnumbered-link-hop

```



```

|
|
| +---:(generic)
| | +---rw generic? int32
| +---rw range-bitmap? yang:hex-string
+---rw is-allowed? boolean
+---rw underlay {te-topology-hierarchy}?
| +---rw enabled? boolean
+---rw primary-path
| +---rw network-ref?
| | -> /nw:networks/network/network-id
+---rw path-element* [path-element-id]
| +---rw path-element-id uint32
+---rw (type)?
| +---:(numbered-node-hop)
| | +---rw numbered-node-hop
| | | +---rw node-id te-node-id
| | | +---rw hop-type? te-hop-type
+---:(numbered-link-hop)
| | +---rw numbered-link-hop
| | | +---rw link-tp-id te-tp-id
| | | +---rw hop-type? te-hop-type
| | | +---rw direction?
| | | | te-link-direction
+---:(unnumbered-link-hop)
| | +---rw unnumbered-link-hop
| | | +---rw link-tp-id te-tp-id
| | | +---rw node-id te-node-id
| | | +---rw hop-type? te-hop-type
| | | +---rw direction?
| | | | te-link-direction
+---:(as-number)
| | +---rw as-number-hop
| | | +---rw as-number inet:as-number
| | | +---rw hop-type? te-hop-type
+---:(label)
| +---rw label-hop
| | +---rw te-label
| | | +---rw (technology)?
| | | | +---:(generic)
| | | | | +---rw generic?
| | | | | | rt-
types:generalized-label

```



```

| +--rw direction?
| te-label-direction
+--rw backup-path* [index]
| +--rw index uint32
| +--rw network-ref?
| | -> /nw:networks/network/network-id
+--rw path-element* [path-element-id]
| +--rw path-element-id uint32
| +--rw (type)?
| | +--:(numbered-node-hop)
| | | +--rw numbered-node-hop
| | | | +--rw node-id te-node-id
| | | | +--rw hop-type? te-hop-type
| | +--:(numbered-link-hop)
| | | +--rw numbered-link-hop
| | | | +--rw link-tp-id te-tp-id
| | | | +--rw hop-type? te-hop-type
| | | | +--rw direction?
| | | | te-link-direction
| | +--:(unnumbered-link-hop)
| | | +--rw unnumbered-link-hop
| | | | +--rw link-tp-id te-tp-id
| | | | +--rw node-id te-node-id
| | | | +--rw hop-type? te-hop-type
| | | | +--rw direction?
| | | | te-link-direction
| | +--:(as-number)
| | | +--rw as-number-hop
| | | | +--rw as-number inet:as-number
| | | | +--rw hop-type? te-hop-type
| | +--:(label)
| | | +--rw label-hop
| | | | +--rw te-label
| | | | | +--rw (technology)?
| | | | | | +--:(generic)
| | | | | | | +--rw generic?
| | | | | | | rt-
| | | | | | | +--rw direction?
| | | | | | | te-label-direction
+--rw protection-type? identityref

```

types:generalized-label

```

+--rw tunnel-termination-points
| +--rw source? binary
| +--rw destination? binary
+--rw tunnels
| +--rw sharing? boolean
| +--rw tunnel* [tunnel-name]
| +--rw tunnel-name string
| +--rw sharing? boolean
+--rw path-constraints
+--rw te-bandwidth
| +--rw (technology)?
| +--:(generic)
| +--rw generic? te-bandwidth
+--rw link-protection? identityref
+--rw setup-priority? uint8
+--rw hold-priority? uint8
+--rw signaling-type? identityref
+--rw path-metric-bounds
| +--rw path-metric-bound* [metric-type]
| +--rw metric-type identityref
| +--rw upper-bound? uint64
+--rw path-affinities-values
| +--rw path-affinities-value* [usage]
| +--rw usage identityref
| +--rw value? admin-groups
+--rw path-affinity-names
| +--rw path-affinity-name* [usage]
| +--rw usage identityref
| +--rw affinity-name* [name]
| +--rw name string
+--rw path-srlgs-lists
| +--rw path-srlgs-list* [usage]
| +--rw usage identityref
| +--rw values* srlg
+--rw path-srlgs-names
| +--rw path-srlgs-name* [usage]
| +--rw usage identityref
| +--rw names* string
+--rw disjointness?
| te-path-disjointness
+--rw optimizations

```

```

+--rw (algorithm)?
 +--:(metric) {path-optimization-metric}?
 +--rw optimization-metric* [metric-type]
 +--rw metric-type
 | identityref
 +--rw weight?
 | uint8
 +--rw explicit-route-exclude-objects
 +--rw route-object-exclude-object*
 [index]
 +--rw index
 | uint32
 +--rw (type)?
 +--:(numbered-node-hop)
 +--rw numbered-node-hop
 +--rw node-id
 | te-node-id
 +--rw hop-type?
 | te-hop-type
 +--:(numbered-link-hop)
 +--rw numbered-link-hop
 +--rw link-tp-id
 | te-tp-id
 +--rw hop-type?
 | te-hop-type
 +--rw direction?
 | te-link-direction
 +--:(unnumbered-link-hop)
 +--rw unnumbered-link-hop
 +--rw link-tp-id
 | te-tp-id
 +--rw node-id
 | te-node-id
 +--rw hop-type?
 | te-hop-type
 +--rw direction?
 | te-link-direction
 +--:(as-number)
 +--rw as-number-hop
 +--rw as-number
 | inet:as-number

```

|                         |  |  |                                      |                     |                  |
|-------------------------|--|--|--------------------------------------|---------------------|------------------|
|                         |  |  |                                      |                     | +--rw hop-type?  |
|                         |  |  |                                      |                     | te-hop-type      |
|                         |  |  |                                      | +--:(label)         |                  |
|                         |  |  |                                      | +--rw label-hop     |                  |
|                         |  |  |                                      | +--rw te-label      |                  |
|                         |  |  |                                      | +--rw (technology)? |                  |
|                         |  |  |                                      | +--:(generic)       |                  |
|                         |  |  |                                      | +--rw generic?      |                  |
|                         |  |  |                                      | rt-                 |                  |
| types:generalized-label |  |  |                                      |                     |                  |
|                         |  |  |                                      |                     | +--rw direction? |
|                         |  |  |                                      |                     | te-label-        |
| direction               |  |  |                                      |                     |                  |
|                         |  |  |                                      | +--:( srlg )        |                  |
|                         |  |  |                                      | +--rw srlg          |                  |
|                         |  |  |                                      | +--rw srlg? uint32  |                  |
|                         |  |  | +--rw explicit-route-include-objects |                     |                  |
|                         |  |  | +--rw route-object-include-object*   |                     |                  |
|                         |  |  | [index]                              |                     |                  |
|                         |  |  | +--rw index                          |                     |                  |
|                         |  |  | uint32                               |                     |                  |
|                         |  |  | +--rw (type)?                        |                     |                  |
|                         |  |  | +--:( numbered-node-hop )            |                     |                  |
|                         |  |  | +--rw numbered-node-hop              |                     |                  |
|                         |  |  | +--rw node-id                        |                     |                  |
|                         |  |  | te-node-id                           |                     |                  |
|                         |  |  | +--rw hop-type?                      |                     |                  |
|                         |  |  | te-hop-type                          |                     |                  |
|                         |  |  | +--:( numbered-link-hop )            |                     |                  |
|                         |  |  | +--rw numbered-link-hop              |                     |                  |
|                         |  |  | +--rw link-tp-id                     |                     |                  |
|                         |  |  | te-tp-id                             |                     |                  |
|                         |  |  | +--rw hop-type?                      |                     |                  |
|                         |  |  | te-hop-type                          |                     |                  |
|                         |  |  | +--rw direction?                     |                     |                  |
|                         |  |  | te-link-direction                    |                     |                  |
|                         |  |  | +--:( unnumbered-link-hop )          |                     |                  |
|                         |  |  | +--rw unnumbered-link-hop            |                     |                  |
|                         |  |  | +--rw link-tp-id                     |                     |                  |
|                         |  |  | te-tp-id                             |                     |                  |
|                         |  |  | +--rw node-id                        |                     |                  |

[illegible]

```

 +--ro usage identityref
 +--ro affinity-name* [name]
 +--ro name string
+--ro path-srlgs-lists
 +--ro path-srlgs-list* [usage]
 +--ro usage identityref
 +--ro values* srlg
+--ro path-srlgs-names
 +--ro path-srlgs-name* [usage]
 +--ro usage identityref
 +--ro names* string
+--ro path-route-objects
 +--ro path-route-object* [index]
 +--ro index uint32
 +--ro (type)?
 +--:(numbered-node-hop)
 +--ro numbered-node-hop
 +--ro node-id te-node-id
 +--ro hop-type? te-hop-type
 +--:(numbered-link-hop)
 +--ro numbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro hop-type? te-hop-type
 +--ro direction?
 te-link-direction
 +--:(unnumbered-link-hop)
 +--ro unnumbered-link-hop
 +--ro link-tp-id te-tp-id
 +--ro node-id te-node-id
 +--ro hop-type? te-hop-type
 +--ro direction?
 te-link-direction
 +--:(as-number)
 +--ro as-number-hop
 +--ro as-number inet:as-number
 +--ro hop-type? te-hop-type
 +--:(label)
 +--ro label-hop
 +--ro te-label
 +--ro (technology)?
 +--:(generic)

```

```

| | +--ro generic?
| | rt-
types:generalized-label
| | +--ro direction?
| | te-label-direction
| +--ro oper-status?
| | te-types:te-oper-status
| +--ro geolocation
| | +--ro altitude? int64
| | +--ro latitude? geographic-coordinate-degree
| | +--ro longitude? geographic-coordinate-degree
| +--ro statistics
| | +--ro discontinuity-time? yang:date-and-time
| | +--ro tunnel-termination-point
| | | +--ro disables? yang:counter32
| | | +--ro enables? yang:counter32
| | | +--ro maintenance-clears? yang:counter32
| | | +--ro maintenance-sets? yang:counter32
| | | +--ro modifies? yang:counter32
| | | +--ro downs? yang:counter32
| | | +--ro ups? yang:counter32
| | | +--ro in-service-clears? yang:counter32
| | | +--ro in-service-sets? yang:counter32
| | +--ro local-link-connectivity
| | | +--ro creates? yang:counter32
| | | +--ro deletes? yang:counter32
| | | +--ro disables? yang:counter32
| | | +--ro enables? yang:counter32
| | | +--ro modifies? yang:counter32
| +--rw supporting-tunnel-termination-point*
| | [node-ref tunnel-tp-ref]
| | +--rw node-ref inet:uri
| | +--rw tunnel-tp-ref binary
augment /nw:networks/nw:network/nt:link:
+--rw te!
| +--rw (bundle-stack-level)?
| | +--:(bundle)
| | | +--rw bundled-links
| | | | +--rw bundled-link* [sequence]
| | | | | +--rw sequence uint32
| | | | | +--rw src-tp-ref? leafref

```

```

| +---rw des-tp-ref? leafref
+---:(component)
| +---rw component-links
| | +---rw component-link* [sequence]
| | | +---rw sequence uint32
| | | +---rw src-interface-ref? string
| | | +---rw des-interface-ref? string
+---rw te-link-template*
| -> ../../../../te/templates/link-template/name
| {template}?
+---rw te-link-attributes
| +---rw access-type?
| | te-types:te-link-access-type
+---rw external-domain
| +---rw network-ref?
| | -> /nw:networks/network/network-id
+---rw remote-te-node-id? te-types:te-node-id
| +---rw remote-te-link-tp-id? te-types:te-tp-id
+---rw is-abstract? empty
+---rw name? string
+---rw underlay {te-topology-hierarchy}?
| +---rw enabled? boolean
+---rw primary-path
| +---rw network-ref?
| | -> /nw:networks/network/network-id
+---rw path-element* [path-element-id]
| +---rw path-element-id uint32
| +---rw (type)?
| | +---:(numbered-node-hop)
| | | +---rw numbered-node-hop
| | | | +---rw node-id te-node-id
| | | | +---rw hop-type? te-hop-type
| | | +---:(numbered-link-hop)
| | | | +---rw numbered-link-hop
| | | | | +---rw link-tp-id te-tp-id
| | | | | +---rw hop-type? te-hop-type
| | | | | +---rw direction? te-link-direction
| | | +---:(unnumbered-link-hop)
| | | | +---rw unnumbered-link-hop
| | | | | +---rw link-tp-id te-tp-id
| | | | | +---rw node-id te-node-id

```



|       |  |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------|--|--|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  |  | <pre>       +--rw hop-type?      te-hop-type       +--rw direction?    te-link-direction +--:(as-number)       +--rw as-number-hop       +--rw as-number      inet:as-number       +--rw hop-type?      te-hop-type +--:(label)       +--rw label-hop       +--rw te-label           +--rw (technology)?               +--:(generic)                   +--rw generic?                       rt-types:generalized- </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| label |  |  |  | <pre>           +--rw direction?               te-label-direction +--rw backup-path* [index] +--rw index          uint32 +--rw network-ref?     -&gt; /nw:networks/network/network-id +--rw path-element* [path-element-id] +--rw path-element-id          uint32 +--rw (type)? +--:(numbered-node-hop)     +--rw numbered-node-hop         +--rw node-id      te-node-id         +--rw hop-type?    te-hop-type +--:(numbered-link-hop)     +--rw numbered-link-hop         +--rw link-tp-id    te-tp-id         +--rw hop-type?    te-hop-type         +--rw direction?   te-link-direction +--:(unnumbered-link-hop)     +--rw unnumbered-link-hop         +--rw link-tp-id    te-tp-id         +--rw node-id      te-node-id         +--rw hop-type?    te-hop-type         +--rw direction?   te-link-direction +--:(as-number)     +--rw as-number-hop         +--rw as-number      inet:as-number </pre> |

```

| +--rw hop-type? te-hop-type
+---:(label)
| +--rw label-hop
| +--rw te-label
| +--rw (technology)?
| | +---:(generic)
| | +--rw generic?
| | rt-types:generalized-
label
| +--rw direction?
| | te-label-direction
+---rw protection-type? identityref
+---rw tunnel-termination-points
| +--rw source? binary
| +--rw destination? binary
+---rw tunnels
| +--rw sharing? boolean
| +--rw tunnel* [tunnel-name]
| | +--rw tunnel-name string
| | +--rw sharing? boolean
+---rw admin-status?
| te-types:te-admin-status
+---rw link-index? uint64
+---rw administrative-group?
| te-types:admin-groups
+---rw interface-switching-capability*
| [switching-capability encoding]
| +--rw switching-capability identityref
| +--rw encoding identityref
| +--rw max-lsp-bandwidth* [priority]
| | +--rw priority uint8
| | +--rw te-bandwidth
| | | +--rw (technology)?
| | | +---:(generic)
| | | +--rw generic? te-bandwidth
+---rw label-restrictions
| +--rw label-restriction* [index]
| | +--rw restriction? enumeration
| | +--rw index uint32
| +--rw label-start
| | +--rw te-label

```

```

 +---rw (technology)?
 | +---:(generic)
 | +---rw generic?
 | rt-types:generalized-label
 +---rw direction? te-label-direction
+---rw label-end
| +---rw te-label
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic?
| | rt-types:generalized-label
| +---rw direction? te-label-direction
+---rw label-step
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic? int32
+---rw range-bitmap? yang:hex-string
+---rw link-protection-type? identityref
+---rw max-link-bandwidth
| +---rw te-bandwidth
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic? te-bandwidth
+---rw max-resv-link-bandwidth
| +---rw te-bandwidth
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic? te-bandwidth
+---rw unreserved-bandwidth* [priority]
| +---rw priority uint8
| +---rw te-bandwidth
| +---rw (technology)?
| | +---:(generic)
| | +---rw generic? te-bandwidth
+---rw te-default-metric? uint32
+---rw te-delay-metric? uint32
+---rw te-igp-metric? uint32
+---rw te-srlgs
| +---rw value* te-types:srlg
+---rw te-nsrlgs {nsrlg}?
| +---rw id* uint32

```

```

+--ro oper-status? te-types:te-oper-status
+--ro is-transitional? empty
+--ro information-source? te-info-source
+--ro information-source-instance? string
+--ro information-source-state
| +--ro credibility-preference? uint16
| +--ro logical-network-element? string
| +--ro network-instance? string
| +--ro topology
| +--ro link-ref? leafref
| +--ro network-ref? -> /nw:networks/network/network-id
+--ro information-source-entry*
| [information-source information-source-instance]
| +--ro information-source te-info-source
| +--ro information-source-instance string
| +--ro information-source-state
| +--ro credibility-preference? uint16
| +--ro logical-network-element? string
| +--ro network-instance? string
| +--ro topology
| +--ro link-ref? leafref
| +--ro network-ref? -> /nw:networks/network/network-id
+--ro link-index? uint64
+--ro administrative-group?
| te-types:admin-groups
+--ro interface-switching-capability*
| [switching-capability encoding]
| +--ro switching-capability identityref
| +--ro encoding identityref
| +--ro max-lsp-bandwidth* [priority]
| +--ro priority uint8
| +--ro te-bandwidth
| +--ro (technology)?
| +--:(generic)
| +--ro generic? te-bandwidth
+--ro label-restrictions
| +--ro label-restriction* [index]
| +--ro restriction? enumeration
| +--ro index uint32
| +--ro label-start

```

```

 +--ro te-label
 +--ro (technology)?
 +--:(generic)
 +--ro generic?
 rt-types:generalized-label
 +--ro direction? te-label-direction
 +--ro label-end
 +--ro te-label
 +--ro (technology)?
 +--:(generic)
 +--ro generic?
 rt-types:generalized-label
 +--ro direction? te-label-direction
 +--ro label-step
 +--ro (technology)?
 +--:(generic)
 +--ro generic? int32
 +--ro range-bitmap? yang:hex-string
+--ro link-protection-type? identityref
+--ro max-link-bandwidth
 +--ro te-bandwidth
 +--ro (technology)?
 +--:(generic)
 +--ro generic? te-bandwidth
+--ro max-resv-link-bandwidth
 +--ro te-bandwidth
 +--ro (technology)?
 +--:(generic)
 +--ro generic? te-bandwidth
+--ro unreserved-bandwidth* [priority]
 +--ro priority uint8
 +--ro te-bandwidth
 +--ro (technology)?
 +--:(generic)
 +--ro generic? te-bandwidth
+--ro te-default-metric? uint32
+--ro te-delay-metric? uint32
+--ro te-igp-metric? uint32
+--ro te-srlgs
 +--ro value* te-types:srlg
+--ro te-nsrlgs {nsrlg}?

```

```

 | +---ro id* uint32
+---ro recovery
 | +---ro restoration-status? te-types:te-recovery-status
 | +---ro protection-status? te-types:te-recovery-status
+---ro underlay {te-topology-hierarchy}?
 | +---ro dynamic? boolean
 | +---ro committed? boolean
+---ro statistics
 | +---ro discontinuity-time? yang:date-and-time
 | +---ro disables? yang:counter32
 | +---ro enables? yang:counter32
 | +---ro maintenance-clears? yang:counter32
 | +---ro maintenance-sets? yang:counter32
 | +---ro modifies? yang:counter32
 | +---ro downs? yang:counter32
 | +---ro ups? yang:counter32
 | +---ro fault-clears? yang:counter32
 | +---ro fault-detects? yang:counter32
 | +---ro protection-switches? yang:counter32
 | +---ro protection-reverts? yang:counter32
 | +---ro restoration-failures? yang:counter32
 | +---ro restoration-starts? yang:counter32
 | +---ro restoration-successes? yang:counter32
 | +---ro restoration-reversion-failures? yang:counter32
 | +---ro restoration-reversion-starts? yang:counter32
 | +---ro restoration-reversion-successes? yang:counter32
augment /nw:networks/nw:network/nw:node/nt:termination-point:
+---rw te-tp-id? te-types:te-tp-id
+---rw te!
 +---rw admin-status?
 | te-types:te-admin-status
 +---rw name? string
 +---rw interface-switching-capability*
 | [switching-capability encoding]
 | +---rw switching-capability identityref
 | +---rw encoding identityref
 | +---rw max-lsp-bandwidth* [priority]
 | | +---rw priority uint8
 | | +---rw te-bandwidth
 | | | +---rw (technology)?
 | | | | +---:(generic)

```

```
| +--rw generic? te-bandwidth
+--rw inter-domain-plug-id? binary
+--rw inter-layer-lock-id* uint32
+--ro oper-status?
| te-types:te-oper-status
+--ro geolocation
 +--ro altitude? int64
 +--ro latitude? geographic-coordinate-degree
 +--ro longitude? geographic-coordinate-degree
```

## Appendix B. Companion YANG Model for Non-NMDA Compliant Implementations

The YANG module `ietf-te-topology` defined in this document is designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [RFC8342]. In order to allow implementations to use the model even in cases when NMDA is not supported, the following companion module `ietf-te-topology-state` is defined as a state model, which mirrors the module `ietf-te-topology` defined earlier in this document. However, all data nodes in the companion module are non-configurable, to represent the applied configuration or the derived operational states.

The companion module, `ietf-te-topology-state`, is redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the module `ietf-te-topology-state` mirrors that of the module `ietf-te-topology`. The YANG tree of the module `ietf-te-topology-state` is not depicted separately.

## B.1. TE Topology State YANG Module

This module references [RFC6001], [RFC8345], and [I-D.ietf-teas-yang-te-types].

```
<CODE BEGINS> file "ietf-te-topology-state@2019-02-07.yang"
module ietf-te-topology-state {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-state";

 prefix "tet-s";

 import ietf-te-types {
 prefix "te-types";
 reference
 "I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
 Types";
 }

 import ietf-te-topology {
 prefix "tet";
 }

 import ietf-network-state {
```



```
 prefix "nw-s";
 reference "RFC 8345: A YANG Data Model for Network Topologies";
}

import ietf-network-topology-state {
 prefix "nt-s";
 reference "RFC 8345: A YANG Data Model for Network Topologies";
}

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";

contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 Editor: Xufeng Liu
 <mailto:xufeng.liu.ietf@gmail.com>

 Editor: Igor Bryskin
 <mailto:Igor.Bryskin@huawei.com>

 Editor: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editor: Tarek Saad
 <mailto:tsaad@juniper.net>

 Editor: Himanshu Shah
 <mailto:hshah@ciena.com>

 Editor: Oscar Gonzalez De Dios
 <mailto:oscar.gonzalezdedios@telefonica.com>";

description
 "TE topology state model.

 Copyright (c) 2019 IETF Trust and the persons identified as
 authors of the code. All rights reserved.
```

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices."

```
revision "2019-02-07" {
 description "Initial revision";
 reference "RFC XXXX: YANG Data Model for TE Topologies";
 // RFC Ed.: replace XXXX with actual RFC number and remove
 // this note
}

/*
 * Groupings
 */
grouping te-node-connectivity-matrix-attributes {
 description
 "Termination point references of a connectivity matrix entry.";
 container from {
 description
 "Reference to source link termination point.";
 leaf tp-ref {
 type leafref {
 path "../..//../..//../nt-s:termination-point/nt-s:tp-id";
 }
 description
 "Relative reference to a termination point.";
 }
 uses te-types:label-set-info;
 }
 container to {
 description
 "Reference to destination link termination point.";
 leaf tp-ref {
 type leafref {
 path "../..//../..//../nt-s:termination-point/nt-s:tp-id";
 }
 }
 }
}
```

```
 }
 description
 "Relative reference to a termination point.";
 }
 uses te-types:label-set-info;
}
uses tet:connectivity-matrix-entry-path-attributes;
} // te-node-connectivity-matrix-attributes

grouping te-node-tunnel-termination-point-llc-list {
 description
 "Local link connectivity list of a tunnel termination
 point on a TE node.";
 list local-link-connectivity {
 key "link-tp-ref";
 description
 "The termination capabilities between
 tunnel-termination-point and link termination-point.
 The capability information can be used to compute
 the tunnel path.
 The Interface Adjustment Capability Descriptors (IACD)
 (defined in RFC 6001) on each link-tp can be derived from
 this local-link-connectivity list.";
 reference
 "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
 for Multi-Layer and Multi-Region Networks (MLN/MRN).";

 leaf link-tp-ref {
 type leafref {
 path "../.../.../.../nt-s:termination-point/nt-s:tp-id";
 }
 description
 "Link termination point.";
 }
 uses te-types:label-set-info;
 uses tet:connectivity-matrix-entry-path-attributes;
 } // local-link-connectivity
} // te-node-tunnel-termination-point-config

/*
 * Data nodes
```

```
*/
augment "/nw-s:networks/nw-s:network/nw-s:network-types" {
 description
 "Introduce new network type for TE topology.";
 container te-topology {
 presence "Indicates TE topology.";
 description
 "Its presence identifies the TE topology type.";
 }
}

augment "/nw-s:networks" {
 description
 "Augmentation parameters for TE topologies.";
 uses tet:te-topologies-augment;
}

augment "/nw-s:networks/nw-s:network" {
 when "nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE topology.";
 uses tet:te-topology-augment;
}

augment "/nw-s:networks/nw-s:network/nw-s:node" {
 when "../nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at node level.";
 leaf te-node-id {
 type te-types:te-node-id;
 description
 "The identifier of a node in the TE topology.
 A node is specific to a topology to which it belongs.";
 }
}
```

```
 }
 container te {
 must "../te-node-id" {
 description
 "te-node-id is mandatory.";
 }
 must "count(..nw-s:supporting-node)<=1" {
 description
 "For a node in a TE topology, there cannot be more
 than 1 supporting node. If multiple nodes are abstracted,
 the underlay-topology is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";
 uses tet:te-node-augment;
 } // te
 }

 augment "/nw-s:networks/nw-s:network/nt-s:link" {
 when "../nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at link level.";
 container te {
 must "count(..nt-s:supporting-link)<=1" {
 description
 "For a link in a TE topology, there cannot be more
 than 1 supporting link. If one or more link paths are
 abstracted, the underlay is used.";
 }
 presence "TE support.";
 description
 "Indicates TE support.";
 uses tet:te-link-augment;
 } // te
 }
}
```

```
augment "/nw-s:networks/nw-s:network/nw-s:node/"
 + "nt-s:termination-point" {
 when "../..nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Configuration parameters for TE at termination point level.";
 uses tet:te-termination-point-augment;
 }

augment
 "/nw-s:networks/nw-s:network/nt-s:link/te/bundle-stack-level/"
 + "bundle/bundled-links/bundled-link" {
 when "../..nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE link bundled link.";
 leaf src-tp-ref {
 type leafref {
 path "../..nw-s:node[nw-s:node-id = "
 + "current()../..nt-s:source/"
 + "nt-s:source-node]/"
 + "nt-s:termination-point/nt-s:tp-id";
 require-instance true;
 }
 description
 "Reference to another TE termination point on the
 same source node.";
 }
 leaf des-tp-ref {
 type leafref {
 path "../..nw-s:node[nw-s:node-id = "
 + "current()../..nt-s:destination/"
 + "nt-s:dest-node]/"
 + "nt-s:termination-point/nt-s:tp-id";
 require-instance true;
 }
 }
 }
```

```
 }
 description
 "Reference to another TE termination point on the
 same destination node.";
 }
}

augment
 "/nw-s:networks/nw-s:network/nw-s:node/te/"
+ "information-source-entry/connectivity-matrices/"
+ "connectivity-matrix" {
 when "../..../nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE node connectivity-matrix.";
 uses te-node-connectivity-matrix-attributes;
}

augment
 "/nw-s:networks/nw-s:network/nw-s:node/te/te-node-attributes/"
+ "connectivity-matrices/connectivity-matrix" {
 when "../..../nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
 }
 description
 "Augment TE node connectivity-matrix.";
 uses te-node-connectivity-matrix-attributes;
}

augment
 "/nw-s:networks/nw-s:network/nw-s:node/te/"
+ "tunnel-termination-point/local-link-connectivities" {
 when "../..../nw-s:network-types/tet-s:te-topology" {
 description
 "Augmentation parameters apply only for networks with
 TE topology type.";
```

```
 }
 description
 "Augment TE node tunnel termination point LLCs
 (Local Link Connectivities).";
 uses te-node-tunnel-termination-point-llc-list;
 }
}
<CODE ENDS>
```



## Appendix C. Example: YANG Model for Technology Specific Augmentations

This section provides an example YANG module to define a technology specific TE topology model for the example-topology described in Section 6.

```
module example-topology {
 yang-version 1.1;

 namespace "http://example.com/example-topology";
 prefix "ex-topo";

 import ietf-network {
 prefix "nw";
 }

 import ietf-network-topology {
 prefix "nt";
 }

 import ietf-te-topology {
 prefix "tet";
 }

 organization
 "Example Organization";
 contact
 "Editor: Example Author";

 description
 "This module defines a topology data model for the example
 technology.";

 revision 2018-06-15 {
 description
 "Initial revision.";
 reference
 "Example reference.";
 }

 /*
 * Data nodes
```

```
*/
augment "/nw:networks/nw:network/nw:network-types/"
+ "tet:te-topology" {
 description
 "Augment network types to define example topology type.";
 container example-topology {
 presence
 "Introduce new network type for example topology.";
 description
 "Its presence identifies the example topology type.";
 }
}

augment "/nw:networks/nw:network/tet:te" {
 when "../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment network topology.";
 container attributes {
 description "Attributes for example technology.";
 leaf attribute-1 {
 type uint8;
 description "Attribute 1 for example technology.";
 }
 }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes" {
 when "../..nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment node attributes.";
 container attributes {
 description "Attributes for example technology.";
 }
}
```

```
 leaf attribute-2 {
 type uint8;
 description "Attribute 2 for example technology.";
 }
 }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment node connectivity matrices.";
 container attributes {
 description "Attributes for example technology.";
 leaf attribute-3 {
 type uint8;
 description "Attribute 3 for example technology.";
 }
 }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment node connectivity matrix.";
 container attributes {
 description "Attributes for example technology.";
 leaf attribute-3 {
 type uint8;
 description "Attribute 3 for example technology.";
 }
 }
}
```

```
 }
 }

 augment "/nw:networks/nw:network/nw:node/tet:te/"
 + "tet:tunnel-termination-point" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment tunnel termination point.";
 container attributes {
 description "Attributes for example technology.";
 leaf attribute-4 {
 type uint8;
 description "Attribute 4 for example technology.";
 }
 }
 }
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
 + "tet:te" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 description "Augment link termination point.";
 container attributes {
 description "Attributes for example technology.";
 leaf attribute-5 {
 type uint8;
 description "Attribute 5 for example technology.";
 }
 }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
 + "tet:te-link-attributes" {
```

```
when "../../../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
description "Augment link attributes.";
container attributes {
 description "Attributes for example technology.";
 leaf attribute-6 {
 type uint8;
 description "Attribute 6 for example technology.";
 }
}
}

/*
 * Augment TE bandwidth.
 */

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:max-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 case "example" {
```

```
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:max-resv-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:unreserved-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
```

```

}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../.../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
container example {
description "Attributes for example technology.";
leaf bandwidth-1 {
type uint32;
description "Bandwidth 1 for example technology.";
}
}
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../.../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
container example {
description "Attributes for example technology.";
leaf bandwidth-1 {
type uint32;
description "Bandwidth 1 for example technology.";
}
}
}
}

```

```
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../..../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
}
```



```

 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:client-layer-adaptation/"
+ "tet:switching-capability/tet:te-bandwidth/tet:technology" {
 when "../../../../../../../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
 when "../../../../../../../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;

```

```
 description "Bandwidth 1 for example technology.";
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:path-constraints/tet:te-bandwidth/tet:technology" {
when "../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
when "../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example{
```

```
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:max-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:max-resv-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
```

```
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
 }

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 when "../..../..../..../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:max-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 when "../..../..../..../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
```

```
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:max-resv-link-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
 }
 description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:unreserved-bandwidth/"
```

```
+ "tet:te-bandwidth/tet:technology" {
when "../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
}
description "Augment TE bandwidth.";
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
+ "tet:te/"
+ "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
+ "tet:te-bandwidth/tet:technology" {
when "../..../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf bandwidth-1 {
 type uint32;
 description "Bandwidth 1 for example technology.";
 }
 }
}
description "Augment TE bandwidth.";
}
```

```
/*
 * Augment TE label.
 */

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 case "example" {
```

```
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/tet:te/tet:templates/"
+ "tet:link-template/tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

/* Under te-node-attributes/connectivity-matrices */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 when "../.../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
```



```

 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
 when "../..../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../..../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
}

```

```

 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
 }

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../nw:network-types/"

```

```
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

/* Under te-node-attributes/.../connectivity-matrix */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:from/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
when ".../..."
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
```

```

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:from/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:to/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}

```

```

 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:to/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {

```

```

 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../..../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../..../nw:network-types/"

```

```
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

/* Under information-source-entry/connectivity-matrices */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 when "../../../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
}
description "Augment TE label.";
}
```

```
augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
 when "../../../../../../../../../../../nw:network-types/tet:te-topology/"
 + "ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../../../../../../../../../../../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
}
```



```

 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;

```

```

 description "Label 1 for example technology.";
 }
}
description "Augment TE label.";
}

/* Under information-source-entry/.../connectivity-matrix */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:from/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
container example {
description "Attributes for example technology.";
leaf label-1 {
type uint32;
description "Label 1 for example technology.";
}
}
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:from/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with

```

```
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:to/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 when "../..//../..//../..//../..//../..//nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/tet:to/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
```

```
+ "tet:te-label/tet:technology" {
when "../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}
```

```
augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:information-source-entry/tet:connectivity-matrices/"
+ "tet:connectivity-matrix/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 }
 }
}
```

```
 description "Label 1 for example technology.";
 }
}
description "Augment TE label.";
}

/* Under tunnel-termination-point/local-link-connectivities */

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
container example {
description "Attributes for example technology.";
leaf label-1 {
type uint32;
description "Label 1 for example technology.";
}
}
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../../../../../../../../../../../nw:network-types/tet:te-topology/"
+ "ex-topo:example-topology" {
description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
}
```

```

case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../../.../../.../../.../../.../../.../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../.../../.../../.../../.../../.../../.../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description

```

```

 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../..//../..//../..//../..//../..//../..//nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

/* Under tunnel-termination-point/.../local-link-connectivity */

augment "/nw:networks/nw:network/nw:node/tet:te/"

```



```
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
when "../.../../.../../.../../.../../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
when "../.../../.../../.../../.../../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
```

```

 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../..../..../..../..../..../..../..../..../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";

```

```

 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:tunnel-termination-point/tet:local-link-connectivities/"
+ "tet:local-link-connectivity/"
+ "tet:path-properties/tet:path-route-objects/"
+ "tet:path-route-object/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
 when "../../../../../../../../../../../../../../../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

/* Under te-link-attributes */

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 when "../../../../../../../../../../../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {

```

```
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
 when "../..../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
```

```
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:te-link-attributes/"
+ "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
+ "tet:label/tet:label-hop/tet:te-label/tet:technology" {
when "../../../../../../../../../../../../../../../nw:network-types/"
+ "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
}
case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
}
description "Augment TE label.";
}
```

```
/* Under te-link information-source-entry */

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-start/"
+ "tet:te-label/tet:technology" {
 when "../..../..../..../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
 description "Augment TE label.";
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
+ "tet:information-source-entry/"
+ "tet:label-restrictions/tet:label-restriction/tet:label-end/"
+ "tet:te-label/tet:technology" {
 when "../..../..../..../nw:network-types/"
 + "tet:te-topology/ex-topo:example-topology" {
 description
 "Augmentation parameters apply only for networks with
 example topology type.";
 }
 case "example" {
 container example {
 description "Attributes for example technology.";
 leaf label-1 {
 type uint32;
 description "Label 1 for example technology.";
 }
 }
 }
}
```

```
 }
 }
 description "Augment TE label.";
}
}
```

#### Contributors

Sergio Belotti  
Nokia  
Email: sergio.belotti@nokia.com

Dieter Beller  
Nokia  
Email: Dieter.Beller@nokia.com

Carlo Perocchio  
Ericsson  
Email: carlo.perocchio@ericsson.com

Italo Busi  
Huawei Technologies  
Email: Italo.Busi@huawei.com

#### Authors' Addresses

Xufeng Liu  
Volta Networks  
Email: xufeng.liu.ietf@gmail.com

Igor Bryskin  
Huawei Technologies  
Email: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram  
Juniper Networks  
Email: vbeeram@juniper.net

Tarek Saad  
Juniper Networks  
Email: tsaad@juniper.net

Himanshu Shah  
Ciena  
Email: hshah@ciena.com

Oscar Gonzalez De Dios  
Telefonica  
Email: oscar.gonzalezdedios@telefonica.com





Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 2, 2017

X. Liu  
Kuatro Technologies  
I. Bryskin  
Huawei Technologies  
V. Beeram  
Juniper Networks  
T. Saad  
Cisco Systems Inc  
H. Shah  
Ciena  
O. Gonzalez de Dios  
Telefonica  
October 29, 2016

YANG Data Model for Layer 3 TE Topologies  
draft-liu-teas-yang-l3-te-topo-02

Abstract

This document defines a YANG data model for layer 3 traffic engineering topologies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 2, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                |    |
|----------------------------------------------------------------|----|
| 1. Introduction . . . . .                                      | 2  |
| 1.1. Terminology . . . . .                                     | 2  |
| 2. Modeling Considerations . . . . .                           | 3  |
| 2.1. Relationship Between Layer 3 Topology and TE topology . . | 3  |
| 2.2. Relationship Modeling . . . . .                           | 3  |
| 3. Model Structure . . . . .                                   | 3  |
| 3.1. Layer 3 TE Topology Module . . . . .                      | 3  |
| 3.2. Packet Switching TE Topology Module . . . . .             | 4  |
| 4. YANG Modules . . . . .                                      | 16 |
| 4.1. Layer 3 TE Topology Module . . . . .                      | 16 |
| 4.2. Packet Switching TE Topology Module . . . . .             | 19 |
| 5. IANA Considerations . . . . .                               | 29 |
| 6. Security Considerations . . . . .                           | 30 |
| 7. References . . . . .                                        | 30 |
| 7.1. Normative References . . . . .                            | 30 |
| 7.2. Informative References . . . . .                          | 31 |
| Authors' Addresses . . . . .                                   | 31 |

## 1. Introduction

This document defines a YANG [RFC7950] data model for describing the relationship between a layer 3 network topology [I-D.ietf-i2rs-yang-l3-topology] and a TE topology [I-D.ietf-teas-yang-te-topo].

When traffic engineering is enabled on a layer 3 network topology, there will be a corresponding TE topology. The TE topology may or may not be congruent to the layer 3 network topology. When such a congruent TE topology exists, there will be a one-to-one association between the one modeling element in the layer 3 topology to another element in the TE topology. When such a congruent TE topology does not exist, the association will not be one-to-one. This YANG data model allows both cases.

### 1.1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and

"OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

## 2. Modeling Considerations

### 2.1. Relationship Between Layer 3 Topology and TE topology

In general, layer 3 network topology model and TE topology model can be used independently. When traffic engineering is enabled on a layer 3 network topology, there will be associations between objects in layer 3 network topologies and objects in TE topologies. The properties of these relations are:

- o The associations are between objects of the same class, i.e. node to node or link to link.
- o The multiplicity of such an association is: 0..1 to 0..1. An object in a layer 3 network may have zero or one associated object in the corresponding TE network.

### 2.2. Relationship Modeling

YANG data type leafref is used to model the association relationship between a layer 3 network topology and a TE topology. YANG must statements are used to enforce the referenced objects are in the topologies of proper type.

## 3. Model Structure

### 3.1. Layer 3 TE Topology Module

The model tree structure of the layer 3 TE topology module is as shown below:

```

module: ietf-l3-te-topology
augment /nw:networks/nw:network/nw:network-types/
l3t:l3-unicast-topology:
 +--rw l3-te!
augment /nw:networks/nw:network/l3t:l3-topology-attributes:
 +--rw l3-te-topology-attributes
 +--rw network-ref? leafref
augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
 +--rw l3-te-node-attributes
 +--rw node-ref? leafref
 +--rw network-ref? leafref
augment /nw:networks/nw:network/nw:node/nt:termination-point/
l3t:l3-termination-point-attributes:
 +--rw l3-te-tp-attributes
 +--rw tp-ref? leafref
 +--rw node-ref? leafref
 +--rw network-ref? leafref
augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
 +--rw l3-te-link-attributes
 +--rw link-ref? leafref
 +--rw network-ref? leafref

```

### 3.2. Packet Switching TE Topology Module

This is an augmentation to base TE topology model.

```

module: ietf-te-topology-packet
augment /nw:networks/nw:network/nw:node/tet:te/tet:config/
tet:te-node-attributes/tet:connectivity-matrix:
 +--rw performance-metric {te-performance-metric}?
 +--rw measurement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth? decimal64
 +--rw unidirectional-available-bandwidth? decimal64
 +--rw unidirectional-utilized-bandwidth? decimal64
 +--rw normality
 +--rw unidirectional-delay?
te-types:performance-metric-normality
 +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 +--rw unidirectional-max-delay?

```

```

te-types:performance-metric-normality
| +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
| +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
| +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
| +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
| +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
+--rw performance-metric-throttle {te-performance-metric}?
| +--rw unidirectional-delay-offset? uint32
| +--rw measure-interval? uint32
| +--rw advertisement-interval? uint32
| +--rw suppression-interval? uint32
| +--rw threshold-out
| | +--rw unidirectional-delay? uint32
| | +--rw unidirectional-min-delay? uint32
| | +--rw unidirectional-max-delay? uint32
| | +--rw unidirectional-delay-variation? uint32
| | +--rw unidirectional-packet-loss? decimal64
| | +--rw unidirectional-residual-bandwidth? decimal64
| | +--rw unidirectional-available-bandwidth? decimal64
| | +--rw unidirectional-utilized-bandwidth? decimal64
| +--rw threshold-in
| | +--rw unidirectional-delay? uint32
| | +--rw unidirectional-min-delay? uint32
| | +--rw unidirectional-max-delay? uint32
| | +--rw unidirectional-delay-variation? uint32
| | +--rw unidirectional-packet-loss? decimal64
| | +--rw unidirectional-residual-bandwidth? decimal64
| | +--rw unidirectional-available-bandwidth? decimal64
| | +--rw unidirectional-utilized-bandwidth? decimal64
| +--rw threshold-accelerated-advertisement
| | +--rw unidirectional-delay? uint32
| | +--rw unidirectional-min-delay? uint32
| | +--rw unidirectional-max-delay? uint32
| | +--rw unidirectional-delay-variation? uint32
| | +--rw unidirectional-packet-loss? decimal64
| | +--rw unidirectional-residual-bandwidth? decimal64
| | +--rw unidirectional-available-bandwidth? decimal64
| | +--rw unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nw:node/tet:te/tet:state/
tet:te-node-attributes/tet:connectivity-matrix:
+--ro performance-metric {te-performance-metric}?
| +--ro measurement
| | +--ro unidirectional-delay? uint32

```

```

|--ro unidirectional-min-delay? uint32
|--ro unidirectional-max-delay? uint32
|--ro unidirectional-delay-variation? uint32
|--ro unidirectional-packet-loss? decimal64
|--ro unidirectional-residual-bandwidth? decimal64
|--ro unidirectional-available-bandwidth? decimal64
|--ro unidirectional-utilized-bandwidth? decimal64
+--ro normality
+--ro unidirectional-delay?
te-types:performance-metric-normality
+--ro unidirectional-min-delay?
te-types:performance-metric-normality
+--ro unidirectional-max-delay?
te-types:performance-metric-normality
+--ro unidirectional-delay-variation?
te-types:performance-metric-normality
+--ro unidirectional-packet-loss?
te-types:performance-metric-normality
+--ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
+--ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
+--ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
+--ro performance-metric-throttle {te-performance-metric}?
+--ro unidirectional-delay-offset? uint32
+--ro measure-interval? uint32
+--ro advertisement-interval? uint32
+--ro suppression-interval? uint32
+--ro threshold-out
+--ro unidirectional-delay? uint32
+--ro unidirectional-min-delay? uint32
+--ro unidirectional-max-delay? uint32
+--ro unidirectional-delay-variation? uint32
+--ro unidirectional-packet-loss? decimal64
+--ro unidirectional-residual-bandwidth? decimal64
+--ro unidirectional-available-bandwidth? decimal64
+--ro unidirectional-utilized-bandwidth? decimal64
+--ro threshold-in
+--ro unidirectional-delay? uint32
+--ro unidirectional-min-delay? uint32
+--ro unidirectional-max-delay? uint32
+--ro unidirectional-delay-variation? uint32
+--ro unidirectional-packet-loss? decimal64
+--ro unidirectional-residual-bandwidth? decimal64
+--ro unidirectional-available-bandwidth? decimal64
+--ro unidirectional-utilized-bandwidth? decimal64
+--ro threshold-accelerated-advertisement

```

```

 +--ro unidirectional-delay? uint32
 +--ro unidirectional-min-delay? uint32
 +--ro unidirectional-max-delay? uint32
 +--ro unidirectional-delay-variation? uint32
 +--ro unidirectional-packet-loss? decimal64
 +--ro unidirectional-residual-bandwidth? decimal64
 +--ro unidirectional-available-bandwidth? decimal64
 +--ro unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nw:node/tet:te/tet:state/
tet:information-source-entry/tet:connectivity-matrix:
 +--ro performance-metric {te-performance-metric}?
 |
 | +--ro measurement
 | |
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32
 | | +--ro unidirectional-max-delay? uint32
 | | +--ro unidirectional-delay-variation? uint32
 | | +--ro unidirectional-packet-loss? decimal64
 | | +--ro unidirectional-residual-bandwidth? decimal64
 | | +--ro unidirectional-available-bandwidth? decimal64
 | | +--ro unidirectional-utilized-bandwidth? decimal64
 | +--ro normality
 | |
 | | +--ro unidirectional-delay? uint32
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-min-delay? uint32
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-max-delay? uint32
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-delay-variation? uint32
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-packet-loss? decimal64
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-residual-bandwidth? decimal64
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-available-bandwidth? decimal64
 te-types:performance-metric-normality
 | |
 | | +--ro unidirectional-utilized-bandwidth? decimal64
 te-types:performance-metric-normality
 +--ro performance-metric-throttle {te-performance-metric}?
 |
 | +--ro unidirectional-delay-offset? uint32
 | +--ro measure-interval? uint32
 | +--ro advertisement-interval? uint32
 | +--ro suppression-interval? uint32
 +--ro threshold-out
 |
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64

```



```

 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
+--ro threshold-in
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
+--ro threshold-accelerated-advertisement
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nw:node/tet:te/
tet:tunnel-termination-point/tet:config/tet:local-link-connectivity:
 +--rw performance-metric {te-performance-metric}?
 | +--rw measurement
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth? decimal64
 | | +--rw unidirectional-available-bandwidth? decimal64
 | | +--rw unidirectional-utilized-bandwidth? decimal64
 | +--rw normality
 | | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality

```

```

| +---rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
+---rw performance-metric-throttle {te-performance-metric}?
 +---rw unidirectional-delay-offset? uint32
 +---rw measure-interval? uint32
 +---rw advertisement-interval? uint32
 +---rw suppression-interval? uint32
 +---rw threshold-out
 | +---rw unidirectional-delay? uint32
 | +---rw unidirectional-min-delay? uint32
 | +---rw unidirectional-max-delay? uint32
 | +---rw unidirectional-delay-variation? uint32
 | +---rw unidirectional-packet-loss? decimal64
 | +---rw unidirectional-residual-bandwidth? decimal64
 | +---rw unidirectional-available-bandwidth? decimal64
 | +---rw unidirectional-utilized-bandwidth? decimal64
 +---rw threshold-in
 | +---rw unidirectional-delay? uint32
 | +---rw unidirectional-min-delay? uint32
 | +---rw unidirectional-max-delay? uint32
 | +---rw unidirectional-delay-variation? uint32
 | +---rw unidirectional-packet-loss? decimal64
 | +---rw unidirectional-residual-bandwidth? decimal64
 | +---rw unidirectional-available-bandwidth? decimal64
 | +---rw unidirectional-utilized-bandwidth? decimal64
 +---rw threshold-accelerated-advertisement
 | +---rw unidirectional-delay? uint32
 | +---rw unidirectional-min-delay? uint32
 | +---rw unidirectional-max-delay? uint32
 | +---rw unidirectional-delay-variation? uint32
 | +---rw unidirectional-packet-loss? decimal64
 | +---rw unidirectional-residual-bandwidth? decimal64
 | +---rw unidirectional-available-bandwidth? decimal64
 | +---rw unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nw:node/tet:te/
tet:tunnel-termination-point/tet:state/tet:local-link-connectivity:
+---ro performance-metric {te-performance-metric}?
 +---ro measurement
 | +---ro unidirectional-delay? uint32
 | +---ro unidirectional-min-delay? uint32
 | +---ro unidirectional-max-delay? uint32
 | +---ro unidirectional-delay-variation? uint32
 | +---ro unidirectional-packet-loss? decimal64
 | +---ro unidirectional-residual-bandwidth? decimal64
 | +---ro unidirectional-available-bandwidth? decimal64
 | +---ro unidirectional-utilized-bandwidth? decimal64
 +---ro normality
 | +---ro unidirectional-delay?

```

```

te-types:performance-metric-normality
| +-ro unidirectional-min-delay?
te-types:performance-metric-normality
| +-ro unidirectional-max-delay?
te-types:performance-metric-normality
| +-ro unidirectional-delay-variation?
te-types:performance-metric-normality
| +-ro unidirectional-packet-loss?
te-types:performance-metric-normality
| +-ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
| +-ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
| +-ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
+-ro performance-metric-throttle {te-performance-metric}?
 +-ro unidirectional-delay-offset? uint32
 +-ro measure-interval? uint32
 +-ro advertisement-interval? uint32
 +-ro suppression-interval? uint32
 +-ro threshold-out
 | +-ro unidirectional-delay? uint32
 | +-ro unidirectional-min-delay? uint32
 | +-ro unidirectional-max-delay? uint32
 | +-ro unidirectional-delay-variation? uint32
 | +-ro unidirectional-packet-loss? decimal64
 | +-ro unidirectional-residual-bandwidth? decimal64
 | +-ro unidirectional-available-bandwidth? decimal64
 | +-ro unidirectional-utilized-bandwidth? decimal64
 +-ro threshold-in
 | +-ro unidirectional-delay? uint32
 | +-ro unidirectional-min-delay? uint32
 | +-ro unidirectional-max-delay? uint32
 | +-ro unidirectional-delay-variation? uint32
 | +-ro unidirectional-packet-loss? decimal64
 | +-ro unidirectional-residual-bandwidth? decimal64
 | +-ro unidirectional-available-bandwidth? decimal64
 | +-ro unidirectional-utilized-bandwidth? decimal64
 +-ro threshold-accelerated-advertisement
 | +-ro unidirectional-delay? uint32
 | +-ro unidirectional-min-delay? uint32
 | +-ro unidirectional-max-delay? uint32
 | +-ro unidirectional-delay-variation? uint32
 | +-ro unidirectional-packet-loss? decimal64
 | +-ro unidirectional-residual-bandwidth? decimal64
 | +-ro unidirectional-available-bandwidth? decimal64
 | +-ro unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/tet:te/tet:templates/tet:link-template/

```

```

tet:te-link-attributes:
 +--rw performance-metric {te-performance-metric}?
 | +--rw measurement
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth? decimal64
 | | +--rw unidirectional-available-bandwidth? decimal64
 | | +--rw unidirectional-utilized-bandwidth? decimal64
 | +--rw normality
 | +--rw unidirectional-delay?
 te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
 te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
 te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
 te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
 te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
 te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
 te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?
 te-types:performance-metric-normality
 +--rw performance-metric-throttle {te-performance-metric}?
 | +--rw unidirectional-delay-offset? uint32
 | +--rw measure-interval? uint32
 | +--rw advertisement-interval? uint32
 | +--rw suppression-interval? uint32
 | +--rw threshold-out
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth? decimal64
 | | +--rw unidirectional-available-bandwidth? decimal64
 | | +--rw unidirectional-utilized-bandwidth? decimal64
 | +--rw threshold-in
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64

```

```

 | +--rw unidirectional-residual-bandwidth? decimal64
 | +--rw unidirectional-available-bandwidth? decimal64
 | +--rw unidirectional-utilized-bandwidth? decimal64
+--rw threshold-accelerated-advertisement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth? decimal64
 | +--rw unidirectional-available-bandwidth? decimal64
 | +--rw unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nt:link/tet:te/tet:config/
tet:te-link-attributes:
 +--rw performance-metric {te-performance-metric}?
 | +--rw measurement
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth? decimal64
 | | +--rw unidirectional-available-bandwidth? decimal64
 | | +--rw unidirectional-utilized-bandwidth? decimal64
 | +--rw normality
 | | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw performance-metric-throttle {te-performance-metric}?
 | +--rw unidirectional-delay-offset? uint32
 | +--rw measure-interval? uint32
 | +--rw advertisement-interval? uint32
 | +--rw suppression-interval? uint32
 | +--rw threshold-out
 | | +--rw unidirectional-delay? uint32

```

```

 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth? decimal64
 | +--rw unidirectional-available-bandwidth? decimal64
 | +--rw unidirectional-utilized-bandwidth? decimal64
 +--rw threshold-in
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth? decimal64
 | +--rw unidirectional-available-bandwidth? decimal64
 | +--rw unidirectional-utilized-bandwidth? decimal64
 +--rw threshold-accelerated-advertisement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth? decimal64
 | +--rw unidirectional-available-bandwidth? decimal64
 | +--rw unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nt:link/tet:te/tet:state/
tet:te-link-attributes:
 +--ro performance-metric {te-performance-metric}?
 | +--ro measurement
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32
 | | +--ro unidirectional-max-delay? uint32
 | | +--ro unidirectional-delay-variation? uint32
 | | +--ro unidirectional-packet-loss? decimal64
 | | +--ro unidirectional-residual-bandwidth? decimal64
 | | +--ro unidirectional-available-bandwidth? decimal64
 | | +--ro unidirectional-utilized-bandwidth? decimal64
 | +--ro normality
 | | +--ro unidirectional-delay? uint32
 te-types:performance-metric-normality
 | +--ro unidirectional-min-delay? uint32
 te-types:performance-metric-normality
 | +--ro unidirectional-max-delay? uint32
 te-types:performance-metric-normality
 | +--ro unidirectional-delay-variation? uint32
 te-types:performance-metric-normality
 | +--ro unidirectional-packet-loss? decimal64
 te-types:performance-metric-normality

```

```

 | +--ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--ro performance-metric-throttle {te-performance-metric}?
 +--ro unidirectional-delay-offset? uint32
 +--ro measure-interval? uint32
 +--ro advertisement-interval? uint32
 +--ro suppression-interval? uint32
 +--ro threshold-out
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
 +--ro threshold-in
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
 +--ro threshold-accelerated-advertisement
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth? decimal64
 | +--ro unidirectional-available-bandwidth? decimal64
 | +--ro unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/nw:network/nt:link/tet:te/tet:state/
tet:information-source-entry:
 +--ro performance-metric {te-performance-metric}?
 | +--ro measurement
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32
 | | +--ro unidirectional-max-delay? uint32
 | | +--ro unidirectional-delay-variation? uint32
 | | +--ro unidirectional-packet-loss? decimal64
 | | +--ro unidirectional-residual-bandwidth? decimal64

```

```

 | | +--ro unidirectional-available-bandwidth? decimal64
 | | +--ro unidirectional-utilized-bandwidth? decimal64
 | | +--ro normality
 | | +--ro unidirectional-delay?
te-types:performance-metric-normality
 | | +--ro unidirectional-min-delay?
te-types:performance-metric-normality
 | | +--ro unidirectional-max-delay?
te-types:performance-metric-normality
 | | +--ro unidirectional-delay-variation?
te-types:performance-metric-normality
 | | +--ro unidirectional-packet-loss?
te-types:performance-metric-normality
 | | +--ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | | +--ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | | +--ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 | | +--ro performance-metric-throttle {te-performance-metric}?
 | | +--ro unidirectional-delay-offset? uint32
 | | +--ro measure-interval? uint32
 | | +--ro advertisement-interval? uint32
 | | +--ro suppression-interval? uint32
 | | +--ro threshold-out
 | | | +--ro unidirectional-delay? uint32
 | | | +--ro unidirectional-min-delay? uint32
 | | | +--ro unidirectional-max-delay? uint32
 | | | +--ro unidirectional-delay-variation? uint32
 | | | +--ro unidirectional-packet-loss? decimal64
 | | | +--ro unidirectional-residual-bandwidth? decimal64
 | | | +--ro unidirectional-available-bandwidth? decimal64
 | | | +--ro unidirectional-utilized-bandwidth? decimal64
 | | +--ro threshold-in
 | | | +--ro unidirectional-delay? uint32
 | | | +--ro unidirectional-min-delay? uint32
 | | | +--ro unidirectional-max-delay? uint32
 | | | +--ro unidirectional-delay-variation? uint32
 | | | +--ro unidirectional-packet-loss? decimal64
 | | | +--ro unidirectional-residual-bandwidth? decimal64
 | | | +--ro unidirectional-available-bandwidth? decimal64
 | | | +--ro unidirectional-utilized-bandwidth? decimal64
 | | +--ro threshold-accelerated-advertisement
 | | | +--ro unidirectional-delay? uint32
 | | | +--ro unidirectional-min-delay? uint32
 | | | +--ro unidirectional-max-delay? uint32
 | | | +--ro unidirectional-delay-variation? uint32
 | | | +--ro unidirectional-packet-loss? decimal64

```



```

 +--ro unidirectional-residual-bandwidth? decimal64
 +--ro unidirectional-available-bandwidth? decimal64
 +--ro unidirectional-utilized-bandwidth? decimal64
augment /nw:networks/tet:te/tet:templates/tet:link-template/
tet:te-link-attributes/tet:interface-switching-capability:
 +--rw packet-switch-capable
 +--rw minimum-lsp-bandwidth? decimal64
 +--rw interface-mtu? uint16
augment /nw:networks/nw:network/nt:link/tet:te/tet:config/
tet:te-link-attributes/tet:interface-switching-capability:
 +--rw packet-switch-capable
 +--rw minimum-lsp-bandwidth? decimal64
 +--rw interface-mtu? uint16
augment /nw:networks/nw:network/nt:link/tet:te/tet:state/
tet:te-link-attributes/tet:interface-switching-capability:
 +--ro packet-switch-capable
 +--ro minimum-lsp-bandwidth? decimal64
 +--ro interface-mtu? uint16
augment /nw:networks/nw:network/nt:link/tet:te/tet:state/
tet:information-source-entry/tet:interface-switching-capability:
 +--ro packet-switch-capable
 +--ro minimum-lsp-bandwidth? decimal64
 +--ro interface-mtu? uint16

```

#### 4. YANG Modules

##### 4.1. Layer 3 TE Topology Module

```

<CODE BEGINS> file "ietf-l3-te-topology@2016-10-28.yang"
module ietf-l3-te-topology {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-l3-te-topology";
 prefix "l3tet";

 import ietf-network {
 prefix "nw";
 }
 import ietf-network-topology {
 prefix "nt";
 }
 import ietf-l3-unicast-topology {
 prefix "l3t";
 }
 import ietf-te-topology {
 prefix "tet";
 }

```

```
}

organization "TBD";
contact "TBD";
description "L3 TE Topology model";

revision 2016-10-28 {

 description "Initial revision";
 reference "TBD";
}

grouping l3-te-topology-type {
 description
 "Identifies the L3 TE topology type.";
 container l3-te {
 presence "indicates L3 TE Topology";
 description
 "Its presence identifies the L3 TE topology type.";
 }
}

augment "/nw:networks/nw:network/nw:network-types/"
+ "l3t:l3-unicast-topology" {
 description
 "Defines the L3 TE topology type.";
 uses l3-te-topology-type;
}

augment "/nw:networks/nw:network/l3t:l3-topology-attributes" {
 when "../nw:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment topology configuration";
 uses l3-te-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node/l3t:l3-node-attributes" {
 when "../nw:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment node configuration";
 uses l3-te-node-attributes;
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
+ "l3t:l3-termination-point-attributes" {
 when "../nw:network-types/l3t:l3-unicast-topology/"
```

```
 + "l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment termination point configuration";
 uses l3-te-tp-attributes;
 }

augment "/nw:networks/nw:network/nt:link/l3t:l3-link-attributes" {
 when "../nw:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment link configuration";
 uses l3-te-link-attributes;
}

grouping l3-te-topology-attributes {
 description "L3 TE topology scope attributes";
 container l3-te-topology-attributes {
 must "/nw:networks/nw:network"
 + "[nw:network-id = current()/network-ref]/nw:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE topology references";
 uses nw:network-ref;
 } // l3-te-topology-attributes
} // l3-te-topology-attributes

grouping l3-te-node-attributes {
 description "L3 TE node scope attributes";
 container l3-te-node-attributes {
 must "/nw:networks/nw:network"
 + "[nw:network-id = current()/network-ref]/nw:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE node references";
 uses nw:node-ref;
 } // l3-te
} // l3-te-node-attributes

grouping l3-te-tp-attributes {
```

```

description "L3 TE termination point scope attributes";
container l3-te-tp-attributes {
 must "/nw:networks/nw:network"
 + "[nw:network-id = current()/network-ref]/nw:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE termination point references";
 uses nt:tp-ref;
} // l3-te
} // l3-te-tp-attributes

grouping l3-te-link-attributes {
 description "L3 TE link scope attributes";
 container l3-te-link-attributes {
 must "/nw:networks/nw:network"
 + "[nw:network-id = current()/network-ref]/nw:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE link references";
 uses nt:link-ref;
 }
} // l3-te-link-attributes
}
<CODE ENDS>

```

#### 4.2. Packet Switching TE Topology Module

```

<CODE BEGINS> file "ietf-te-topology-packet@2016-10-28.yang"
module ietf-te-topology-packet {
 yang-version 1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-packet";

 prefix "tet-pkt";

 import ietf-network {
 prefix "nw";
 }
}

```

```
import ietf-network-topology {
 prefix "nt";
}

import ietf-te-topology {
 prefix "tet";
}

import ietf-te-types {
 prefix "te-types";
}

organization
 "Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";

contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 WG Chair: Lou Berger
 <mailto:lberger@labn.net>

 WG Chair: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editors: Xufeng Liu
 <mailto:xliu@kuatrotech.com>

 Igor Bryskin
 <mailto:Igor.Bryskin@huawei.com>

 Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Tarek Saad
 <mailto:tsaad@cisco.com>

 Himanshu Shah
 <mailto:hshah@ciena.com>

 Oscar Gonzalez De Dios
 <mailto:oscar.gonzalezdedios@telefonica.com>";

description "TE topology model";

revision 2016-10-28 {
 description "Initial revision";
```

```
 reference "TBD";
}

/*
 * Features
 */

feature te-performance-metric {
 description
 "This feature indicates that the system supports
 TE performance metric.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

/*
 * Groupings
 */
grouping packet-switch-capable-container {
 description
 "The container of packet switch capable attributes.";
 container packet-switch-capable {
 description
 "Interface has packet-switching capabilities.";
 leaf minimum-lsp-bandwidth {
 type decimal64 {
 fraction-digits 2;
 }
 description
 "Minimum LSP Bandwidth. Units in bytes per second";
 }
 leaf interface-mtu {
 type uint16;
 description
 "Interface MTU.";
 }
 }
}

grouping performance-metric-attributes {
 description
 "Link performance information in real time.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
```

```
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
leaf unidirectional-delay {
 type uint32 {
 range 0..16777215;
 }
 description "Delay or latency in micro seconds.";
}
leaf unidirectional-min-delay {
 type uint32 {
 range 0..16777215;
 }
 description "Minimum delay or latency in micro seconds.";
}
leaf unidirectional-max-delay {
 type uint32 {
 range 0..16777215;
 }
 description "Maximum delay or latency in micro seconds.";
}
leaf unidirectional-delay-variation {
 type uint32 {
 range 0..16777215;
 }
 description "Delay variation in micro seconds.";
}
leaf unidirectional-packet-loss {
 type decimal64 {
 fraction-digits 6;
 range "0 .. 50.331642";
 }
 description
 "Packet loss as a percentage of the total traffic sent
 over a configurable interval. The finest precision is
 0.000003%.";
}
leaf unidirectional-residual-bandwidth {
 type decimal64 {
 fraction-digits 2;
 }
 description
 "Residual bandwidth that subtracts tunnel
 reservations from Maximum Bandwidth (or link capacity)
 [RFC3630] and provides an aggregated remainder across QoS
 classes.";
}
```

```
leaf unidirectional-available-bandwidth {
 type decimal64 {
 fraction-digits 2;
 }
 description
 "Available bandwidth that is defined to be residual
 bandwidth minus the measured bandwidth used for the
 actual forwarding of non-RSVP-TE LSP packets. For a
 bundled link, available bandwidth is defined to be the
 sum of the component link available bandwidths.";
}
leaf unidirectional-utilized-bandwidth {
 type decimal64 {
 fraction-digits 2;
 }
 description
 "Bandwidth utilization that represents the actual
 utilization of the link (i.e. as measured in the router).
 For a bundled link, bandwidth utilization is defined to
 be the sum of the component link bandwidth
 utilizations.";
}
} // performance-metric-attributes

grouping performance-metric-container {
 description
 "A container containing performance metric attributes.";
 container performance-metric {
 if-feature te-performance-metric;
 description
 "Link performance information in real time.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
 container measurement {
 description
 "Measured performance metric values. Static configuration
 and manual overrides of these measurements are also
 allowed.";
 uses performance-metric-attributes;
 }
 container normality
 {
 description
 "Performance metric normality values.";
```



```
 uses performance-metric-normality-attributes;
 }
} // performance-metric-container

grouping performance-metric-normality-attributes {
 description
 "Link performance metric normality attributes.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
 leaf unidirectional-delay {
 type te-types:performance-metric-normality;
 description "Delay normality.";
 }
 leaf unidirectional-min-delay {
 type te-types:performance-metric-normality;
 description "Minimum delay or latency normality.";
 }
 leaf unidirectional-max-delay {
 type te-types:performance-metric-normality;
 description "Maximum delay or latency normality.";
 }
 leaf unidirectional-delay-variation {
 type te-types:performance-metric-normality;
 description "Delay variation normality.";
 }
 leaf unidirectional-packet-loss {
 type te-types:performance-metric-normality;
 description "Packet loss normality.";
 }
 leaf unidirectional-residual-bandwidth {
 type te-types:performance-metric-normality;
 description "Residual bandwidth normality.";
 }
 leaf unidirectional-available-bandwidth {
 type te-types:performance-metric-normality;
 description "Available bandwidth normality.";
 }
 leaf unidirectional-utilized-bandwidth {
 type te-types:performance-metric-normality;
 description "Bandwidth utilization normality.";
 }
} // performance-metric-normality-attributes
```

```
grouping performance-metric-throttle-container {
 description
 "A container controlling performance metric throttle.";
 container performance-metric-throttle {
 if-feature te-performance-metric;
 must "suppression-interval >= measure-interval" {
 error-message
 "suppression-interval cannot be less than
 measure-interval.";
 description
 "Constraint on suppression-interval and
 measure-interval.";
 }
 }
 description
 "Link performance information in real time.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
 leaf unidirectional-delay-offset {
 type uint32 {
 range 0..16777215;
 }
 description
 "Offset value to be added to the measured delay value.";
 }
 leaf measure-interval {
 type uint32;
 default 30;
 description
 "Interval in seconds to measure the extended metric
 values.";
 }
 leaf advertisement-interval {
 type uint32;
 description
 "Interval in seconds to advertise the extended metric
 values.";
 }
 leaf suppression-interval {
 type uint32 {
 range "1 .. max";
 }
 default 120;
 description
 "Interval in seconds to suppress advertising the extended
```

```

 metric values.";
 }
 container threshold-out {
 uses performance-metric-attributes;
 description
 "If the measured parameter falls outside an upper bound
 for all but the min delay metric (or lower bound for
 min-delay metric only) and the advertised value is not
 already outside that bound, anomalous announcement will be
 triggered.";
 }
 container threshold-in {
 uses performance-metric-attributes;
 description
 "If the measured parameter falls inside an upper bound
 for all but the min delay metric (or lower bound for
 min-delay metric only) and the advertised value is not
 already inside that bound, normal (anomalous-flag cleared)
 announcement will be triggered.";
 }
 container threshold-accelerated-advertisement {
 description
 "When the difference between the last advertised value and
 current measured value exceed this threshold, anomalous
 announcement will be triggered.";
 uses performance-metric-attributes;
 }
}
} // performance-metric-throttle-container

/*
 * Augmentations
 */
/* Augmentations to connectivity-matrix */
augment "/nw:networks/nw:network/nw:node/tet:te/tet:config/"
 + "tet:te-node-attributes/tet:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

augment "/nw:networks/nw:network/nw:node/tet:te/tet:state/"
 + "tet:te-node-attributes/tet:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

```

```
}

augment "/nw:networks/nw:network/nw:node/tet:te/tet:state/"
 + "tet:information-source-entry/tet:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

/* Augmentations to tunnel-termination-point */
augment "/nw:networks/nw:network/nw:node/tet:te/"
 + "tet:tunnel-termination-point/tet:config/"
 + "tet:local-link-connectivity" {
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
 + "tet:tunnel-termination-point/tet:state/"
 + "tet:local-link-connectivity" {
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

/* Augmentations to te-link-attributes */
augment "/nw:networks/tet:te/tet:templates/"
 + "tet:link-template/tet:te-link-attributes" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'switching-psc1']" {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
}

augment "/nw:networks/nw:network/nt:link/tet:te/tet:config/"
 + "tet:te-link-attributes" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'switching-psc1']" {
 description "Valid only for PSC";
 }
}
```

```

 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
 }

 augment "/nw:networks/nw:network/nt:link/tet:te/tet:state/"
 + "tet:te-link-attributes" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'switching-psc1']" {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
 }

 augment "/nw:networks/nw:network/nt:link/tet:te/tet:state/"
 + "tet:information-source-entry" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'switching-psc1']" {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses performance-metric-container;
 uses performance-metric-throttle-container;
 }

 /* Augmentations to interface-switching-capability */
 augment "/nw:networks/tet:te/tet:templates/"
 + "tet:link-template/tet:te-link-attributes/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'switching-psc1' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }

 augment "/nw:networks/nw:network/nt:link/tet:te/tet:config/"
 + "tet:te-link-attributes/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'switching-psc1' " {
 description "Valid only for PSC";
 }
 }

```

```
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }

 augment "/nw:networks/nw:network/nt:link/tet:te/tet:state/"
 + "tet:te-link-attributes/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'switching-pscl' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }

 augment "/nw:networks/nw:network/nt:link/tet:te/tet:state/"
 + "tet:information-source-entry/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'switching-pscl' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }
}
<CODE ENDS>
```

## 5. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```

URI: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```

name: ietf-l3-te-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology
prefix: l3te
reference: RFC XXXX

name: ietf-te-topology-packet
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet
prefix: tet-pkt
reference: RFC XXXX

```

## 6. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <<http://www.rfc-editor.org/info/rfc6021>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<http://www.rfc-editor.org/info/rfc7950>>.

## 7.2. Informative References

[I-D.ietf-i2rs-yang-l3-topology]

Clemm, A., Medved, J., Varga, R., Tkacik, T., Liu, X., Bryskin, I., Guo, A., Ananthakrishnan, H., Bahadur, N., and V. Beeram, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-04 (work in progress), September 2016.

[I-D.ietf-teas-yang-te-topo]

Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-06 (work in progress), October 2016.

## Authors' Addresses

Xufeng Liu  
Kuatro Technologies  
8281 Greensboro Drive, Suite 200  
McLean VA 22102  
USA

EMail: xliu@kuatrotech.com

Igor Bryskin  
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram  
Juniper Networks

EMail: vbeeram@juniper.net

Tarek Saad  
Cisco Systems Inc

EMail: tsaad@cisco.com

Himanshu Shah  
Ciena

EMail: hshah@ciena.com



Oscar Gonzalez de Dios  
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2018

X. Liu  
Jabil  
I. Bryskin  
Huawei Technologies  
V. Beeram  
Juniper Networks  
T. Saad  
Cisco Systems Inc  
H. Shah  
Ciena  
O. Gonzalez de Dios  
Telefonica  
October 30, 2017

YANG Data Model for Layer 3 TE Topologies  
draft-liu-teas-yang-l3-te-topo-05

Abstract

This document defines a YANG data model for layer 3 traffic engineering topologies.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                                            | 2  |
| 1.1. Terminology . . . . .                                                           | 3  |
| 2. Modeling Considerations . . . . .                                                 | 3  |
| 2.1. Relationship Between Layer 3 Topology and TE topology . .                       | 3  |
| 2.2. Relationship Modeling . . . . .                                                 | 3  |
| 3. Model Structure . . . . .                                                         | 3  |
| 3.1. Layer 3 TE Topology Module . . . . .                                            | 3  |
| 3.2. Packet Switching TE Topology Module . . . . .                                   | 4  |
| 4. YANG Modules . . . . .                                                            | 18 |
| 4.1. Layer 3 TE Topology Module . . . . .                                            | 18 |
| 4.2. Packet Switching TE Topology Module . . . . .                                   | 22 |
| 5. IANA Considerations . . . . .                                                     | 27 |
| 6. Security Considerations . . . . .                                                 | 28 |
| 7. References . . . . .                                                              | 28 |
| 7.1. Normative References . . . . .                                                  | 28 |
| 7.2. Informative References . . . . .                                                | 29 |
| Appendix A. Companion YANG Model for Non-NMDA Compliant<br>Implementations . . . . . | 30 |
| A.1. Layer 3 TE Topology State Module . . . . .                                      | 30 |
| A.2. Packet Switching TE Topology State Module . . . . .                             | 32 |
| Authors' Addresses . . . . .                                                         | 36 |

## 1. Introduction

This document defines a YANG [RFC7950] data model for describing the relationship between a layer 3 network topology [I-D.ietf-i2rs-yang-l3-topology] and a TE topology [I-D.ietf-teas-yang-te-top].

When traffic engineering is enabled on a layer 3 network topology, there will be a corresponding TE topology. The TE topology may or may not be congruent to the layer 3 network topology. When such a congruent TE topology exists, there will be a one-to-one association between the one modeling element in the layer 3 topology to another element in the TE topology. When such a congruent TE topology does not exist, the association will not be one-to-one. This YANG data model allows both cases.

### 1.1. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, [RFC2119].

The following terms are defined in [RFC7950] and are not redefined here:

- o augment
- o data model
- o data node

## 2. Modeling Considerations

### 2.1. Relationship Between Layer 3 Topology and TE topology

In general, layer 3 network topology model and TE topology model can be used independently. When traffic engineering is enabled on a layer 3 network topology, there will be associations between objects in layer 3 network topologies and objects in TE topologies. The properties of these relations are:

- o The associations are between objects of the same class, i.e. node to node or link to link.
- o The multiplicity of such an association is: 0..1 to 0..1. An object in a layer 3 network may have zero or one associated object in the corresponding TE network.

### 2.2. Relationship Modeling

YANG data type leafref is used to model the association relationship between a layer 3 network topology and a TE topology. YANG must statements are used to enforce the referenced objects are in the topologies of proper type.

## 3. Model Structure

### 3.1. Layer 3 TE Topology Module

The model tree structure of the layer 3 TE topology module is as shown below:

```

module: ietf-l3-te-topology
 augment /nd:networks/nd:network/nd:network-types
 /l3t:l3-unicast-topology:
 +--rw l3-te!
 augment /nd:networks/nd:network/l3t:l3-topology-attributes:
 +--rw l3-te-topology-attributes
 +--rw network-ref? -> /nw:networks/network/network-id
 augment /nd:networks/nd:network/nd:node/l3t:l3-node-attributes:
 +--rw l3-te-node-attributes
 +--rw node-ref? -> /nw:networks
 /network[nw:network-id=current()/../network-ref]/node/node-id
 +--rw network-ref? -> /nw:networks/network/network-id
 augment /nd:networks/nd:network/nd:node/lnk:termination-point
 /l3t:l3-termination-point-attributes:
 +--rw l3-te-tp-attributes
 +--rw tp-ref? -> /nw:networks
 /network[nw:network-id=current()/../network-ref]
 /node[nw:node-id=current()/../node-ref]/nt:termination-point/tp-id
 +--rw node-ref? -> /nw:networks
 /network[nw:network-id=current()/../network-ref]/node/node-id
 +--rw network-ref? -> /nw:networks/network/network-id
 augment /nd:networks/nd:network/lnk:link/l3t:l3-link-attributes:
 +--rw l3-te-link-attributes
 +--rw link-ref? -> /nw:networks
 /network[nw:network-id=current()/../network-ref]/nt:link/link-id
 +--rw network-ref? -> /nw:networks/network/network-id

```

### 3.2. Packet Switching TE Topology Module

This is an augmentation to base TE topology model.

```

module: ietf-te-topology-packet
 augment /nd:networks/nd:network/nd:node/tet:te
 /tet:te-node-attributes/tet:connectivity-matrices:
 +--rw performance-metric
 +--rw measurement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
 rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
 rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
 rt-types:bandwidth-ieee-float32

```

```

 +--rw normality
 | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw throttle
 | +--rw unidirectional-delay-offset? uint32
 | +--rw measure-interval? uint32
 | +--rw advertisement-interval? uint32
 | +--rw suppression-interval? uint32
 | +--rw threshold-out
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw threshold-in
 | | +--rw unidirectional-delay? uint32
 | | +--rw unidirectional-min-delay? uint32
 | | +--rw unidirectional-max-delay? uint32
 | | +--rw unidirectional-delay-variation? uint32
 | | +--rw unidirectional-packet-loss? decimal64
 | | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw threshold-accelerated-advertisement
 | | +--rw unidirectional-delay? uint32

```

```

 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/nd:node/tet:te
/tet:te-node-attributes/tet:connectivity-matrices
/tet:connectivity-matrix:
 +--rw performance-metric
 +--rw measurement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw normality
 +--rw unidirectional-delay? uint32
te-types:performance-metric-normality
 +--rw unidirectional-min-delay? uint32
te-types:performance-metric-normality
 +--rw unidirectional-max-delay? uint32
te-types:performance-metric-normality
 +--rw unidirectional-delay-variation? uint32
te-types:performance-metric-normality
 +--rw unidirectional-packet-loss? decimal64
te-types:performance-metric-normality
 +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw throttle
 +--rw unidirectional-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32

```

```

 +--rw threshold-out
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-in
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-accelerated-advertisement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/nd:node/tet:te
/tet:information-source-entry/tet:connectivity-matrices:
 +--ro performance-metric
 | +--ro measurement
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32
 | | +--ro unidirectional-max-delay? uint32
 | | +--ro unidirectional-delay-variation? uint32
 | | +--ro unidirectional-packet-loss? decimal64
 | | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?

```



```

rt-types:bandwidth-ieee-float32
| +---ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
| +---ro normality
| +---ro unidirectional-delay?
te-types:performance-metric-normality
| +---ro unidirectional-min-delay?
te-types:performance-metric-normality
| +---ro unidirectional-max-delay?
te-types:performance-metric-normality
| +---ro unidirectional-delay-variation?
te-types:performance-metric-normality
| +---ro unidirectional-packet-loss?
te-types:performance-metric-normality
| +---ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
| +---ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
| +---ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
+---ro throttle
+---ro unidirectional-delay-offset? uint32
+---ro measure-interval? uint32
+---ro advertisement-interval? uint32
+---ro suppression-interval? uint32
+---ro threshold-out
| +---ro unidirectional-delay? uint32
| +---ro unidirectional-min-delay? uint32
| +---ro unidirectional-max-delay? uint32
| +---ro unidirectional-delay-variation? uint32
| +---ro unidirectional-packet-loss? decimal64
| +---ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
| +---ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
| +---ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
+---ro threshold-in
| +---ro unidirectional-delay? uint32
| +---ro unidirectional-min-delay? uint32
| +---ro unidirectional-max-delay? uint32
| +---ro unidirectional-delay-variation? uint32
| +---ro unidirectional-packet-loss? decimal64
| +---ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
| +---ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
| +---ro unidirectional-utilized-bandwidth?

```

```

rt-types:bandwidth-ieee-float32
 +--ro threshold-accelerated-advertisement
 +--ro unidirectional-delay? uint32
 +--ro unidirectional-min-delay? uint32
 +--ro unidirectional-max-delay? uint32
 +--ro unidirectional-delay-variation? uint32
 +--ro unidirectional-packet-loss? decimal64
 +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/nd:node/tet:te
/tet:information-source-entry/tet:connectivity-matrices
/tet:connectivity-matrix:
 +--ro performance-metric
 +--ro measurement
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro normality
 | +--ro unidirectional-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-min-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-max-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--ro unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--ro throttle
 +--ro unidirectional-delay-offset? uint32

```

```

 +--ro measure-interval? uint32
 +--ro advertisement-interval? uint32
 +--ro suppression-interval? uint32
 +--ro threshold-out
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro threshold-in
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro threshold-accelerated-advertisement
 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/nd:node/tet:te
/tet:tunnel-termination-point/tet:local-link-connectivities:
 +--rw performance-metric
 +--rw measurement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64

```

```

 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw normality
 | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw throttle
 +--rw unidirectional-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32
 +--rw threshold-out
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-in
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32

```

```

 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-accelerated-advertisement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/nd:node/tet:te
/tet:tunnel-termination-point/tet:local-link-connectivities
/tet:local-link-connectivity:
 +--rw performance-metric
 +--rw measurement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw normality
 +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?

```

```

te-types:performance-metric-normality
 +--rw throttle
 +--rw unidirectional-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32
 +--rw threshold-out
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-in
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-accelerated-advertisement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/tet:te/tet:templates/tet:link-template
/tet:te-link-attributes:
 +--rw performance-metric
 +--rw measurement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32

```

```

 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw normality
 | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw throttle
 +--rw unidirectional-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32
 +--rw threshold-out
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-in
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32

```

```

 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-accelerated-advertisement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/lnk:link/tet:te
/tet:te-link-attributes:
 +--rw performance-metric
 +--rw measurement
 | +--rw unidirectional-delay? uint32
 | +--rw unidirectional-min-delay? uint32
 | +--rw unidirectional-max-delay? uint32
 | +--rw unidirectional-delay-variation? uint32
 | +--rw unidirectional-packet-loss? decimal64
 | +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw normality
 | +--rw unidirectional-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-min-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-max-delay?
te-types:performance-metric-normality
 | +--rw unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--rw unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--rw unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--rw unidirectional-available-bandwidth?

```



```

te-types:performance-metric-normality
| +--rw unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 +--rw throttle
 +--rw unidirectional-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32
 +--rw threshold-out
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
| +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
| +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-in
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
| +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
| +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw threshold-accelerated-advertisement
 +--rw unidirectional-delay? uint32
 +--rw unidirectional-min-delay? uint32
 +--rw unidirectional-max-delay? uint32
 +--rw unidirectional-delay-variation? uint32
 +--rw unidirectional-packet-loss? decimal64
 +--rw unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--rw unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/nd:network/lnk:link/tet:te
/tet:information-source-entry:
 +--ro performance-metric
 +--ro measurement

```

```

 | +--ro unidirectional-delay? uint32
 | +--ro unidirectional-min-delay? uint32
 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro normality
 | +--ro unidirectional-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-min-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-max-delay?
te-types:performance-metric-normality
 | +--ro unidirectional-delay-variation?
te-types:performance-metric-normality
 | +--ro unidirectional-packet-loss?
te-types:performance-metric-normality
 | +--ro unidirectional-residual-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-available-bandwidth?
te-types:performance-metric-normality
 | +--ro unidirectional-utilized-bandwidth?
te-types:performance-metric-normality
 | +--ro throttle
 | +--ro unidirectional-delay-offset? uint32
 | +--ro measure-interval? uint32
 | +--ro advertisement-interval? uint32
 | +--ro suppression-interval? uint32
 | +--ro threshold-out
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32
 | | +--ro unidirectional-max-delay? uint32
 | | +--ro unidirectional-delay-variation? uint32
 | | +--ro unidirectional-packet-loss? decimal64
 | | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro threshold-in
 | | +--ro unidirectional-delay? uint32
 | | +--ro unidirectional-min-delay? uint32

```

```

 | +--ro unidirectional-max-delay? uint32
 | +--ro unidirectional-delay-variation? uint32
 | +--ro unidirectional-packet-loss? decimal64
 | +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 | +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro threshold-accelerated-advertisement
 +--ro unidirectional-delay? uint32
 +--ro unidirectional-min-delay? uint32
 +--ro unidirectional-max-delay? uint32
 +--ro unidirectional-delay-variation? uint32
 +--ro unidirectional-packet-loss? decimal64
 +--ro unidirectional-residual-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-available-bandwidth?
rt-types:bandwidth-ieee-float32
 +--ro unidirectional-utilized-bandwidth?
rt-types:bandwidth-ieee-float32
 augment /nd:networks/tet:te/tet:templates/tet:link-template
/tet:te-link-attributes/tet:interface-switching-capability:
 +--rw packet-switch-capable
 +--rw minimum-lsp-bandwidth? rt-types:bandwidth-ieee-float32
 +--rw interface-mtu? uint16
 augment /nd:networks/nd:network/lnk:link/tet:te
/tet:te-link-attributes/tet:interface-switching-capability:
 +--rw packet-switch-capable
 +--rw minimum-lsp-bandwidth? rt-types:bandwidth-ieee-float32
 +--rw interface-mtu? uint16
 augment /nd:networks/nd:network/lnk:link/tet:te
/tet:information-source-entry/tet:interface-switching-capability:
 +--ro packet-switch-capable
 +--ro minimum-lsp-bandwidth? rt-types:bandwidth-ieee-float32
 +--ro interface-mtu? uint16

```

#### 4. YANG Modules

##### 4.1. Layer 3 TE Topology Module

```

<CODE BEGINS> file "ietf-l3-te-topology@2017-07-03.yang"
module ietf-l3-te-topology {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-l3-te-topology";
 prefix "l3tet";

```

```
import ietf-network {
 prefix "nd";
}
import ietf-network-topology {
 prefix "lnk";
}
import ietf-l3-unicast-topology {
 prefix "l3t";
}
import ietf-te-topology {
 prefix "tet";
}

organization "TBD";
contact "TBD";
description "L3 TE Topology model";

revision 2017-07-03 {

 description "Initial revision";
 reference "TBD";
}

grouping l3-te-topology-type {
 description
 "Identifies the L3 TE topology type.";
 container l3-te {
 presence "indicates L3 TE Topology";
 description
 "Its presence identifies the L3 TE topology type.";
 }
}

augment "/nd:networks/nd:network/nd:network-types/"
+ "l3t:l3-unicast-topology" {
 description
 "Defines the L3 TE topology type.";
 uses l3-te-topology-type;
}

augment "/nd:networks/nd:network/l3t:l3-topology-attributes" {
 when "../nd:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment topology configuration";
 uses l3-te-topology-attributes;
}
```

```
augment "/nd:networks/nd:network/nd:node/l3t:l3-node-attributes" {
 when "../..//nd:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment node configuration";
 uses l3-te-node-attributes;
}

augment "/nd:networks/nd:network/nd:node/l3t:l3-termination-point/"
+ "l3t:l3-termination-point-attributes" {
 when "../..//nd:network-types/l3t:l3-unicast-topology/"
+ "l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment termination point configuration";
 uses l3-te-tp-attributes;
}

augment "/nd:networks/nd:network/l3t:l3-link-attributes" {
 when "../..//nd:network-types/l3t:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment link configuration";
 uses l3-te-link-attributes;
}

grouping l3-te-topology-attributes {
 description "L3 TE topology scope attributes";
 container l3-te-topology-attributes {
 must "/nd:networks/nd:network"
 + "[nd:network-id = current()/network-ref]/nd:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE topology references";
 uses nd:network-ref;
 } // l3-te-topology-attributes
} // l3-te-topology-attributes

grouping l3-te-node-attributes {
 description "L3 TE node scope attributes";
 container l3-te-node-attributes {
 must "/nd:networks/nd:network"
 + "[nd:network-id = current()/network-ref]/nd:network-types/"
 + "tet:te-topology" {
```

```
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE node references";
 uses nd:node-ref;
 } // l3-te
 } // l3-te-node-attributes

 grouping l3-te-tp-attributes {
 description "L3 TE termination point scope attributes";
 container l3-te-tp-attributes {
 must "/nd:networks/nd:network"
 + "[nd:network-id = current()/network-ref]/nd:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE termination point references";
 uses lnk:tp-ref;
 } // l3-te
 } // l3-te-tp-attributes

 grouping l3-te-link-attributes {
 description "L3 TE link scope attributes";
 container l3-te-link-attributes {
 must "/nd:networks/nd:network"
 + "[nd:network-id = current()/network-ref]/nd:network-types/"
 + "tet:te-topology" {
 error-message
 "The referenced network must be a TE topology.";
 description
 "The referenced network must be a TE topology.";
 }
 description "Containing TE link references";
 uses lnk:link-ref;
 }
 } // l3-te-link-attributes
}
<CODE ENDS>
```

#### 4.2. Packet Switching TE Topology Module

```
<CODE BEGINS> file "ietf-te-topology-packet@2017-10-29.yang"
module ietf-te-topology-packet {
 yang-version 1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-packet";

 prefix "tet-pkt";

 import ietf-network {
 prefix "nd";
 }

 import ietf-network-topology {
 prefix "lnk";
 }

 import ietf-routing-types {
 prefix "rt-types";
 }

 import ietf-te-topology {
 prefix "tet";
 }

 import ietf-te-types {
 prefix "te-types";
 }

 organization
 "Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";

 contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 WG Chair: Lou Berger
 <mailto:lberger@labn.net>

 WG Chair: Vishnu Pavan Beeram
 <mailto:vbeeram@juniper.net>

 Editors: Xufeng Liu
 <mailto:Xufeng_Liu@jabil.com>

 Igor Bryskin
```

<mailto:Igor.Bryskin@huawei.com>

Vishnu Pavan Beeram  
<mailto:vbeeram@juniper.net>

Tarek Saad  
<mailto:tsaad@cisco.com>

Himanshu Shah  
<mailto:hshah@ciena.com>

Oscar Gonzalez De Dios  
<mailto:oscar.gonzalezdedios@telefonica.com>;

```
description "TE topology model";

revision 2017-10-29 {
 description "Initial revision";
 reference "TBD";
}

/*
 * Features
 */

feature te-performance-metric {
 description
 "This feature indicates that the system supports
 TE performance metric.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

/*
 * Groupings
 */

grouping packet-switch-capable-container {
 description
 "The container of packet switch capable attributes.";
 container packet-switch-capable {
 description
 "Interface has packet-switching capabilities.";
 leaf minimum-lsp-bandwidth {
 type rt-types:bandwidth-ieee-float32;
 }
 }
}
```



```
 description
 "Minimum LSP Bandwidth. Units in bytes per second";
 }
 leaf interface-mtu {
 type uint16;
 description
 "Interface MTU.";
 }
}

/*
 * Augmentations
 */
/* Augmentations to connectivity-matrix */
augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:te-node-attributes/tet:connectivity-matrices" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:te-node-attributes/tet:connectivity-matrices/"
 + "tet:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:information-source-entry/tet:connectivity-matrices" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:information-source-entry/tet:connectivity-matrices/"
 + "tet:connectivity-matrix" {
```

```
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
 }

/* Augmentations to tunnel-termination-point */
augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:tunnel-termination-point/"
 + "tet:local-link-connectivities" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

augment "/nd:networks/nd:network/nd:node/tet:te/"
 + "tet:tunnel-termination-point/"
 + "tet:local-link-connectivities/"
 + "tet:local-link-connectivity" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

/* Augmentations to te-link-attributes */
augment "/nd:networks/tet:te/tet:templates/"
 + "tet:link-template/tet:te-link-attributes" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'te-types:switching-psc1']" {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
}

augment "/nd:networks/nd:network/lnk:link/tet:te/"
 + "tet:te-link-attributes" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'te-types:switching-psc1']" {
 description "Valid only for PSC";
 }
}
```

```

 }
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
 }

augment "/nd:networks/nd:network/lnk:link/tet:te/"
 + "tet:information-source-entry" {
 when "tet:interface-switching-capability "
 + "[tet:switching-capability = 'te-types:switching-psc1']" {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature te-performance-metric;
 }
 }

/* Augmentations to interface-switching-capability */
augment "/nd:networks/tet:te/tet:templates/"
 + "tet:link-template/tet:te-link-attributes/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'te-types:switching-psc1' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }

augment "/nd:networks/nd:network/lnk:link/tet:te/"
 + "tet:te-link-attributes/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'te-types:switching-psc1' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
 }

augment "/nd:networks/nd:network/lnk:link/tet:te/"
 + "tet:information-source-entry/"
 + "tet:interface-switching-capability" {
 when "tet:switching-capability = 'te-types:switching-psc1' " {

```

```
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses packet-switch-capable-container;
}
}
<CODE ENDS>
```

## 5. IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```

URI: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-state
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

```

name: ietf-l3-te-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology
prefix: l3te
reference: RFC XXXX

```

```

name: ietf-l3-te-topology-state
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state
prefix: l3te-s
reference: RFC XXXX

```

```

name: ietf-te-topology-packet
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet
prefix: tet-pkt
reference: RFC XXXX

```

```

name: ietf-te-topology-packet-state
namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-state
prefix: tet-pkt-s
reference: RFC XXXX

```

## 6. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

[I-D.ietf-netmod-revised-datastores]  
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-05 (work in progress), October 2017.

## 7.2. Informative References

[I-D.ietf-i2rs-yang-l3-topology]  
Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", draft-ietf-i2rs-yang-l3-topology-12 (work in progress), October 2017.

[I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-12 (work in progress), July 2017.

## Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations

The YANG modules `ietf-l3-te-topology` and `ietf-te-topology-packet` defined in this document are designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [I-D.ietf-netmod-revised-datastores]. In order to allow implementations to use the model even in cases when NMDA is not supported, the following companion modules, `ietf-l3-te-topology-state` and `ietf-te-topology-packet-state`, are defined as state models, which mirror the modules `ietf-l3-te-topology` and `ietf-te-topology-packet` defined earlier in this document. However, all data nodes in the companion module are non-configurable, to represent the applied configuration or the derived operational states.

The companion modules, `ietf-l3-te-topology-state` and `ietf-te-topology-packet-state`, are redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the companion modules mirrors that of the cooresponding NMDA models, the YANG trees of the companion modules are not depicted separately.

## A.1. Layer 3 TE Topology State Module

```
<CODE BEGINS> file "ietf-l3-te-topology-state@2017-07-03.yang"
module ietf-l3-te-topology-state {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state";
 prefix "l3tet-s";

 import ietf-l3-te-topology {
 prefix "l3tet";
 }
 import ietf-network-state {
 prefix "nd-s";
 }
 import ietf-network-topology-state {
 prefix "lnk-s";
 }
 import ietf-l3-unicast-topology-state {
 prefix "l3t-s";
 }

 organization "TBD";
 contact "TBD";
 description "L3 TE Topology model";
```

```
revision 2017-07-03 {
 description "Initial revision";
 reference "TBD";
}

augment "/nd-s:networks/nd-s:network/nd-s:network-types/"
+ "l3t-s:l3-unicast-topology" {
 description
 "Defines the L3 TE topology type.";
 uses l3tet:l3-te-topology-type;
}

augment "/nd-s:networks/nd-s:network/"
+ "l3t-s:l3-topology-attributes" {
 when "../nd-s:network-types/l3t-s:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment topology configuration";
 uses l3tet:l3-te-topology-attributes;
}

augment "/nd-s:networks/nd-s:network/nd-s:node/"
+ "l3t-s:l3-node-attributes" {
 when "../..../nd-s:network-types/l3t-s:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment node configuration";
 uses l3tet:l3-te-node-attributes;
}

augment "/nd-s:networks/nd-s:network/nd-s:node/"
+ "lnk-s:termination-point/"
+ "l3t-s:l3-termination-point-attributes" {
 when "../..../nd-s:network-types/l3t-s:l3-unicast-topology/"
 + "l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment termination point configuration";
 uses l3tet:l3-te-tp-attributes;
}

augment "/nd-s:networks/nd-s:network/lnk-s:link/"
+ "l3t-s:l3-link-attributes" {
 when "../..../nd-s:network-types/l3t-s:l3-unicast-topology/l3-te" {
 description "Augment only for L3 TE topology";
 }
 description "Augment link configuration";
}
```



```
 uses l3tet:l3-te-link-attributes;
 }
}
<CODE ENDS>
```

## A.2. Packet Switching TE Topology State Module

```
<CODE BEGINS> file "ietf-te-topology-packet-state@2017-10-29.yang"
module ietf-te-topology-packet-state {
 yang-version 1;
 namespace
 "urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-state";

 prefix "tet-pkt-s";

 import ietf-te-topology-packet {
 prefix "tet-pkt";
 }

 import ietf-network-state {
 prefix "nd-s";
 }

 import ietf-network-topology-state {
 prefix "lnk-s";
 }

 import ietf-te-topology-state {
 prefix "tet-s";
 }

 import ietf-te-types {
 prefix "te-types";
 }

 organization
 "Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";

 contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

 WG Chair: Lou Berger
 <mailto:lberger@labn.net>
```

WG Chair: Vishnu Pavan Beeram  
<mailto:vbeeram@juniper.net>

Editors: Xufeng Liu  
<mailto:Xufeng\_Liu@jabil.com>

Igor Bryskin  
<mailto:Igor.Bryskin@huawei.com>

Vishnu Pavan Beeram  
<mailto:vbeeram@juniper.net>

Tarek Saad  
<mailto:tsaad@cisco.com>

Himanshu Shah  
<mailto:hshah@ciena.com>

Oscar Gonzalez De Dios  
<mailto:oscar.gonzalezdedios@telefonica.com>;

```
description "TE topology model";

revision 2017-10-29 {
 description "Initial revision";
 reference "TBD";
}

/*
 * Augmentations
 */
/* Augmentations to connectivity-matrix */
augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
+ "tet-s:te-node-attributes/tet-s:connectivity-matrices" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
}

augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
+ "tet-s:te-node-attributes/tet-s:connectivity-matrices/"
+ "tet-s:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
}
```

```

 }
 }

 augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
 + "tet-s:information-source-entry/"
 + "tet-s:connectivity-matrices" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
 }

 augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
 + "tet-s:information-source-entry/"
 + "tet-s:connectivity-matrices/"
 + "tet-s:connectivity-matrix" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
 }

 /* Augmentations to tunnel-termination-point */
 augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
 + "tet-s:tunnel-termination-point/"
 + "tet-s:local-link-connectivities" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
 }

 augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te/"
 + "tet-s:tunnel-termination-point/"
 + "tet-s:local-link-connectivities/"
 + "tet-s:local-link-connectivity" {
 description
 "Parameters for PSC TE topology.";
 uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
 }
 }

 /* Augmentations to te-link-attributes */
 augment "/nd-s:networks/tet-s:te/tet-s:templates/"

```

```

 + "tet-s:link-template/tet-s:te-link-attributes" {
when "tet-s:interface-switching-capability "
+ "[tet-s:switching-capability = 'te-types:switching-pscl']" {
 description "Valid only for PSC";
}
description
 "Parameters for PSC TE topology.";
uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
}
}

augment "/nd-s:networks/nd-s:network/lnk-s:link/tet-s:te/"
+ "tet-s:te-link-attributes" {
when "tet-s:interface-switching-capability "
+ "[tet-s:switching-capability = 'te-types:switching-pscl']" {
 description "Valid only for PSC";
}
description
 "Parameters for PSC TE topology.";
uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
}
}

augment "/nd-s:networks/nd-s:network/lnk-s:link/tet-s:te/"
+ "tet-s:information-source-entry" {
when "tet-s:interface-switching-capability "
+ "[tet-s:switching-capability = 'te-types:switching-pscl']" {
 description "Valid only for PSC";
}
description
 "Parameters for PSC TE topology.";
uses te-types:performance-metric-container {
 if-feature tet-pkt:te-performance-metric;
}
}

/* Augmentations to interface-switching-capability */
augment "/nd-s:networks/tet-s:te/tet-s:templates/"
+ "tet-s:link-template/tet-s:te-link-attributes/"
+ "tet-s:interface-switching-capability" {
when "tet-s:switching-capability = 'te-types:switching-pscl' " {
 description "Valid only for PSC";
}
description
 "Parameters for PSC TE topology.";
uses tet-pkt:packet-switch-capable-container;
}

```

```
 }

 augment "/nd-s:networks/nd-s:network/lnk-s:link/tet-s:te/"
 + "tet-s:te-link-attributes/"
 + "tet-s:interface-switching-capability" {
 when "tet-s:switching-capability = 'te-types:switching-psc1' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses tet-pkt:packet-switch-capable-container;
 }

 augment "/nd-s:networks/nd-s:network/lnk-s:link/tet-s:te/"
 + "tet-s:information-source-entry/"
 + "tet-s:interface-switching-capability" {
 when "tet-s:switching-capability = 'te-types:switching-psc1' " {
 description "Valid only for PSC";
 }
 description
 "Parameters for PSC TE topology.";
 uses tet-pkt:packet-switch-capable-container;
 }
 }
}
<CODE ENDS>
```

## Authors' Addresses

Xufeng Liu  
Jabil  
8281 Greensboro Drive, Suite 200  
McLean VA 22102  
USA

EMail: Xufeng\_Liu@jabil.com

Igor Bryskin  
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram  
Juniper Networks

EMail: vbeeram@juniper.net

Tarek Saad  
Cisco Systems Inc

EMail: tsaad@cisco.com

Himanshu Shah  
Ciena

EMail: hshah@ciena.com

Oscar Gonzalez de Dios  
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

MPLS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 8, 2017

M. Taillon  
T. Saad, Ed.  
R. Gandhi  
Cisco Systems, Inc.  
A. Deshmukh  
M. Jork  
V. Beeram  
Juniper Networks  
March 07, 2017

RSVP-TE Summary Fast Reroute Extensions for LSP Tunnels  
draft-mtaillon-mpls-summary-frr-rsvpte-05

Abstract

This document defines Resource Reservation Protocol (RSVP) Traffic-Engineering (TE) signaling extensions that reduce the amount of RSVP signaling required for Fast Reroute (FRR) procedures and subsequently improve the scalability of the RSVP-TE signaling when undergoing FRR convergence after a link or node failure. Such extensions allow the RSVP message exchange between the Point of Local Repair (PLR) and the Merge Point (MP) to be independent of the number of protected Label Switched Paths (LSPs) traversing between them when facility bypass FRR protection is used. The signaling extensions are fully backwards compatible with nodes that do not support them.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                      |    |
|------------------------------------------------------|----|
| 1. Introduction . . . . .                            | 2  |
| 2. Conventions Used in This Document . . . . .       | 3  |
| 2.1. Key Word Definitions . . . . .                  | 3  |
| 2.2. Terminology . . . . .                           | 3  |
| 3. Summary FRR Signaling Procedures . . . . .        | 3  |
| 3.1. Signaling Procedures Prior to Failure . . . . . | 4  |
| 3.1.1. Extended ASSOCIATION Object . . . . .         | 5  |
| 3.1.2. PLR Summary FRR Signaling Procedure . . . . . | 9  |
| 3.1.3. MP Summary FRR Signaling Procedure . . . . .  | 9  |
| 3.2. Signaling Procedures Post Failure . . . . .     | 10 |
| 3.2.1. SUMMARY_FRR_BYPASS Object . . . . .           | 10 |
| 3.2.2. PLR Summary FRR Signaling Procedure . . . . . | 12 |
| 3.2.3. MP Summary FRR Signaling Procedure . . . . .  | 12 |
| 3.3. Refreshing Summary FRR Active LSPs . . . . .    | 13 |
| 4. Compatibility . . . . .                           | 13 |
| 5. Security Considerations . . . . .                 | 13 |
| 6. IANA Considerations . . . . .                     | 13 |
| 7. Acknowledgments . . . . .                         | 14 |
| 8. Contributors . . . . .                            | 14 |
| 9. Normative References . . . . .                    | 14 |
| Authors' Addresses . . . . .                         | 15 |

## 1. Introduction

The Fast Reroute (FRR) procedures defined in [RFC4090] describe the mechanisms for the Point of Local Repair (PLR) to reroute traffic and signaling of a protected RSVP-TE LSP onto the bypass tunnel in the event of a TE link or node failure. Such signaling procedures are performed individually for each affected protected LSP. This may eventually lead to control plane scalability and latency issues under limited (memory and CPU processing) resources after a failure that



affects a large number of protected LSPs traversing the same PLR and Merge Point (MP) nodes.

For example, in a large RSVP-TE LSPs scale deployment, a single LSR acting as a PLR node may host tens of thousands of protected RSVP-TE LSPs egressing the same link, and also act as a MP node for similar number of LSPs ingressing the same link. In the event of the failure of the link or neighbor node, the RSVP-TE control plane of the node when acting as PLR becomes busy rerouting protected LSPs signaling over the bypass tunnel(s) in one direction, and when acting as an MP node becomes busy merging RSVP states from signaling received over bypass tunnels for LSP(s) in the reverse direction. Subsequently, the head-end LER(s) that are notified of the local repair at downstream LSR will attempt to (re)converge affected RSVP-TE LSPs onto newly computed paths - possibly traversing the same previously affected LSR(s). As a result, the RSVP-TE control plane at the PLR and MP becomes overwhelmed by the amount of FRR RSVP-TE processing overhead following the link or node failure, and the competing other control plane protocol(s) (e.g. the IGP) that undergo their convergence at the same time.

The extensions defined in this document enable a MP node to become aware of the PLR node's bypass tunnel assignment group and allow FRR procedures between PLR node and MP node to be signaled and processed on groups of LSPs. Further, the MESSAGE\_ID for the rerouted PATH and RESV states are exchanged a priori to the fault such that Summary Refresh procedures defined in [RFC2961] can continue to be used to refresh the rerouted state(s) after FRR has occurred.

## 2. Conventions Used in This Document

### 2.1. Key Word Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

### 2.2. Terminology

The reader is assumed to be familiar with the terminology in [RFC3209] and [RFC4090].

## 3. Summary FRR Signaling Procedures

The RSVP ASSOCIATION object is defined in [RFC4872] as a means to associate LSPs with each other. For example, in the context of GMPLS-controlled LSP(s), the object is used to associate recovery

LSPs with the LSP they are protecting. The Extended ASSOCIATION object is introduced in [RFC6780] to expand on the possible usage of the ASSOCIATION object and generalize the definition of the Association ID field.

This document proposes the use of the Extended ASSOCIATION object to carry the Summary FRR information and associate the protected LSP(s) with the bypass tunnel that protects them. To this extent, a new Association Type for the Extended ASSOCIATION object, and a new Association ID are proposed in this draft to describe the Bypass Summary FRR (B-SFRR) association.

The PLR creates and manages the Summary FRR LSP groups (Bypass\_Group\_Identifier) and shares them with the MP via signaling. Protected LSPs sharing the same egress link and bypass assignment are grouped together and are assigned the same group. The MP maintains the PLR group assignments learned via signaling, and acknowledges the group assignments via signaling. Once the PLR receives the acknowledgment, FRR signaling can proceed as group based.

The PLR node that supports Summary FRR procedures adds the Extended ASSOCIATION object with Bypass Summary FRR Association Type - referred to thereon in this document as B-SFRR Extended ASSOCIATION object- in the RSVP Path message of the protected LSP to inform the MP of the PLR's assigned bypass tunnel, Summary FRR Bypass\_Group\_Identifier, and the MESSAGE\_ID object that the PLR will use to refresh the protected LSP PATH state after FRR occurs.

The MP node that supports Summary FRR procedures adds the B-SFRR Extended ASSOCIATION object in a RSVP Resv message of the protected LSP to acknowledge the PLR's bypass tunnel assignment, and provide the MESSAGE\_ID object that the MP node will use to refresh the protected LSP RESV state after FRR occurs.

This document also defines a new RSVP FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object that is sent within the RSVP Path message of a bypass LSP to inform the MP node that one or more groups of protected LSPs that are being protected by the bypass tunnel are being rerouted i.e. signaling is rerouted over the bypass tunnel.

### 3.1. Signaling Procedures Prior to Failure

Before Summary FRR procedures can be used, a handshake MUST be completed between the PLR and MP. This handshake is performed using B-SFRR Extended ASSOCIATION object that is carried in both the RSVP Path and Resv messages of the protected LSP.

### 3.1.1.1. Extended ASSOCIATION Object

The B-SFRR Extended ASSOCIATION object is populated using the rules defined below to associate the Summary FRR enabled protected LSP with the bypass LSP that is protecting it.

The Association Type, Association ID, and Association Source MUST be set as defined in [RFC4872] for the ASSOCIATION Object. More specifically:

#### Association Source:

The Association Source is set to an address selected by the node that originates the association. For Bypass Summary FRR association it is set to an address of the PLR node.

#### Association Type:

The Association Type is set to indicate the Bypass Summary FRR association. A new Association Type is defined as follows:

| Value   | Type                                    |
|---------|-----------------------------------------|
| -----   | -----                                   |
| (TBD-1) | Bypass Summary FRR Association (B-SFRR) |

#### Extended Association ID:

The Extended Association ID is populated by the node originating the association -- i.e. the PLR for the Bypass Summary FRR association. The rules to populate the Extended Association ID in this case is described below.

### 3.1.1.1.1. IPv4 Extended Association ID

The IPv4 Extended Association ID for Summary FRR bypass assignment has the following format:

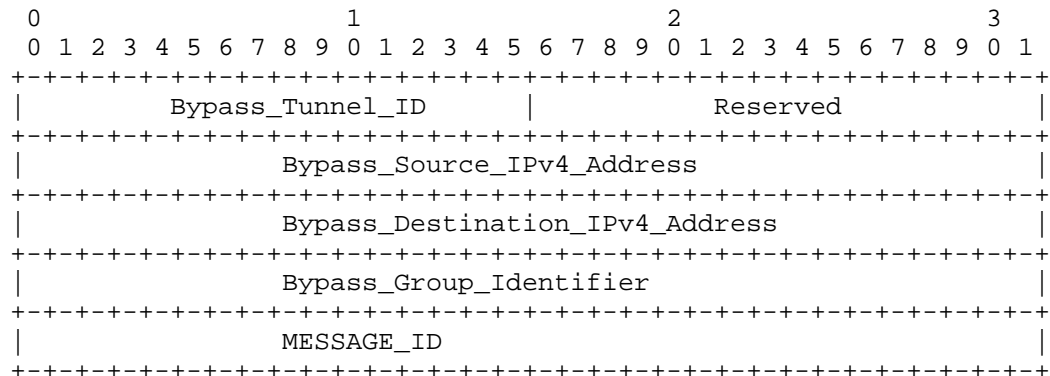


Figure 1: The IPv4 Extended Association ID field

Bypass\_Tunnel\_ID: 16 bits

The bypass tunnel identifier.

Reserved: 16 bits

Reserved for future use.

Bypass\_Source\_IPv4\_Address: 32 bits

The bypass tunnel source IPV4 address.

Bypass\_Destination\_IPv4\_Address: 32 bits

The bypass tunnel destination IPV4 address.

Bypass\_Group\_Identifier: 32 bits

The bypass tunnel group identifier.

MESSAGE\_ID

A MESSAGE\_ID object as defined by [RFC2961].

### 3.1.1.2. IPv6 Extended Association ID

The IPv6 Extended Association ID field for the Summary FRR information has the following format:

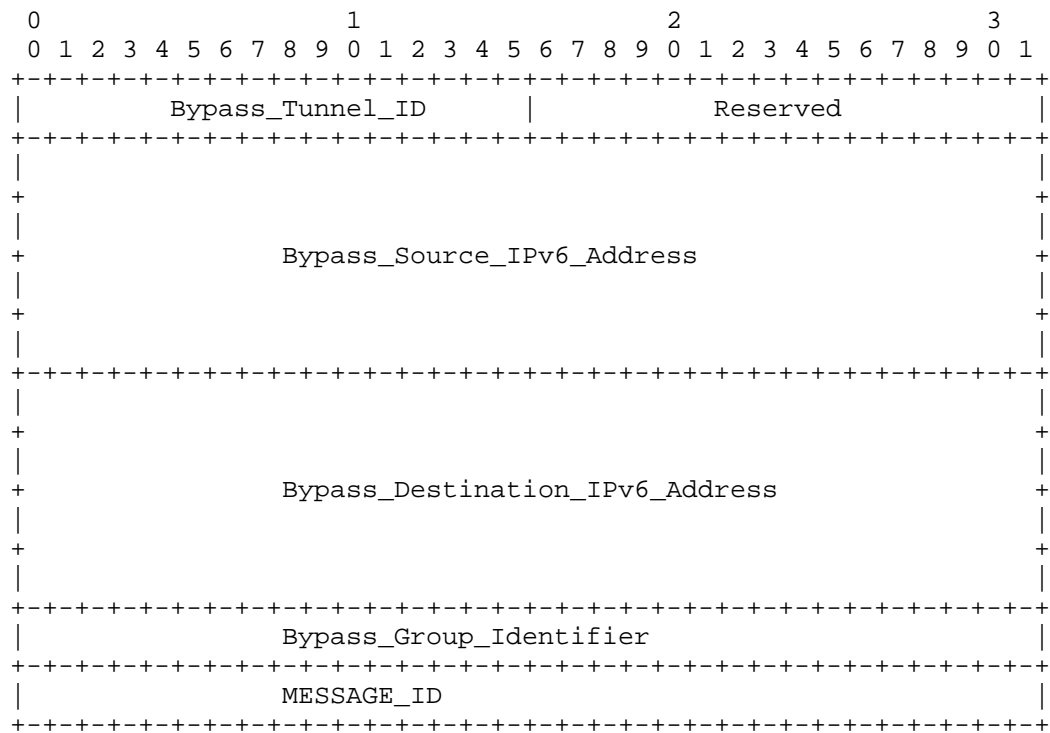


Figure 2: The IPv6 Extended Association ID field

Bypass\_Tunnel\_ID: 16 bits

The bypass tunnel identifier.

Reserved: 16 bits

Reserved for future use.

Bypass\_Source\_IPv6\_Address: 128 bits

The bypass tunnel source IPV6 address.

Bypass\_Destination\_IPv6\_Address: 128 bits

The bypass tunnel destination IPV6 address.

Bypass\_Group\_Identifier: 32 bits

The bypass tunnel group identifier.

MESSAGE\_ID

A MESSAGE\_ID object as defined by [RFC2961].

The PLR assigns a bypass tunnel and Bypass\_Group\_Identifier for each protected LSP. The same Bypass\_Group\_Identifier is used for the set of protected LSPs that share the same bypass tunnel and traverse the same egress link and are not already rerouted. The PLR also generates a MESSAGE\_ID object (flags SHOULD be clear, Epoch and Message\_Identifier MUST be set according to [RFC2961]).

The PLR MUST generate a new Message\_Identifier each time the contents of the B-SFRR Extended ASSOCIATION object change; for example, when PLR node changes the bypass tunnel assignment.

The PLR node notifies the MP node of the bypass tunnel assignment via adding a B-SFRR Extended ASSOCIATION object in the RSVP Path message for the protected LSP using procedures described in Section 3.2.

The MP node acknowledges the PLR node assignment by signaling the B-SFRR Extended Association object within the RSVP Resv message of the protected LSP. With exception of the MESSAGE\_ID objects, all other fields of the received B-SFRR Extended ASSOCIATION object in the RSVP Path message are copied into the B-SFRR Extended ASSOCIATION object to be added in the Resv message. The MESSAGE\_ID object is set according to [RFC2961] with the Flags being clear. A new Message\_Identifier MUST be used to acknowledge an updated PLR assignment.

The PLR considers the protected LSP as Summary FRR capable only if the B-SFRR Extended ASSOCIATION objects sent in the RSVP Path message and the one received in the RSVP Resv message (with exception of the MESSAGE\_ID) match. If it does not match, or if B-SFRR Extended Association object is absent in a subsequent refresh, the PLR node MUST consider the protected LSP as not Summary FRR capable.

### 3.1.2. PLR Summary FRR Signaling Procedure

The B-SFRR Extended ASSOCIATION object is added by each PLR in the RSVP Path message of the protected LSP to record the bypass tunnel assignment. This object is updated every time the PLR updates the bypass tunnel assignment (which triggers an RSVP Path change message).

Upon receiving an RSVP Resv message with B-SFRR Extended ASSOCIATION object, the PLR node checks if the expected subobjects in the B-SFRR Extended ASSOCIATION ID are present. If present, the PLR determines if the MP has acknowledged the current PLR assignment.

To be a valid acknowledgement, the received B-SFRR Extended ASSOCIATION object contents within the RSVP Resv message of the protected LSP MUST match the latest B-SFRR Extended ASSOCIATION object contents that the PLR node had sent within the RSVP Path message (with exception of the MESSAGE\_ID).

Note, when forwarding an RSVP Resv message upstream, the PLR node SHOULD remove any/all B-SFRR Extended ASSOCIATION objects whose Association Source matches the PLR node address.

### 3.1.3. MP Summary FRR Signaling Procedure

Upon receiving an RSVP Path message with an B-SFRR Extended ASSOCIATION object, the MP node processes all (there may be multiple PLRs for a single MP) B-SFRR Extended ASSOCIATION objects that have the MP node address as Bypass Destination address in the Association ID.

The MP node first ensures the existence of the bypass tunnel and that the Bypass\_Group\_Identifier is not already FRR active. That is, an LSP cannot join a group that is already FRR rerouted.

The MP node builds a mirrored Summary FRR Group database per PLR, which is determined using the Bypass\_Source\_Address field. The MESSAGE\_ID is extracted and recorded for the protected LSP PATH state. The MP node signals a B-SFRR Extended Association object within the RSVP Resv message of the protected LSP. With exception of the MESSAGE\_ID objects, all other fields of the received B-SFRR

Extended ASSOCIATION object in the RSVP Path message are copied into the B-SFRR Extended ASSOCIATION object to be added in the Resv message. The MESSAGE\_ID object is set according to [RFC2961] with the Flags being clear.

Note, an MP may receive more than one RSVP Path message with the B-SFRR Extended ASSOCIATION object from different upstream PLR node(s). In this case, the MP node is expected to save all the received MESSAGE\_IDs from the different upstream PLR node(s). After a failure, the MP node determines and activates the associated Summary Refresh ID to use once it receives and processes the RSVP Path message with FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object over the bypass LSP from the PLR.

When forwarding an RSVP Path message downstream, the MP SHOULD remove any/all B-SFRR Extended ASSOCIATION object(s) whose Association ID contains Bypass\_Destination\_Address matching the MP node address.

### 3.2. Signaling Procedures Post Failure

Upon detection of the fault (egress link or node failure) the PLR first performs the object modification procedures described by Section 6.4.3 of [RFC4090] for all affected protected LSPs. For Summary FRR LSPs assigned to the same bypass tunnel a common RSVP\_HOP and SENDER\_TEMPLATE MUST be used.

The PLR MUST signal non-Summary FRR enabled LSPs over the bypass tunnel before signaling the Summary FRR enabled LSPs. This is needed to allow for the case when the PLR node has recently changed a bypass assignment and the MP has not processed the change yet.

A new object FRR\_ACTIVE SUMMARY\_FRR\_BYPASS is defined in Section 3.2.1 and sent within the RSVP Path message of the bypass LSP to reroute RSVP state of Summary FRR enabled LSPs.

#### 3.2.1. SUMMARY\_FRR\_BYPASS Object

The SUMMARY\_FRR\_BYPASS Object with Type FRR\_ACTIVE is carried in the Path message of a bypass LSP. This object is added by the PLR node to indicate to the MP node (bypass tunnel destination) that one or more groups of protected LSPs that are being protected by the specified bypass tunnel are being rerouted over the bypass tunnel.

The FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object is assigned the C-Type (TBD-3). The FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object has the below format.

SUMMARY\_FRR\_BYPASS Class-Num = (TBD-2) (of the form 11bbbbbb) Class-Name = SUMMARY\_FRR\_BYPASS Class, FRR\_ACTIVE C-Type = (TBD-3)



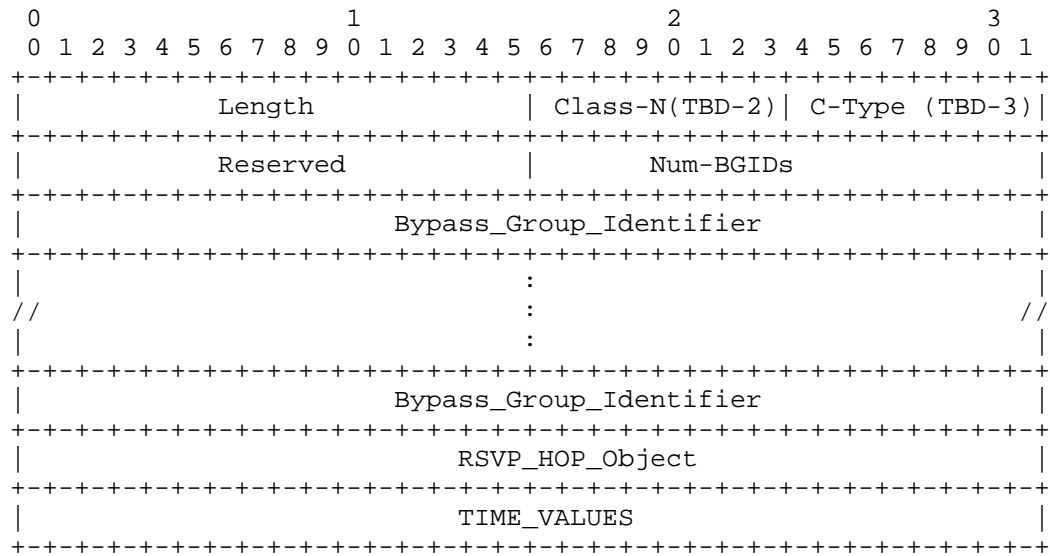


Figure 3: Summary FRR Bypass Object

Reserved: 16 bits

Reserved for future use.

Num-BGIDs: 16 bits

Number of Bypass\_Group\_Identifier fields.

Bypass\_Group\_Identifier: 32 bits

The Bypass\_Group\_Identifier that is previously advertised by the PLR using the Extended Association object. One or more Bypass\_Group\_Identifiers may be included.

RSVP\_HOP\_Object: Class 3, as defined by [RFC2205]

Replacement RSVP HOP object to be applied to all LSPs associated with each of the following Bypass\_Group\_Identifiers. This corresponds to C-Type = 1 for IPv4 RSVP HOP, or C-Type = 2 for IPv6 RSVP HOP depending on the IP address family carried within the object.

TIME\_VALUES object: Class 5, as defined by [RFC2205]

Replacement TIME\_VALUES object to be applied to all LSPs associated with each of the following Bypass\_Group\_Identifiers after receiving the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object.

### 3.2.2. PLR Summary FRR Signaling Procedure

After a failure event, when using the Summary FRR path signaling procedures, an individual RSVP Path message for each Summary FRR LSP is not signaled. Instead, to reroute Summary FRR LSPs via the bypass tunnel, the PLR adds the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object in the RSVP Path message of the RSVP session of the bypass tunnel.

The RSVP\_HOP\_Object field of the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object is set to the common RSVP\_HOP that was used by the PLR in Section 3.2 of this document.

The previously received MESSAGE\_ID from the MP is activated. As a result, the MP may refresh the protected rerouted RESV state using Summary Refresh procedures.

For each affected Summary FRR group, its Bypass\_Group\_Identifier is added to the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object.

### 3.2.3. MP Summary FRR Signaling Procedure

Upon receiving an RSVP Path message with a FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object, the MP performs normal merge point processing for each protected LSP associated with each Bypass\_Group\_Identifier, as if it received individual RSVP Path messages for the LSP.

For each Summary FRR LSP being merged, the MP first modifies the Path state as follows:

1. The RSVP\_HOP object is copied from the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS RSVP\_HOP\_Object field.
2. The TIME\_VALUES object is copied from the FRR\_ACTIVE SUMMARY\_FRR\_BYPASS TIMES\_VALUE field. The TIME\_VALUES object contains the refresh time of the PLR to generate refreshes and that would have exchanged in a Path message sent to the MP after the failure when no SFRR procedures are in effect.
3. The SENDER\_TEMPLATE object SrcAddress field is copied from the bypass tunnel SENDER\_TEMPLATE object. For the case where PLR is also the head-end, and SENDER\_TEMPLATE SrcAddress of the protected LSP and bypass tunnel are the same, the MP MUST use the modified HOP Address field instead.
4. The ERO object is modified as per Section 6.4.4. of [RFC4090]. Once the above modifications are completed, the MP then performs the merge processing as per [RFC4090].

5. The previously received MESSAGE\_ID from the PLR is activated, meaning that the PLR may now refresh the protected rerouted PATH state using Summary Refresh procedures.

A failure during merge processing of any individual rerouted LSP MUST result in an RSVP Path Error message.

An individual RSVP Resv message for each successfully merged Summary FRR LSP is not signaled. The MP node SHOULD immediately use Summary Refresh procedures to refresh the protected LSP RESV state.

### 3.3. Refreshing Summary FRR Active LSPs

Refreshing of Summary FRR active LSPs is performed using Summary Refresh as defined by [RFC2961].

## 4. Compatibility

The (Extended) ASSOCIATION object is defined in [RFC4872] with a class number in the form 1lbbbbbb, which ensures compatibility with non-supporting node(s). Such nodes will ignore the object and forward it without modification.

The new FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object is to be defined with a class number in the form 1lbbbbbb, which ensures compatibility with non-supporting nodes. Per [RFC2205], the nodes not supporting this extension will ignore the object but forward it, unexamined and unmodified, in all messages.

## 5. Security Considerations

This document updates an existing RSVP object, and introduces a new RSVP object. Thus, in the event of the interception of a signaling message, a slightly more information could be deduced about the state of the network than was previously the case. Existing mechanisms for maintaining the integrity and authenticity of RSVP protocol messages [RFC2747] can be applied.

## 6. IANA Considerations

IANA maintains the "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Parameters" registry (see <http://www.iana.org/assignments/gmpls-sig-parameters>). The "Association Type" subregistry is included in this registry.

This registry has been updated by new Association Type for Extended ASSOCIATION Object defined in this document as follows:

| Value | Name                                    | Reference     |
|-------|-----------------------------------------|---------------|
| ----- | -----                                   | -----         |
| TBD-1 | Bypass Summary FRR Association (B-SFRR) | Section 2.1.1 |

IANA also maintains and assigns the values for the RSVP-TE protocol parameters "Resource Reservation Protocol (RSVP) Parameters" (see <http://www.iana.org/assignments/rsvp-parameters>).

From this registry, a new RSVP Class (TBD-2) and of the form 11bbbbbb and a new C-Type (TBD-3) are requested for the new FRR\_ACTIVE SUMMARY\_FRR\_BYPASS object defined in this document.

Class-Number = (TBD-2), Class-Name = SUMMARY\_FRR\_BYPASS

C-Type = (TBD-3) Name = FRR\_ACTIVE

## 7. Acknowledgments

The authors would like to thank Loa Andersson, Lou Berger, Eric Osborne, Gregory Mirsky, and Mach Chen for reviewing and providing valuable comments to this document.

## 8. Contributors

Nicholas Tan  
Arista Networks

Email: [ntan@arista.com](mailto:ntan@arista.com)

## 9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <<http://www.rfc-editor.org/info/rfc2747>>.

- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<http://www.rfc-editor.org/info/rfc2961>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<http://www.rfc-editor.org/info/rfc4090>>.
- [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou, Ed., "RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007, <<http://www.rfc-editor.org/info/rfc4872>>.
- [RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP ASSOCIATION Object Extensions", RFC 6780, DOI 10.17487/RFC6780, October 2012, <<http://www.rfc-editor.org/info/rfc6780>>.

#### Authors' Addresses

Mike Taillon  
Cisco Systems, Inc.

Email: [mtaillon@cisco.com](mailto:mtaillon@cisco.com)

Tarek Saad (editor)  
Cisco Systems, Inc.

Email: [tsaad@cisco.com](mailto:tsaad@cisco.com)

Rakesh Gandhi  
Cisco Systems, Inc.

Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Abhishek Deshmukh  
Juniper Networks

Email: [adeshmukh@juniper.net](mailto:adeshmukh@juniper.net)

Markus Jork  
Juniper Networks

Email: [mjork@juniper.net](mailto:mjork@juniper.net)

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

CCAMP  
Internet Draft  
Intended status: Standards Track

J. Ahlberg  
Ericsson  
M. Ye  
Huawei  
X. Li  
NEC Laboratories Europe  
K. Kawada  
NEC Corporation  
CJ. Bernardos  
UC3M  
December 23, 2016

Expires: June 2017

## A YANG Data Model for Microwave Radio Link

draft-mwdt-ccamp-mw-yang-01

### Abstract

This document defines a YANG data model in order to control and manage the radio link interfaces, and the connectivity to packet (typically Ethernet) interfaces in a microwave/millimeter wave node.

### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 23, 2017.

### Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                          |    |
|------------------------------------------|----|
| 1. Terminology and Definitions .....     | 2  |
| 2. Introduction.....                     | 3  |
| 3. YANG Data Model (Tree Structure)..... | 3  |
| 4. YANG Module.....                      | 7  |
| 5. Security Considerations .....         | 63 |
| 6. IANA Considerations .....             | 63 |
| 7. References.....                       | 63 |
| 7.1. Normative References .....          | 63 |
| 7.2. Informative References .....        | 63 |

## 1. Terminology and Definitions

The following terms are used in this document:

Carrier Termination (CT) is an interface for the capacity provided over the air by a single carrier. It is typically defined by its transmitting and receiving frequencies.

Radio Link Terminal (RLT) is an interface providing packet capacity and/or TDM capacity to the associated Ethernet and/or TDM interfaces in a node and used for setting up a transport service over a microwave/millimeter wave link.

The following acronyms are used in this document:

ACM Adaptive Coding Modulation  
ATPC Automatic Transmit Power Control  
CM Coding Modulation  
CT Carrier Termination  
RLT Radio Link Terminal  
RTPC Remote Transmit Power Control  
XPIC Cross Polarization Interference Cancellation





## 2. Introduction

This document defines a YANG data model for management and control of the radio link interface(s) and the relationship to packet (typically Ethernet) and/or TDM interfaces in a microwave/millimeter wave node. The data model includes configuration and state data.

The design of the data model follows the framework for management and control of microwave and millimeter wave interface parameters defined in [mw-fmwk]. This framework identifies the need and the scope of the YANG data model, the use cases and requirements that the model needs to support. Moreover, it provides a detailed gap analysis to identify the missing parameters and functionalities of the existing and established models to support the specified use cases and requirements, and based on that recommends how the gaps should be filled with the development of the new model.

According to the conclusion of the gap analysis, the structure of the data model is based on the structure defined in [I-D.ahlberg-ccamp-microwave-radio-link] and it augments RFC 7223 to align with the same structure for management of the packet interfaces. More specifically, the model will include interface layering to manage the capacity provided by a radio link terminal for the associated Ethernet and TDM interfaces, using the principles for interface layering described in RFC 7223 as a basis.

The designed YANG data model uses the IETF: Radio Link Model [I-D.ahlberg-ccamp-microwave-radio-link] and the ONF: Microwave Modeling [ONF-model] as the basis for the definition of the detailed leafs/parameters, and proposes new ones to cover identified gaps which are analyzed in [mw-fmwk].

## 3. YANG Data Model (Tree Structure)



```

module: ietf-microwave-radio-link
 +--rw radio-link-protection-groups
 | +--rw radio-link-protection-group* [name]
 | | +--rw name string
 | | +--rw protection-architecture-type? identityref
 | | +--rw protection-operation-type? enumeration
 | | +--rw working-entity* if:interface-ref
 | | +--rw revertive-wait-to-restore? uint16
 | | +--rw radio-link-protection-members* if:interface-ref
 | | +---x protection-external-commands
 | | | +---w input
 | | | +---w protection-external-command? identityref
 | +--ro radio-link-protection-groups-state
 | | +--ro radio-link-protection-group* [name]
 | | | +--ro name string
 | | | +--ro protection-status? identityref
 | +--rw xpic-pairs {xpic}?
 | | +--rw xpic-pair* [name]
 | | | +--rw name string
 | | | +--rw enabled? boolean
 | | | +--rw xpic-members* if:interface-ref
 | +--rw mimo-groups {mimo}?
 | | +--rw mimo-group* [name]
 | | | +--rw name string
 | | | +--rw enabled? boolean
 | | | +--rw mimo-members* if:interface-ref
 augment /if:interfaces/if:interface:
 +--rw id? string
 +--rw mode identityref
 +--rw carrier-terminations* if:interface-ref
 +--rw rlp-groups* -> /radio-link-protection-groups
 | /radio-link-protection-group/name
 +--rw xpic-pairs* -> /xpic-pairs/xpic-pair/name {xpic}?
 +--rw mimo-group? -> /mimo-groups/mimo-group/name {mimo}?
 +--rw tdm-connections* [tdm-type] {tdm}?
 | +--rw tdm-type identityref
 | +--rw tdm-connections unit16
 augment /if:interfaces/if:interface:
 +--rw carrier-id? string
 +--rw tx-enabled? boolean
 +--rw tx-frequency uint32
 +--rw rx-frequency? uint32

```



```

+--rw rx-frequency-config? boolean
+--rw duplex-distance uint32
+--rw channel-separation decimal64
+--rw polarization? enumeration
+--rw power-mode enumeration
+--rw selected-output-power power
+--rw atpc-lower-threshold power
+--rw atpc-upper-threshold power
+--rw coding-modulation-mode enumeration
+--rw selected-cm identityref
+--rw selected-min-acm identityref
+--rw selected-max-acm identityref
+--rw if-loop? enumeration
+--rw rf-loop? enumeration
+--rw ct-performance-thresholds
 +--rw received-level-alarm-threshold? power
 +--rw transmitted-level-alarm-threshold? power
 +--rw ber-alarm-threshold? enumeration
augment /if:interfaces-state/if:interface:
+--ro tx-oper-status? enumeration
+--ro actual-transmitted-level? power
+--ro actual-received-level? power
+--ro actual-tx-cm? identityref
+--ro actual-snr? decimal64
+--ro actual-xpi? decimal64 {xpic}?
+--ro capabilities
 +--ro min-tx-frequency? uint32
 +--ro max-tx-frequency? uint32
 +--ro min-rx-frequency? uint32
 +--ro max-rx-frequency? uint32
 +--ro available-min-output-power? power
 +--ro available-max-output-power? power
 +--ro available-min-acm? identityref
 +--ro available-max-acm? identityref
augment /if:interfaces-state/if:interface/if:statistics:
+--ro bbe? yang:counter32
+--ro es? yang:counter32
+--ro ses? yang:counter32
+--ro uas? yang:counter32
+--ro min-rltm? power
+--ro max-rltm? power
+--ro min-tltm? power
+--ro max-tltm? power
```



## 4. YANG Module

```
<CODE BEGINS> file "ietf-microwave-radio-link.yang"

module ietf-microwave-radio-link {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-microwave-radio-link";
 prefix mrl;

 import ietf-yang-types {
 prefix yang;
 }

 import ietf-interfaces {
 prefix if;
 }

 import iana-if-type {
 prefix ianaift;
 }

 organization
 "IETF CCAMP Working Group";
 contact
 "jonas.ahlberg@ericsson.com
 amy.yemin@huawei.com
 Xi.Li@neclab.eu
 cjbc@it.uc3m.es
 k-kawada@ah.jp.nec.com";

 description
 "This is a module for the entities in a generic
 microwave system.";

 revision 2016-12-22 {
 description
 "Draft revision covering a complete scope for
 configuration and state data for radio link
 interfaces.";

 reference "";
 }
}
```





```
revision 2016-10-29 {
 description
 "Draft revision.";
 reference "";
}

/*
 * Features
 */

feature xpic {
 description
 "Indicates that the device supports XPIC.";
}

feature mimo {
 description
 "Indicates that the device supports MIMO.";
}

feature tdm {
 description
 "Indicates that the device supports TDM.";
}

/*
 * Interface identities
 */

identity radio-link-terminal {
 base ianaift:iana-interface-type;
 description
 "Interface identity for a radio link terminal.";
}

identity carrier-termination {
 base ianaift:iana-interface-type;
 description
 "Interface identity for a carrier termination.";
}

/*
 * Radio-link-terminal mode identities
 */
```



```
identity rlt-mode {
 description
 "A description of the mode in which the radio
 link terminal is configured. The format is X
 plus Y. X represent the number of bonded
 carrier terminations. Y represent the number
 of protecting carrier terminations.";
}

identity one-plus-zero {
 base rlt-mode;
 description
 "1 carrier termination only.";
}

identity one-plus-one {
 base rlt-mode;
 description
 "1 carrier termination
 and 1 protecting carrier termination.";
}

identity two-plus-zero {
 base rlt-mode;
 description
 "2 bonded carrier terminations.";
}

/*
* Coding and modulation identities
*/

identity coding-modulation {
 description
 "The coding and modulation schemes.";
}

identity half-bpsk-strong {
 base coding-modulation;
 description
 "Half BPSK strong coding and modulation scheme.";
}
```



```
identity half-bpsk {
 base coding-modulation;
 description
 "Half BPSK coding and modulation scheme.";
}

identity half-bpsk-light {
 base coding-modulation;
 description
 "Half BPSK light coding and modulation scheme.";
}

identity bpsk-strong {
 base coding-modulation;
 description
 "BPSK strong coding and modulation scheme.";
}

identity bpsk {
 base coding-modulation;
 description
 "BPSK coding and modulation scheme.";
}

identity bpsk-light {
 base coding-modulation;
 description
 "BPSK light coding and modulation scheme.";
}

identity qpsk {
 base coding-modulation;
 description
 "QPSK coding and modulation scheme.";
}

identity qam-4-strong {
 base coding-modulation;
 description
 "4 QAM strong coding and modulation scheme.";
}

identity qam-4 {
 base coding-modulation;
 description
 "4 QAM coding and modulation scheme.";
}
```



```
identity qam-4-light {
 base coding-modulation;
 description
 "4 QAM light coding and modulation scheme.";
}

identity qam-16-strong {
 base coding-modulation;
 description
 "16 QAM strong coding and modulation scheme.";
}

identity qam-16 {
 base coding-modulation;
 description
 "16 QAM coding and modulation scheme.";
}

identity qam-16-light {
 base coding-modulation;
 description
 "16 QAM light coding and modulation scheme.";
}

identity qam-32-strong {
 base coding-modulation;
 description
 "32 QAM strong coding and modulation scheme.";
}

identity qam-32 {
 base coding-modulation;
 description
 "32 QAM coding and modulation scheme.";
}

identity qam-32-light {
 base coding-modulation;
 description
 "32 QAM light coding and modulation scheme.";
}

identity qam-64-strong {
 base coding-modulation;
 description
 "64 QAM strong coding and modulation scheme.";
}
```





```
identity qam-64 {
 base coding-modulation;
 description
 "64 QAM coding and modulation scheme.";
}

identity qam-64-light {
 base coding-modulation;
 description
 "64 QAM light coding and modulation scheme.";
}

identity qam-128-strong {
 base coding-modulation;
 description
 "128 QAM strong coding and modulation scheme.";
}

identity qam-128 {
 base coding-modulation;
 description
 "128 QAM coding and modulation scheme.";
}

identity qam-128-light {
 base coding-modulation;
 description
 "128 QAM light coding and modulation scheme.";
}

identity qam-256-strong {
 base coding-modulation;
 description
 "256 QAM strong coding and modulation scheme.";
}

identity qam-256 {
 base coding-modulation;
 description
 "256 QAM coding and modulation scheme.";
}

identity qam-256-light {
 base coding-modulation;
 description
 "256 QAM light coding and modulation scheme.";
}
```



```
identity qam-512-strong {
 base coding-modulation;
 description
 "512 QAM strong coding and modulation scheme.";
}

identity qam-512 {
 base coding-modulation;
 description
 "512 QAM coding and modulation scheme.";
}

identity qam-512-light {
 base coding-modulation;
 description
 "512 QAM light coding and modulation scheme.";
}

identity qam-1024-strong {
 base coding-modulation;
 description
 "1024 QAM strong coding and modulation scheme.";
}

identity qam-1024 {
 base coding-modulation;
 description
 "1024 QAM coding and modulation scheme.";
}

identity qam-1024-light {
 base coding-modulation;
 description
 "1024 QAM light coding and modulation scheme.";
}

identity qam-2048-strong {
 base coding-modulation;
 description
 "2048 QAM strong coding and modulation scheme.";
}

identity qam-2048 {
 base coding-modulation;
 description
 "2048 QAM coding and modulation scheme.";
}
```



```
identity qam-2048-light {
 base coding-modulation;
 description
 "2048 QAM light coding and modulation scheme.";
}

identity qam-4096-strong {
 base coding-modulation;
 description
 "4096 QAM strong coding and modulation scheme.";
}

identity qam-4096 {
 base coding-modulation;
 description
 "4096 QAM coding and modulation scheme.";
}

identity qam-4096-light {
 base coding-modulation;
 description
 "4096 QAM light coding and modulation scheme.";
}

/*
 * Protection architecture type identities
 */
identity protection-architecture-type {
 description
 "protection architecture type";
}

identity one-plus-one-type {
 base protection-architecture-type;
 description
 "One carrier termination and
 one protecting carrier termination.";
}

identity one-to-n-type {
 base protection-architecture-type;
 description
 "One carrier termination protecting
 n other carrier terminations.";
}
```



```
/*
 * Protection states identities
 */

identity protection-states {
 description
 "Identities describing the status of the protection,
 in a group of carrier terminations configured in
 a radio link protection mode.";
}

identity unprotected {
 base protection-states;
 description "Not protected";
}

identity protected {
 base protection-states;
 description "Protected";
}

identity unable-to-protect {
 base protection-states;
 description "Unable to protect";
}

/*
 * protection-external-commands identities
 */

identity protection-external-commands{
 description
 "Protection external commands for trouble shooting
 purpose.";
}

identity manual-switch{
 base protection-external-commands;
 description
 "A switch action initiated by an operator command.
 It switches normal traffic signal to the protection
 transport entity.";
}
```





```
/*
 * TDM-type identities
 */
identity tdm-type {
 description
 "A description of the type of TDM connection,
 also indicating the supported capacity of the
 connection.";
}

identity E1 {
 base tdm-type;
 description
 "E1 connection, 2,048 Mbit/s.";
}

identity STM-1 {
 base tdm-type;
 description
 "STM-1 connection, 155,52 Mbit/s.";
}

/*
 * Typedefs
 */
typedef power {
 type decimal64 {
 fraction-digits 1;
 }
 description
 "Type used for power values, selected and measured.";
}

/*
 * Radio Link Terminal (RLT) - Configuration data nodes
 */
augment "/if:interfaces/if:interface" {
 when "if:type = 'mrl:radio-link-terminal'";
 description
 "Addition of data nodes for radio link terminal to
 the standard Interface data model, for interfaces of
 the type 'radio-link-terminal'.";
}
```



```
leaf id {
 type string;
 default "";
 description
 "ID of the radio link terminal. Used by far-end when
 checking that it's connected to the correct RLT.";
}

leaf mode {
 type identityref {
 base rlt-mode;
 }
 mandatory true;
 description
 "A description of the mode in which the radio link
 terminal is configured. The format is X plus Y.
 X represent the number of bonded carrier terminations.
 Y represent the number of protecting carrier
 terminations.";
}

leaf-list carrier-terminations {
 type if:interface-ref;
 must "/if:interfaces/if:interface[if:name = current()]"
 + "/if:type = 'mrl:carrier-termination'" {
 description
 "The type of interface must be
 'carrier-termination'.";
 }
 min-elements 1;
 description
 "A list of references to carrier terminations
 included in the radio link terminal.";
}

leaf-list rlp-groups {
 type leafref {
 path "/mrl:radio-link-protection-groups/"
 + "mrl:radio-link-protection-group/mrl:name";
 }
 description
 "A list of references to the carrier termination
 groups configured for radio link protection in this
 radio link terminal.";
}
```



```
leaf-list xp-pic-pairs {
 if-feature xp-pic;
 type leafref {
 path "/mrl:xp-pic-pairs/mrl:xp-pic-pair/mrl:name";
 }
 description
 "A list of references to the XPIC pairs used in this
 radio link terminal. One pair can be used by two
 terminals.";
}

leaf mimo-group {
 if-feature mimo;
 type leafref {
 path "/mrl:mimo-groups/mrl:mimo-group/mrl:name";
 }
 description
 "A reference to the MIMO group used in this
 radio link terminal. One group can be used by more
 than one terminal.";
}

list tdm-connections {
 if-feature tdm;
 key "tdm-type";
 description
 "A list stating the number of TDM connections of a
 specified tdm-type that is supported by the RLT.";
 leaf tdm-type {
 type identityref {
 base tdm-type;
 }
 description
 "The type of TDM connection, which also indicates
 the supported capacity.";
 }

 leaf tdm-connections {
 type uint16;
 mandatory true;
 description "Number of connections of the specified type.";
 }
}
}
```



```
/*
 * Carrier Termination - Configuration data nodes
 */

augment "/if:interfaces/if:interface" {
 when "if:type = 'mrl:carrier-termination'";
 description
 "Addition of data nodes for carrier termination to
 the standard Interface data model, for interfaces
 of the type 'carrier-termination'.";
 leaf carrier-id {
 type string;
 default "A";
 description
 "ID of the carrier. (e.g. A, B, C or D)
 Used in XPIC & MIMO configurations to check that
 the carrier termination is connected to the correct
 far-end carrier termination. Should be the same
 carrier ID on both sides of the hop.
 Defaulted when not MIMO or XPIC.";
 }

 leaf tx-enabled {
 type boolean;
 default "false";
 description
 " Disables (false) or enables (true) the
 transmitter. Only applicable when the interface
 is enabled (interface:enabled = true) otherwise
 it's always disabled.";
 }

 leaf tx-frequency {
 type uint32;
 units "kHz";
 mandatory true;
 description
 "Selected transmitter frequency.";
 }
}
```





```
leaf rx-frequency {
 type uint32;
 units "kHz";
 description
 "Selected receiver frequency.
 Mandatory and writeable when rx-frequency-config=true.
 Otherwise read-only and calculated from tx-frequency
 and duplex-distance.";
}

leaf rx-frequency-config {
 type boolean;
 default "true";
 description
 "Enable (true) or disable (false) direct
 configuration of rx-frequency and instead
 using a defined duplex distance.";
}

leaf duplex-distance {
 when "../rx-frequency-config = 'false'";
 type uint32;
 units "kHz";
 mandatory true;
 description
 "Distance between Tx & Rx frequencies.
 Used to calculate rx-frequency when
 rx-frequency-config=false.";
}

leaf channel-separation {
 type decimal64 {
 fraction-digits 1;
 }
 units "MHz";
 mandatory true;
 description
 "The amount of bandwidth allocated to a carrier.";
}

leaf polarization {
 type enumeration {
 enum "horizontal" {
 description "Horizontal polarization.";
 }
 }
}
```



```

 enum "vertical" {
 description "Vertical polarization.";
 }
 enum "not-specified" {
 description "Polarization not specified.";
 }
}
default "not-specified";
description
 "Polarization - A textual description for info only.";
}

leaf power-mode {
 type enumeration {
 enum rtpc {
 description "Remote Transmit Power Control (RTPC).";
 }
 enum atpc {
 description "Automatic Transmit Power Control
 (ATPC).";
 }
 }
 mandatory true;
 description
 "A choice of Remote Transmit Power Control (RTPC)
 or Automatic Transmit Power Control (ATPC).";
}

leaf selected-output-power {
 type power {
 range "-99..40";
 }
 units "dBm";
 mandatory true;
 description
 "Selected output power in RTPC mode and selected
 maximum selected maximum output power in ATPC mode.
 Minimum output power in ATPC mode is the same as the
 system capability, available-min-output-power.";
}

leaf atpc-lower-threshold {
 when "../power-mode = 'atpc'";
 type power {
 range "-99..-30";
 }
 units "dBm";

```



```
 mandatory true;
 description
 "The lower threshold for the input power at far-end used in
 the ATPC mode.";
}

leaf atpc-upper-threshold {
 when "../power-mode = 'atpc'";
 type power {
 range "-99..-30";
 }
 units "dBm";
 mandatory true;
 description
 "The upper threshold for the input power
 at far-end used in the ATPC mode.";
}

leaf coding-modulation-mode {
 type enumeration {
 enum fixed {
 description "Fixed coding/modulation.";
 }
 enum adaptive {
 description "Adaptive coding/modulation.";
 }
 }
 mandatory true;
 description
 "A selection of fixed or
 adaptive coding/modulation mode.";
}

leaf selected-cm {
 when "../coding-modulation-mode = 'fixed'";
 type identityref {
 base coding-modulation;
 }
 mandatory true;
 description
 "Selected fixed coding/modulation.";
}
```



```
leaf selected-min-acm {
 when "../coding-modulation-mode = 'adaptive'";
 type identityref {
 base coding-modulation;
 }
 mandatory true;
 description
 "Selected minimum coding/modulation.
 Adaptive coding/modulation shall not go
 below this value.";
}

leaf selected-max-acm {
 when "../coding-modulation-mode = 'adaptive'";
 type identityref {
 base coding-modulation;
 }
 mandatory true;
 description
 "Selected maximum coding/modulation.
 Adaptive coding/modulation shall not go
 above this value.";
}

leaf if-loop {
 type enumeration {
 enum disabled {
 description "Disables the IF Loop.";
 }
 enum client {
 description "Loops the signal back to the client side.";
 }
 enum radio {
 description "Loops the signal back to the radio side.";
 }
 }
 default "disabled";
 description
 "Enable (client/radio) or disable (disabled) the IF loop,
 which loops the signal back to the client side or the
 radio side.";
}
```





```
leaf rf-loop {
 type enumeration {
 enum disabled {
 description "Disables the RF Loop.";
 }
 enum client {
 description "Loops the signal back to the client side.";
 }
 enum radio {
 description "Loops the signal back to the radio side.";
 }
 }
 default "disabled";
 description
 "Enable (client/radio) or disable (disabled) the RF loop,
 which loops the signal back to the client side or
 the radio side.";
}

container ct-performance-thresholds {
 description
 "Specification of thresholds for when alarms should
 be sent and cleared for various performance counters.";

 leaf received-level-alarm-threshold {
 type power {
 range "-99..-30";
 }
 units "dBm";
 default "-99";
 description
 "Specification of at which received power level an alarm
 should be raised.";
 }

 leaf transmitted-level-alarm-threshold {
 type power {
 range "-99..40";
 }
 units "dBm";
 default "-99";
 description
 "An alarm is sent when the transmitted power level
 is below the specified threshold.";
 }
}
```



```
 leaf ber-alarm-threshold {
 type enumeration {
 enum "10e-9" {
 description "Threshold at 10e-9.";
 }
 enum "10e-8" {
 description "Threshold at 10e-8.";
 }
 enum "10e-7" {
 description "Threshold at 10e-7.";
 }
 enum "10e-6" {
 description "Threshold at 10e-6.";
 }
 enum "10e-5" {
 description "Threshold at 10e-5.";
 }
 enum "10e-4" {
 description "Threshold at 10e-4.";
 }
 enum "10e-3" {
 description "Threshold at 10e-3.";
 }
 enum "10e-2" {
 description "Threshold at 10e-2.";
 }
 enum "10e-1" {
 description "Threshold at 10e-1.";
 }
 }
 default "10e-6";
 description
 "Specification of at which BER an alarm should
 be raised.";
 }
 }
}

/*
 * Radio Link Terminal - Operational state data nodes
 * Currently nothing in addition to the general
 * interface-state model.
 */

/*
 * Carrier Termination - Operational state data nodes
 */
```



```
augment "/if:interfaces-state/if:interface" {
 when "if:type = 'mrl:carrier-termination'";
 description
 "Addition of state data nodes for carrier termination to
 the standard Interface state data model, for interfaces
 of the type 'carrier-termination'.";

 leaf tx-oper-status {
 type enumeration {
 enum "off" {
 description "Transmitter is off.";
 }
 enum "on" {
 description "Transmitter is on.";
 }
 enum "standby" {
 description "Transmitter is in standby.";
 }
 }
 description
 "Shows the operative status of the transmitter.";
 }

 leaf actual-transmitted-level {
 type power {
 range "-99..40";
 }
 units "dBm";
 description
 "Actual transmitted power level (0.1 dBm resolution).";
 }

 leaf actual-received-level {
 type power {
 range "-99..-20";
 }
 units "dBm";
 description
 "Actual received power level (0.1 dBm resolution).";
 }

 leaf actual-tx-cm {
 type identityref {
 base coding-modulation;
 }
 description
 "Actual coding/modulation in transmitting direction.";
 }
}
```



```
leaf actual-snr {
 type decimal64 {
 fraction-digits 1;
 range "0..99";
 }
 units "dB";
 description
 "Actual signal to noise plus interference ratio.
 (0.1 dB resolution).";
}

leaf actual-xpi {
 if-feature xpica;
 type decimal64 {
 fraction-digits 1;
 range "0..99";
 }
 units "dB";
 description
 "The actual carrier to cross-polar interference.
 Only valid if XPIC is enabled. (0.1 dB resolution).";
}

container capabilities {
 description
 "Capabilities of the the installed equipment and
 some selected configurations.";

 leaf min-tx-frequency {
 type uint32;
 units "kHz";
 description
 "Minimum Tx frequency possible to use.";
 }

 leaf max-tx-frequency {
 type uint32;
 units "kHz";
 description
 "Maximum Tx frequency possible to use.";
 }
}
```





```

 leaf min-rx-frequency {
 type uint32;
 units "kHz";
 description
 "Minimum Rx frequency possible to use.";
 }

 leaf max-rx-frequency {
 type uint32;
 units "kHz";
 description
 "Maximum Tx frequency possible to use.";
 }

 leaf available-min-output-power {
 type power;
 units "dBm";
 description
 "The minimum output power supported.";
 }

 leaf available-max-output-power {
 type power;
 units "dBm";
 description
 "The maximum output power supported.";
 }

 leaf available-min-acm {
 type identityref {
 base coding-modulation;
 }
 description
 "Minimum coding-modulation possible to use.";
 }

 leaf available-max-acm {
 type identityref {
 base coding-modulation;
 }
 description
 "Maximum coding-modulation possible to use.";
 }
 }
}

```



```

Internet-Draft Microwave YANG Model December 2016
augment "/if:interfaces-state/if:interface/if:statistics" {
 when "../if:type = 'mrl:carrier-termination'";
 description
 "Addition of state data nodes in the container statistics
 for carrier terminations to the standard Interface data
 model, for interfaces of the type 'carrier-termination'.";

 leaf bbe {
 type yang:counter32;
 units "number of block errors";
 description
 "Number of Background Block Errors (BBE) during the
 interval. A BBE is an errored block not occurring as
 part of an SES.";
 }

 leaf es {
 type yang:counter32;
 units "seconds";
 description
 "Number of Errored Seconds (ES) since last
 reset. An ES is a one-second period with
 one or more errored blocks or at least one
 defect.";
 }

 leaf ses {
 type yang:counter32;
 units "seconds";
 description
 "Number of Severely Errored Seconds (SES) during the
 interval. SES is a one-second period which contains
 equal or more than 30% errored blocks or at least
 one defect. SES is a subset of ES.";
 }

 leaf uas {
 type yang:counter32;
 units "seconds";
 description
 "Number of Unavailable Seconds (UAS), that is, the
 total time that the node has been unavailable during
 a fixed measurement interval.";
 }
}

```



```

 leaf min-rltm {
 type power {
 range "-99..-20";
 }
 units "dBm";
 description
 "Minimum received power level since last reset.";
 }

 leaf max-rltm {
 type power {
 range "-99..-20";
 }
 units "dBm";
 description
 "Maximum received power level since last reset.";
 }

 leaf min-tltm {
 type power {
 range "-99..40";
 }
 units "dBm";
 description
 "Minimum transmitted power level since last reset.";
 }

 leaf max-tltm {
 type power {
 range "-99..40";
 }
 units "dBm";
 description
 "Maximum transmitted power level since last reset.";
 }
 }

/*
 * Radio Link Protection Groups - Configuration data nodes
 */

container radio-link-protection-groups {
 description
 "Configuration of radio link protected groups
 (1+1) of carrier terminations in a radio link.
 More than one protected group per radio-link-terminal
 is allowed.";

```



```
list radio-link-protection-group {
 key "name";
 description
 "List of protected groups of carrier terminations
 in a radio link.";

 leaf name {
 type string;
 description
 "Name used for identification of the radio
 link protection group";
 }

 leaf protection-architecture-type {
 type identityref{
 base protection-architecture-type;
 }
 default "one-plus-one-type";
 description
 "The type of protection architecture
 used, e.g. one carrier termination
 protecting one carrier termination.";
 }

 leaf protection-operation-type {
 type enumeration {
 enum "non-revertive" {
 description
 "In non revertive operation, the
 traffic does not return to the
 working carrier termination if the
 switch requests are terminated. ";
 }
 enum "revertive" {
 description
 "In revertive operation, the
 traffic always returns to (or
 remains on) the working carrier
 termination if the switch requests
 are terminated. ";
 }
 }
 default "non-revertive";
 description
 "The type of protection operation, i.e.
 revertive or non-revertive operation.";
 }
}
```





```
leaf-list working-entity {
 when "../protection-operation-type = 'revertive'";
 type if:interface-ref;
 must "/if:interfaces/if:interface[if:name = current()]"
 + "/if:type = 'mrl:carrier-termination'" {
 description
 "The type of a working-entity must be
 'carrier-termination'.";
 }
 min-elements 1;
 description
 "The carrier terminations over which the
 traffic normally should be transported
 over when there is no need to use the
 protecting carrier termination.";
}

leaf revertive-wait-to-restore {
 when "../protection-operation-type = 'revertive'";
 type uint16;
 units "seconds";
 default "0";
 description
 "The time to wait before switching back
 to the working carrier termination if
 protection-operation-type is revertive.";
}

leaf-list radio-link-protection-members {
 type if:interface-ref;
 must "/if:interfaces/if:interface[if:name = current()]"
 + "/if:type = 'mrl:carrier-termination'" {
 description
 "The type of a protection member must
 be 'carrier-termination'.";
 }
 min-elements 2;
 description
 "Association to a group of carrier
 terminations configured for radio link
 protection and used in the radio link terminal.";
}
```



```

 action protection-external-commands {
 input {
 leaf protection-external-command {
 type identityref {
 base protection-external-commands;
 }
 description
 "Execution of protection external
 commands for trouble shooting purpose.";
 }
 }
 }
}

/*
 * Radio Link Protection - Operational state data nodes
 */
container radio-link-protection-groups-state {
 config false;
 description
 "State data for radio link protected groups
 of carrier terminations in a radio link.";
 list radio-link-protection-group {
 key "name";
 description
 "List of protected groups of carrier
 terminations in a radio link.";

 leaf name {
 type string;
 description
 "Name used for identification of the
 radio link protection group.";
 }

 leaf protection-status {
 type identityref {
 base protection-states;
 }
 description
 "Status of the protection, in a group of
 carrier terminations configured in a
 radio link protection mode.";
 }
 }
}

```



```

/*
 * XPIC & MIMO - Configuration data nodes
 */

container xpic-pairs {
 if-feature xpic;
 description
 "Configuration of carrier termination pairs
 for operation in XPIC mode.";

 list xpic-pair {
 key "name";
 description
 "List of carrier termination pairs in XPIC mode.";

 leaf name {
 type string;
 description
 "Name used for identification of the XPIC pair.";
 }
 leaf enabled {
 type boolean;
 default "false";
 description
 "Enable(true)/disable(false) XPIC";
 }
 }

 leaf-list xpic-members {
 type if:interface-ref;
 must "/if:interfaces/if:interface[if:name = current()]"
 + "/if:type = 'mrl:carrier-termination'" {
 description
 "The type of a xpic-member must be
 'carrier-termination'.";
 }
 min-elements 2;
 max-elements 2;
 description
 "Association to XPIC pairs used in the
 radio link terminal.";
 }
}

```



```
container mimo-groups {
 if-feature mimo;
 description
 "Configuration of carrier terminations
 for operation in MIMO mode.";

 list mimo-group {
 key "name";
 description
 "List of carrier terminations in MIMO mode.";

 leaf name {
 type string;
 description
 "Name used for identification of the MIMO group.";
 }

 leaf enabled {
 type boolean;
 default "false";
 description
 "Enable(true)/disable(false) MIMO";
 }
 }

 leaf-list mimo-members {
 type if:interface-ref;
 must "/if:interfaces/if:interface[if:name = current()]"
 + "/if:type = 'mrl:carrier-termination'" {
 description
 "The type of a mimo-member must be
 'carrier-termination'.";
 }
 min-elements 2;
 description
 "Association to a MIMO group if used in
 the radio link terminal.";
 }
}
}
}
<CODE ENDS>
```





## 5. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., <edit-config>) to these data nodes without proper protection can have a negative effect on network operations.

The security considerations of [RFC7223] also apply to this document.

## 6. IANA Considerations

TBD.

## 7. References

### 7.1. Normative References

[RFC7223] Bjorklund M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.

### 7.2. Informative References

[draft-ietf-ccamp-microwave-framework-00]  
Ahlberg J., et al., " A framework for Management and Control of microwave and millimeter wave interface parameters ", draft-ietf-ccamp-microwave-framework-00 (work in progress), December 2016.



[I-D.ahlberg-ccamp-microwave-radio-link]

Ahlberg J., Carlson J., Lund H., Olausson T.,  
Ye M., and Vaupotic M., "Microwave Radio Link  
YANG Data Models", draft-ahlberg-ccamp-  
microwave-radio-link-01 (work in progress),  
May 2016.

[ONF-model]

"Microwave Modeling - ONF Wireless Transport  
Group", May 2016.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J.,  
and A.Bierman, "Network Configuration Protocol  
(NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol  
over SecureShell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network  
Configuration Protocol (NETCONF) Access Control  
Model", RFC 6536, March 2012.



Authors' Addresses

Jonas Ahlberg  
Ericsson AB  
Lindholmspiren 11  
Goeteborg 417 56  
Sweden

Email: [jonas.ahlberg@ericsson.com](mailto:jonas.ahlberg@ericsson.com)

Ye Min  
Huawei Technologies Co., Ltd.  
No.1899, Xiyuan Avenue  
Chengdu 611731  
P.R.China

Email: [amy.yemin@huawei.com](mailto:amy.yemin@huawei.com)

Xi Li  
NEC Laboratories Europe  
Kurfursten-Anlage 36  
69115 Heidelberg  
Germany

Email: [Xi.Li@neclab.eu](mailto:Xi.Li@neclab.eu)

Koji Kawada  
NEC Corporation  
1753, Shimonumabe Nakahara-ku  
Kawasaki, Kanagawa 211-8666  
Japan

Email: [k-kawada@ah.jp.nec.com](mailto:k-kawada@ah.jp.nec.com)

Carlos J. Bernardos  
Universidad Carlos III de Madrid  
Av. Universidad, 30  
Leganes, Madrid 28911  
Spain

Email: [cjbc@it.uc3m.es](mailto:cjbc@it.uc3m.es)

CCAMP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 12, 2017

X. Zhang  
K. Xiang  
Huawei Technologies  
A. Sharma  
R. Rao  
Infinera  
March 11, 2017

OTN Tunnel YANG Model  
draft-sharma-ccamp-otn-tunnel-model-01

Abstract

This document describes the YANG data model for OTN Tunnels.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 12, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                        |    |
|--------------------------------------------------------|----|
| 1. Introduction . . . . .                              | 2  |
| 2. Terminology and Notations . . . . .                 | 2  |
| 3. Model Overview . . . . .                            | 3  |
| 3.1. Mux Service in Multi-Domain OTN Network . . . . . | 3  |
| 3.2. Bookended and Non-BookEnded OTN Tunnel . . . . .  | 4  |
| 3.3. Network and Client side tunnel services . . . . . | 4  |
| 3.4. OTN Tunnel YANG Tree . . . . .                    | 4  |
| 3.5. OTN Tunnel YANG Code . . . . .                    | 5  |
| 3.6. Transport Types YANG Code . . . . .               | 10 |
| 4. Security Considerations . . . . .                   | 19 |
| 5. IANA Considerations . . . . .                       | 19 |
| 6. Acknowledgements . . . . .                          | 19 |
| 7. Normative References . . . . .                      | 19 |
| Authors' Addresses . . . . .                           | 19 |

## 1. Introduction

OTN transport networks can carry various types of client services. In many cases, the client signal is carried over an OTN tunnel across connected domains in a multi-domain network. These OTN services can either be transported or switched in the OTN network. If an OTN tunnel is switched, then additional parameters need to be provided to create a Mux OTN service.

This document provides YANG model for creating OTN tunnel. The model augments the TE Tunnel model, which is an abstract model to create TE Tunnels.

## 2. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in . They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).



- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

### 3. Model Overview

#### 3.1. Mux Service in Multi-Domain OTN Network

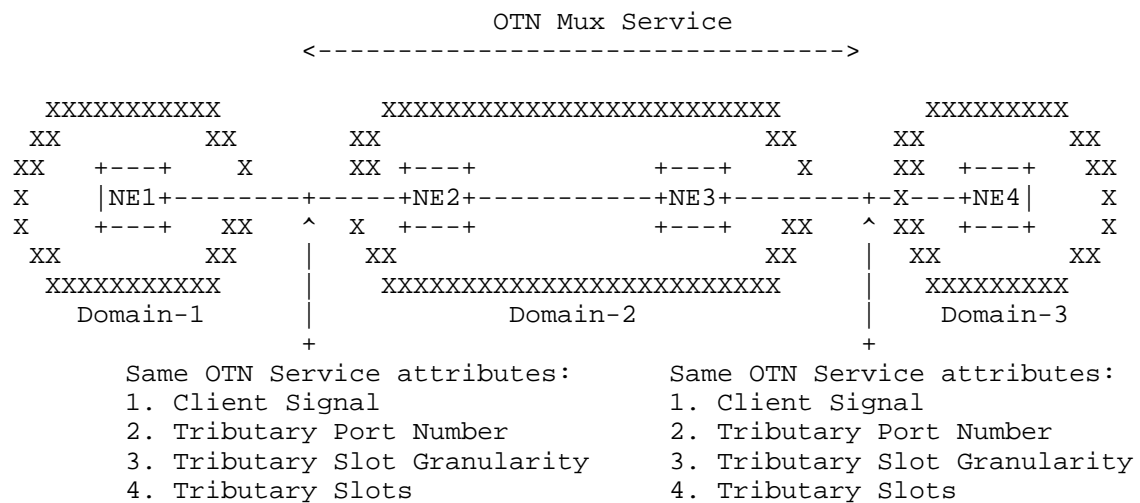


Figure 1: OTN Mux Service in a multi-domain network topology

Figure 1 shows a multi-domain OTN network with three domains. In this example, user wants to setup an end-to-end OTN service that passes through Domain-2. In order to create an OTN mux service in Domain-2, user will need to specify the exact details of the client side LO-ODU on NE2 and NE3, so that these service endpoints can be paired with the LO-ODU endpoints on NE1 and NE4, respectively.

Let's assume that ODU4 is the client side HO-ODU on NE2 and NE3, and the client signal is ODU2. User will need to specify the OTN client signal (ODU2 in this example), the Tributary Port Number (TPN), Tributary Slot Granularities (TSG) and tributary slots to be used.

As shown in the figure above, these service parameters must be the same between NE1 and NE2, and NE3 and NE4.

Once the OTN Mux service is setup in Domain-2, the incoming signal from either NE1 and/or NE4 will be switched inside Domain-2, and delivered to NE at the other end.

### 3.2. Bookended and Non-BookEnded OTN Tunnel

OTN tunnel model provides support for both bookended and non-bookended OTN tunnels.

For bookended tunnels, the same client signal is present on source and destination endpoints. For example, ODU2e bookended tunnel will have the same ODU2e client signal at both source and destination endpoints.

For non-bookended tunnels, different client signals are present on source and destination endpoints. For example, the client signal can be ODU2e on the source endpoint and the handoff at the destination can be 10GbE-LAN client signal.

### 3.3. Network and Client side tunnel services

The OTN tunnel model provides support for both network to network and client to client tunnels. For network to network tunnel, network termination points on source and destination node represent source and destination endpoints. For client to client tunnel, client termination points on source and destination node represent source and destination endpoints.

If a client to client tunnel needs to use one or more HO (or server) network to network tunnels, ERO and routing constraints, defined in the base TE model, can be used to route the client tunnel over one or more server tunnels.

### 3.4. OTN Tunnel YANG Tree

```

module: ietf-otn-tunnel
augment /te:te/te:tunnels/te:tunnel/te:config:
 +--rw payload-treatment? enumeration
 +--rw src-client-signal? identityref
 +--rw src-tpn? uint16
 +--rw src-tsg? identityref
 +--rw src-tributary-slot-count? uint16
 +--rw src-tributary-slots
 | +--rw values* uint8
 +--rw dst-client-signal? identityref
 +--rw dst-tpn? uint16
 +--rw dst-tsg? identityref
 +--rw dst-tributary-slot-count? uint16
 +--rw dst-tributary-slots
 | +--rw values* uint8
augment /te:te/te:tunnels/te:tunnel/te:state:
 +--ro payload-treatment? enumeration
 +--ro src-client-signal? identityref
 +--ro src-tpn? uint16
 +--ro src-tsg? identityref
 +--ro src-tributary-slot-count? uint16
 +--ro src-tributary-slots
 | +--ro values* uint8
 +--ro dst-client-signal? identityref
 +--ro dst-tpn? uint16
 +--ro dst-tsg? identityref
 +--ro dst-tributary-slot-count? uint16
 +--ro dst-tributary-slots
 | +--ro values* uint8

```

### 3.5. OTN Tunnel YANG Code

```

<CODE BEGINS>file "ietf-otn-tunnel@2017-03-11.yang"

module ietf-otn-tunnel {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-otn-tunnel";
 prefix "otn-tunnel";

 import ietf-te { prefix "te"; }
 import ietf-transport-types { prefix "tran-types"; }
 //import yang-ext { prefix ext; revision-date 2013-07-09; }

 organization

```

```
"IETF CCAMP Working Group";

contact
 "WG Web: <http://tools.ietf.org/wg/ccamp/>
 WG List: <mailto:ccamp@ietf.org>

 Editor: Anurag Sharma
 <mailto:AnSharma@infinera.com>

 Editor: Rajan Rao
 <mailto:rrao@infinera.com>

 Editor: Xian Zhang
 <mailto:zhang.xian@huawei.com>

 Editor: Kun Xiang
 <mailto:xiangkun@huawei.com>";

description
 "This module defines a model for OTN Tunnel Services.";

revision "2017-03-11" {
 description
 "Revision 0.3";
 reference "TBD";
}

grouping otn-tunnel-endpoint {
 description "Parameters for OTN tunnel.";

 leaf payload-treatment {
 type enumeration {
 enum switching;
 enum transport;
 }
 default switching;
 description
 "Treatment of the incoming payload. Payload can
 either be switched, or transported as is.";
 }

 leaf src-client-signal {
 type identityref {
 base tran-types:client-signal;
 }
 description
 "Client signal at the source endpoint of
 the tunnel.";
 }
 }
}
```

```
 }

 leaf src-tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux
 services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of
 Evolving G.709 Optical Transport Networks.";
 }

 leaf src-tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description
 "Tributary slot granularity. Applicable in case of mux
 services.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }

 leaf src-tributary-slot-count {
 type uint16;
 description
 "Number of tributary slots used at the source.";
 }

 container src-tributary-slots {
 description
 "A list of tributary slots used by the client
 service. Applicable in case of mux services.";
 leaf-list values {
 type uint8;
 description
 "Tributary tributary slot value.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }
 }

 leaf dst-client-signal {
 type identityref {
```

```
 base tran-types:client-signal;
 }
 description
 "Client signal at the destination endpoint of
 the tunnel.";
}

leaf dst-tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux
 services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of
 Evolving G.709 Optical Transport Networks.";
}

leaf dst-tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description
 "Tributary slot granularity. Applicable in case of mux
 services.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
}

leaf dst-tributary-slot-count {
 type uint16;
 description
 "Number of tributary slots used at the destination.";
}

container dst-tributary-slots {
 description
 "A list of tributary slots used by the client
 service. Applicable in case of mux services.";
 leaf-list values {
 type uint8;
 description
 "Tributary slot value.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }
}
```

```

 }
 }
}

/*
Note: Comment has been given to authors of TE Tunnel model to add
tunnel-types to the model in order to identify the technology
type of the service.

grouping otn-service-type {
 description
 "Identifies the OTN Service type.";
 container otn-service {
 presence "Indicates OTN Service.";
 description
 "Its presence identifies the OTN Service type.";
 }
} // otn-service-type

augment "/te:te/te:tunnels/te:tunnel/te:tunnel-types" {
 description
 "Introduce OTN service type for tunnel.";
 ext:augment-identifier otn-service-type-augment;
 uses otn-service-type;
}
*/

/*
Note: Comment has been given to authors of TE Tunnel model to add
list of endpoints under config to support P2MP tunnel.
*/
augment "/te:te/te:tunnels/te:tunnel/te:config" {
 description
 "Augment with additional parameters required for OTN
 service.";
 //ext:augment-identifier otn-tunnel-endpoint-config-augment;
 uses otn-tunnel-endpoint;
}

augment "/te:te/te:tunnels/te:tunnel/te:state" {
 description
 "Augment with additional parameters required for OTN
 service.";
 //ext:augment-identifier otn-tunnel-endpoint-state-augment;
 uses otn-tunnel-endpoint;
}

/*

```

Note: Comment has been given to authors of TE Tunnel model to add tunnel-lifecycle-event to the model. This notification is reported for all lifecycle changes (create, delete, and update) to the tunnel or lsp.

```
augment "/te:tunnel-lifecycle-event" {
 description
 "OTN service event";
 uses otn-service-type;
 uses otn-tunnel-params;

 list endpoint {
 key
 "endpoint-address tp-id";
 description
 "List of Tunnel Endpoints.";
 uses te:tunnel-endpoint;
 uses otn-tunnel-params;
 }
}
*/
}
```

<CODE ENDS>

### 3.6. Transport Types YANG Code

```
<CODE BEGINS> file "ietf-transport-types@2016-10-25.yang"

module ietf-transport-types {
 namespace "urn:ietf:params:xml:ns:yang:ietf-transport-types";
 prefix "tran-types";

 organization
 "IETF CCAMP Working Group";
 contact
 "WG Web: <http://tools.ietf.org/wg/ccamp/>
 WG List: <mailto:ccamp@ietf.org>

 Editor: Anurag Sharma
 <mailto:AnSharma@infinera.com>

 Editor: Rajan Rao
 <mailto:rrao@infinera.com>

 Editor: Xian Zhang
```



```
<mailto:zhang.xian@huawei.com>";

description
 "This module defines transport types.";

revision "2016-10-25" {
 description
 "Revision 0.2";
 reference "TBD";
}

identity tributary-slot-granularity {
 description
 "Tributary slot granularity.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
}

identity tsg-1.25G {
 base tributary-slot-granularity;
 description
 "1.25G tributary slot granularity.";
}

identity tsg-2.5G {
 base tributary-slot-granularity;
 description
 "2.5G tributary slot granularity.";
}

identity tributary-protocol-type {
 description
 "Base identity for protocol framing used by
 tributary signals.";
}

identity prot-OTU1 {
 base tributary-protocol-type;
 description
 "OTU1 protocol (2.66G)";
}

/*
identity prot-OTU1e {
 base tributary-protocol-type;
 description
 "OTU1e type (11.04G)";
}
```

```
 }

 identity prot-OTU1f {
 base tributary-protocol-type;
 description
 "OTU1f type (11.27G)";
 }

 /*
 identity prot-OTU2 {
 base tributary-protocol-type;
 description
 "OTU2 type (10.70G)";
 }

 identity prot-OTU2e {
 base tributary-protocol-type;
 description
 "OTU2e type (11.09G)";
 }

 /*
 identity prot-OTU2f {
 base tributary-protocol-type;
 description
 "OTU2f type (11.31G)";
 }
 */

 identity prot-OTU3 {
 base tributary-protocol-type;
 description
 "OTU3 type (43.01G)";
 }

 /*
 identity prot-OTU3e1 {
 base tributary-protocol-type;
 description
 "OTU3e1 type (44.57G)";
 }

 identity prot-OTU3e2 {
 base tributary-protocol-type;
 description
 "OTU3e2 type (44.58G)";
 }
 */
```

```
identity prot-OTU4 {
 base tributary-protocol-type;
 description
 "OTU4 type (111.80G)";
}

identity prot-OTUCn {
 base tributary-protocol-type;
 description
 "OTUCn type (beyond 100G)";
}

identity prot-ODU0 {
 base tributary-protocol-type;
 description
 "ODU0 protocol (1.24G).";
}

identity prot-ODU1 {
 base tributary-protocol-type;
 description
 "ODU1 protocol (2.49G).";
}

/*
identity prot-ODU1e {
 base tributary-protocol-type;
 description
 "ODU1e protocol (10.35G).";
}

identity prot-ODU1f {
 base tributary-protocol-type;
 description
 "ODU1f protocol (10.56G).";
}
*/

identity prot-ODU2 {
 base tributary-protocol-type;
 description
 "ODU2 protocol (10.03G).";
}

identity prot-ODU2e {
 base tributary-protocol-type;
 description
 "ODU2e protocol (10.39G).";
}
```

```
 }

 /*
 identity prot-ODU2f {
 base tributary-protocol-type;
 description
 "ODU2f protocol (10.60G).";
 }
 */

 identity prot-ODU3 {
 base tributary-protocol-type;
 description
 "ODU3 protocol (40.31G).";
 }

 /*
 identity prot-ODU3e1 {
 base tributary-protocol-type;
 description
 "ODU3e1 protocol (41.77G).";
 }

 identity prot-ODU3e2 {
 base tributary-protocol-type;
 description
 "ODU3e2 protocol (41.78G).";
 }
 */

 identity prot-ODU4 {
 base tributary-protocol-type;
 description
 "ODU4 protocol (104.79G).";
 }

 identity prot-ODUFlex-cbr {
 base tributary-protocol-type;
 description
 "ODU Flex CBR protocol for transporting constant bit
 rate signal.";
 }

 identity prot-ODUFlex-gfp {
 base tributary-protocol-type;
 description
 "ODU Flex GFP protocol for transporting stream of packets
 using Generic Framing Procedure.";
```

```
 }

 identity prot-ODUCn {
 base tributary-protocol-type;
 description
 "ODUCn protocol (beyond 100G).";
 }

 identity prot-1GbE {
 base tributary-protocol-type;
 description
 "1G Ethernet protocol";
 }

 identity prot-10GbE-LAN {
 base tributary-protocol-type;
 description
 "10G Ethernet LAN protocol";
 }

 identity prot-40GbE {
 base tributary-protocol-type;
 description
 "40G Ethernet protocol";
 }

 identity prot-100GbE {
 base tributary-protocol-type;
 description
 "100G Ethernet protocol";
 }

 identity client-signal {
 description
 "Base identity from which specific client signals for the
 tunnel are derived.";
 }

 identity client-signal-1GbE {
 base client-signal;
 description
 "Client signal type of 1GbE";
 }

 identity client-signal-10GbE-LAN {
 base client-signal;
 description
 "Client signal type of 10GbE LAN";
 }
```

```
 }

 identity client-signal-10GbE-WAN {
 base client-signal;
 description
 "Client signal type of 10GbE WAN";
 }

 identity client-signal-40GbE {
 base client-signal;
 description
 "Client signal type of 40GbE";
 }

 identity client-signal-100GbE {
 base client-signal;
 description
 "Client signal type of 100GbE";
 }

 identity client-signal-OC3_STM1 {
 base client-signal;
 description
 "Client signal type of OC3 & STM1";
 }

 identity client-signal-OC12_STM4 {
 base client-signal;
 description
 "Client signal type of OC12 & STM4";
 }

 identity client-signal-OC48_STM16 {
 base client-signal;
 description
 "Client signal type of OC48 & STM16";
 }

 identity client-signal-OC192_STM64 {
 base client-signal;
 description
 "Client signal type of OC192 & STM64";
 }

 identity client-signal-OC768_STM256 {
 base client-signal;
 description
 "Client signal type of OC768 & STM256";
 }
```

```
 }

 identity client-signal-ODU0 {
 base client-signal;
 description
 "Client signal type of ODU0 (1.24G)";
 }

 identity client-signal-ODU1 {
 base client-signal;
 description
 "ODU1 protocol (2.49G)";
 }

 identity client-signal-ODU2 {
 base client-signal;
 description
 "Client signal type of ODU2 (10.03G)";
 }

 identity client-signal-ODU2e {
 base client-signal;
 description
 "Client signal type of ODU2e (10.39G)";
 }

 identity client-signal-ODU3 {
 base client-signal;
 description
 "Client signal type of ODU3 (40.31G)";
 }

 /*
 identity client-signal-ODU3e2 {
 base client-signal;
 description
 "Client signal type of ODU3e2 (41.78G)";
 }
 */

 identity client-signal-ODU4 {
 base client-signal;
 description
 "Client signal type of ODU4 (104.79G)";
 }

 identity client-signal-ODUFlex-cbr {
 base client-signal;
```

```
 description
 "Client signal type of ODU Flex CBR";
 }

 identity client-signal-ODUFlex-gfp {
 base client-signal;
 description
 "Client signal type of ODU Flex GFP";
 }

 identity client-signal-ODUCn {
 base client-signal;
 description
 "Client signal type of ODUCn (beyond 100G).";
 }

 identity client-signal-FC400 {
 base client-signal;
 description
 "Client signal type of Fibre Channel FC400.";
 }

 identity client-signal-FC800 {
 base client-signal;
 description
 "Client signal type of Fibre Channel FC800.";
 }

 identity client-signal-FICON-4G {
 base client-signal;
 description
 "Client signal type of Fibre Connection 4G.";
 }

 identity client-signal-FICON-8G {
 base client-signal;
 description
 "Client signal type of Fibre Connection 8G.";
 }
}
<CODE ENDS>
```



## 4. Security Considerations

TBD.

## 5. IANA Considerations

TBD.

## 6. Acknowledgements

TBD.

## 7. Normative References

- [G.709] "Interfaces for the Optical Transport Network(OTN)", G.709/Y.1331 Recommendation , June 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC7139] Zhang, F., Ed., Zhang, G., Belotti, S., Ceccarelli, D., and K. Pithewan, "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, DOI 10.17487/RFC7139, March 2014, <<http://www.rfc-editor.org/info/rfc7139>>.

## Authors' Addresses

Xian Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: [zhang.xian@huawei.com](mailto:zhang.xian@huawei.com)

Kun Xiang  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: xiangkun@huawei.com

Anurag Sharma  
Infinera

Email: ansharma@infinera.com

Rajan Rao  
Infinera  
169 Java Drive  
Sunnyvale, CA 94089  
USA

Email: rrao@infinera.com

CCAMP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: November 26, 2017

H. Zheng  
Z. Fan  
Huawei Technologies  
A. Sharma  
Google  
R. Rao  
Infinera  
S. Belotti  
Nokia  
V. Lopez  
Telefonica  
Y. Li  
China Mobile  
May 25, 2017

OTN Tunnel YANG Model  
draft-sharma-ccamp-otn-tunnel-model-02

Abstract

This document describes the YANG data model for OTN Tunnels.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 26, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                        |    |
|--------------------------------------------------------|----|
| 1. Introduction . . . . .                              | 2  |
| 2. Terminology and Notations . . . . .                 | 3  |
| 3. Model Overview . . . . .                            | 3  |
| 3.1. Mux Service in Multi-Domain OTN Network . . . . . | 3  |
| 3.2. Bookended and Non-BookEnded OTN Tunnel . . . . .  | 4  |
| 3.3. Network and Client side tunnel services . . . . . | 4  |
| 3.4. OTN Tunnel YANG Tree . . . . .                    | 5  |
| 3.5. OTN Tunnel YANG Code . . . . .                    | 5  |
| 3.6. Transport Types YANG Code . . . . .               | 10 |
| 4. Security Considerations . . . . .                   | 19 |
| 5. IANA Considerations . . . . .                       | 19 |
| 6. Acknowledgements . . . . .                          | 19 |
| 7. Contributors . . . . .                              | 19 |
| 8. Normative References . . . . .                      | 20 |
| Authors' Addresses . . . . .                           | 20 |

## 1. Introduction

OTN transport networks can carry various types of client services. In many cases, the client signal is carried over an OTN tunnel across connected domains in a multi-domain network. These OTN services can either be transported or switched in the OTN network. If an OTN tunnel is switched, then additional parameters need to be provided to create a Mux OTN service.

This document provides YANG model for creating OTN tunnel. The model augments the TE Tunnel model, which is an abstract model to create TE Tunnels.

## 2. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [I-D.ietf-netmod-rfc6087bis]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

## 3. Model Overview

### 3.1. Mux Service in Multi-Domain OTN Network

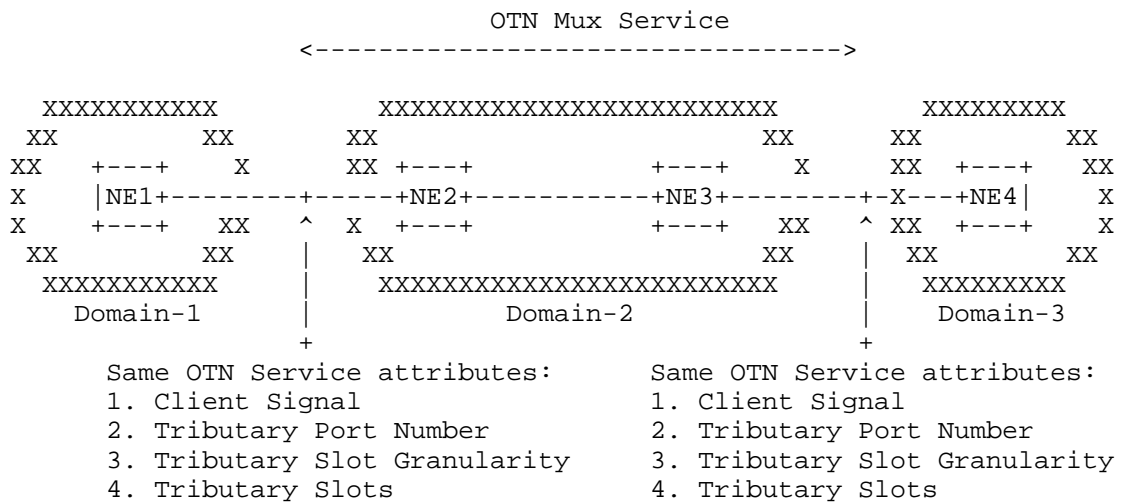


Figure 1: OTN Mux Service in a multi-domain network topology

Figure 1 shows a multi-domain OTN network with three domains. In this example, user wants to setup an end-to-end OTN service that passes through Domain-2. In order to create an OTN mux service in Domain-2, user will need to specify the exact details of the client side LO-ODU on NE2 and NE3, so that these service endpoints can be paired with the LO-ODU endpoints on NE1 and NE4, respectively.

Let's assume that ODU4 is the client side HO-ODU on NE2 and NE3, and the client signal is ODU2. User will need to specify the OTN client signal (ODU2 in this example), the Tributary Port Number (TPN), Tributary Slot Granularities (TSG) and tributary slots to be used. As shown in the figure above, these service parameters must be the same between NE1 and NE2, and NE3 and NE4.

Once the OTN Mux service is setup in Domain-2, the incoming signal from either NE1 and/or NE4 will be switched inside Domain-2, and delivered to NE at the other end.

### 3.2. Bookended and Non-BookEnded OTN Tunnel

OTN tunnel model provides support for both bookended and non-bookended OTN tunnels.

For bookended tunnels, the same client signal is present on source and destination endpoints. For example, ODU2e bookended tunnel will have the same ODU2e client signal at both source and destination endpoints.

For non-bookended tunnels, different client signals are present on source and destination endpoints. For example, the client signal can be ODU2e on the source endpoint and the handoff at the destination can be 10GbE-LAN client signal.

### 3.3. Network and Client side tunnel services

The OTN tunnel model provides support for both network to network and client to client tunnels. For network to network tunnel, network termination points on source and destination node represent source and destination endpoints. For client to client tunnel, client termination points on source and destination node represent source and destination endpoints.

If a client to client tunnel needs to use one or more HO (or server) network to network tunnels, ERO and routing constraints, defined in the base TE model, can be used to route the client tunnel over one or more server tunnels.

### 3.4. OTN Tunnel YANG Tree

```

module: ietf-otn-tunnel
augment /te:te/te:tunnels/te:tunnel/te:config:
 +--rw payload-treatment? enumeration
 +--rw src-client-signal? identityref
 +--rw src-tpn? uint16
 +--rw src-tsg? identityref
 +--rw src-tributary-slot-count? uint16
 +--rw src-tributary-slots
 | +--rw values* uint8
 +--rw dst-client-signal? identityref
 +--rw dst-tpn? uint16
 +--rw dst-tsg? identityref
 +--rw dst-tributary-slot-count? uint16
 +--rw dst-tributary-slots
 +--rw values* uint8
augment /te:te/te:tunnels/te:tunnel/te:state:
 +--ro payload-treatment? enumeration
 +--ro src-client-signal? identityref
 +--ro src-tpn? uint16
 +--ro src-tsg? identityref
 +--ro src-tributary-slot-count? uint16
 +--ro src-tributary-slots
 | +--ro values* uint8
 +--ro dst-client-signal? identityref
 +--ro dst-tpn? uint16
 +--ro dst-tsg? identityref
 +--ro dst-tributary-slot-count? uint16
 +--ro dst-tributary-slots
 +--ro values* uint8

```

### 3.5. OTN Tunnel YANG Code

```

<CODE BEGINS>file "ietf-otn-tunnel@2017-05-25.yang"

module ietf-otn-tunnel {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-otn-tunnel";
 prefix "otn-tunnel";

 import ietf-te { prefix "te"; }
 import ietf-transport-types { prefix "tran-types"; }
 //import yang-ext { prefix ext; revision-date 2013-07-09; }

```

organization

"IETF CCAMP Working Group";

contact

"WG Web: <<http://tools.ietf.org/wg/ccamp/>>

WG List: <<mailto:ccamp@ietf.org>>

Editor: Haomian Zheng

<<mailto:zhenghaomian@huawei.com>>

Editor: Zheyu Fan

<<mailto:fanzheyu2@huawei.com>>

Editor: Anurag Sharma

<<mailto:ansha@google.com>>

Editor: Rajan Rao

<<mailto:rrao@infinera.com>>

Editor: Sergio Belotti

<<mailto:sergio.belotti@nokia.com>>

Editor: Victor Lopez

<<mailto:victor.lopezalvarez@telefonica.com>>

Editor: Yunbo Li

<<mailto:liyunbo@chinamobile.com>>"

description

"This module defines a model for OTN Tunnel Services.";

revision "2017-05-25" {

description

"Revision 0.3";

reference

"draft-sharma-ccamp-otn-tunnel-model-02.txt";

}

grouping otn-tunnel-endpoint {

description "Parameters for OTN tunnel.";

leaf payload-treatment {

type enumeration {

enum switching;

enum transport;

}

default switching;

description



```
 "Treatment of the incoming payload. Payload can
 either be switched, or transported as is.";
 }

 leaf src-client-signal {
 type identityref {
 base tran-types:client-signal;
 }
 description
 "Client signal at the source endpoint of
 the tunnel.";
 }

 leaf src-tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux
 services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of
 Evolving G.709 Optical Transport Networks.";
 }

 leaf src-tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description
 "Tributary slot granularity. Applicable in case of
 mux services.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }

 leaf src-tributary-slot-count {
 type uint16;
 description
 "Number of tributary slots used at the source.";
 }

 container src-tributary-slots {
 description
 "A list of tributary slots used by the client
 service. Applicable in case of mux services.";
 leaf-list values {
```

```
 type uint8;
 description
 "Tributary tributary slot value.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }
}

leaf dst-client-signal {
 type identityref {
 base tran-types:client-signal;
 }
 description
 "Client signal at the destination endpoint of
 the tunnel.";
}

leaf dst-tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux
 services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of
 Evolving G.709 Optical Transport Networks.";
}

leaf dst-tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description
 "Tributary slot granularity. Applicable in case of
 mux services.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
}

leaf dst-tributary-slot-count {
 type uint16;
 description
 "Number of tributary slots used at the destination.";
}
```

```
 container dst-tributary-slots {
 description
 "A list of tributary slots used by the client
 service. Applicable in case of mux services.";
 leaf-list values {
 type uint8;
 description
 "Tributary slot value.";
 reference
 "G.709/Y.1331, February 2016: Interfaces for the
 Optical Transport Network (OTN)";
 }
 }
 }

/*
Note: Comment has been given to authors of TE Tunnel model to add
tunnel-types to the model in order to identify the technology
type of the service.

grouping otn-service-type {
 description
 "Identifies the OTN Service type.";
 container otn-service {
 presence "Indicates OTN Service.";
 description
 "Its presence identifies the OTN Service type.";
 }
} // otn-service-type

augment "/te:te/te:tunnels/te:tunnel/te:tunnel-types" {
 description
 "Introduce OTN service type for tunnel.";
 ext:augment-identifier otn-service-type-augment;
 uses otn-service-type;
}
*/

/*
Note: Comment has been given to authors of TE Tunnel model to add
list of endpoints under config to support P2MP tunnel.
*/
augment "/te:te/te:tunnels/te:tunnel/te:config" {
 description
 "Augment with additional parameters required for OTN
 service.";
 //ext:augment-identifier otn-tunnel-endpoint-config-augment;
 uses otn-tunnel-endpoint;
```

```

 }

 augment "/te:te/te:tunnels/te:tunnel/te:state" {
 description
 "Augment with additional parameters required for OTN
 service.";
 //ext:augment-identifier otn-tunnel-endpoint-state-augment;
 uses otn-tunnel-endpoint;
 }

 /*
 Note: Comment has been given to authors of TE Tunnel model to
 add tunnel-lifecycle-event to the model. This notification is
 reported for all lifecycle changes (create, delete, and update)
 to the tunnel or lsp.
 augment "/te:tunnel-lifecycle-event" {
 description
 "OTN service event";
 uses otn-service-type;
 uses otn-tunnel-params;

 list endpoint {
 key
 "endpoint-address tp-id";
 description
 "List of Tunnel Endpoints.";
 uses te:tunnel-endpoint;
 uses otn-tunnel-params;
 }
 }
 */
}
<CODE ENDS>

```

### 3.6. Transport Types YANG Code

```

<CODE BEGINS> file "ietf-transport-types@2017-05-25.yang"

module ietf-transport-types {
 namespace "urn:ietf:params:xml:ns:yang:ietf-transport-types";
 prefix "tran-types";

 organization
 "IETF CCAMP Working Group";
 contact
 "WG Web: <http://tools.ietf.org/wg/ccamp/>"

```

WG List: <mailto:ccamp@ietf.org>

Editor: Haomian Zheng  
<mailto:zhenghaomian@huawei.com>

Editor: Zheyu Fan  
<mailto:fanzheyu2@huawei.com>

Editor: Anurag Sharma  
<mailto:ansha@google.com>

Editor: Rajan Rao  
<mailto:rrao@infinera.com>

Editor: Sergio Belotti  
<mailto:sergio.belotti@nokia.com>

Editor: Victor Lopez  
<mailto:victor.lopezalvarez@telefonica.com>

Editor: Yunbo Li  
<mailto:liyunbo@chinamobile.com>";

description  
"This module defines transport types.";

revision "2017-05-25" {  
  description  
    "Revision 0.3";  
  reference  
    "draft-sharma-ccamp-otn-tunnel-model-02.txt";  
}

identity tributary-slot-granularity {  
  description  
    "Tributary slot granularity.";  
  reference  
    "G.709/Y.1331, February 2016: Interfaces for the  
      Optical Transport Network (OTN)";  
}

identity tsg-1.25G {  
  base tributary-slot-granularity;  
  description  
    "1.25G tributary slot granularity.";  
}

identity tsg-2.5G {

```
 base tributary-slot-granularity;
 description
 "2.5G tributary slot granularity.";
 }

 identity tributary-protocol-type {
 description
 "Base identity for protocol framing used by
 tributary signals.";
 }

 identity prot-OTU1 {
 base tributary-protocol-type;
 description
 "OTU1 protocol (2.66G)";
 }

 /*
 identity prot-OTU1e {
 base tributary-protocol-type;
 description
 "OTU1e type (11.04G)";
 }

 identity prot-OTU1f {
 base tributary-protocol-type;
 description
 "OTU1f type (11.27G)";
 }
 */

 identity prot-OTU2 {
 base tributary-protocol-type;
 description
 "OTU2 type (10.70G)";
 }

 identity prot-OTU2e {
 base tributary-protocol-type;
 description
 "OTU2e type (11.09G)";
 }

 /*
 identity prot-OTU2f {
 base tributary-protocol-type;
 description
 "OTU2f type (11.31G)";
 }
 */
```

```
*/

identity prot-OTU3 {
 base tributary-protocol-type;
 description
 "OTU3 type (43.01G)";
}

/*
identity prot-OTU3e1 {
 base tributary-protocol-type;
 description
 "OTU3e1 type (44.57G)";
}

identity prot-OTU3e2 {
 base tributary-protocol-type;
 description
 "OTU3e2 type (44.58G)";
}
*/

identity prot-OTU4 {
 base tributary-protocol-type;
 description
 "OTU4 type (111.80G)";
}

identity prot-OTUCn {
 base tributary-protocol-type;
 description
 "OTUCn type (beyond 100G)";
}

identity prot-ODU0 {
 base tributary-protocol-type;
 description
 "ODU0 protocol (1.24G).";
}

identity prot-ODU1 {
 base tributary-protocol-type;
 description
 "ODU1 protocol (2.49G).";
}

/*
identity prot-ODU1e {
```

```
 base tributary-protocol-type;
 description
 "ODU1e protocol (10.35G).";
 }

 identity prot-ODU1f {
 base tributary-protocol-type;
 description
 "ODU1f protocol (10.56G).";
 }
 */

 identity prot-ODU2 {
 base tributary-protocol-type;
 description
 "ODU2 protocol (10.03G).";
 }

 identity prot-ODU2e {
 base tributary-protocol-type;
 description
 "ODU2e protocol (10.39G).";
 }

 /*
 identity prot-ODU2f {
 base tributary-protocol-type;
 description
 "ODU2f protocol (10.60G).";
 }
 */

 identity prot-ODU3 {
 base tributary-protocol-type;
 description
 "ODU3 protocol (40.31G).";
 }

 /*
 identity prot-ODU3e1 {
 base tributary-protocol-type;
 description
 "ODU3e1 protocol (41.77G).";
 }

 identity prot-ODU3e2 {
 base tributary-protocol-type;
 description
```



```
 "ODU3e2 protocol (41.78G).";
 }
 */

 identity prot-ODU4 {
 base tributary-protocol-type;
 description
 "ODU4 protocol (104.79G).";
 }

 identity prot-ODUFlex-cbr {
 base tributary-protocol-type;
 description
 "ODU Flex CBR protocol for transporting constant bit
 rate signal.";
 }

 identity prot-ODUFlex-gfp {
 base tributary-protocol-type;
 description
 "ODU Flex GFP protocol for transporting stream of packets
 using Generic Framing Procedure.";
 }

 identity prot-ODUCn {
 base tributary-protocol-type;
 description
 "ODUCn protocol (beyond 100G).";
 }

 identity prot-1GbE {
 base tributary-protocol-type;
 description
 "1G Ethernet protocol";
 }

 identity prot-10GbE-LAN {
 base tributary-protocol-type;
 description
 "10G Ethernet LAN protocol";
 }

 identity prot-40GbE {
 base tributary-protocol-type;
 description
 "40G Ethernet protocol";
 }
}
```

```
identity prot-100GbE {
 base tributary-protocol-type;
 description
 "100G Ethernet protocol";
}

identity client-signal {
 description
 "Base identity from which specific client signals for the
 tunnel are derived.";
}

identity client-signal-1GbE {
 base client-signal;
 description
 "Client signal type of 1GbE";
}

identity client-signal-10GbE-LAN {
 base client-signal;
 description
 "Client signal type of 10GbE LAN";
}

identity client-signal-10GbE-WAN {
 base client-signal;
 description
 "Client signal type of 10GbE WAN";
}

identity client-signal-40GbE {
 base client-signal;
 description
 "Client signal type of 40GbE";
}

identity client-signal-100GbE {
 base client-signal;
 description
 "Client signal type of 100GbE";
}

identity client-signal-OC3_STM1 {
 base client-signal;
 description
 "Client signal type of OC3 & STM1";
}
```

```
identity client-signal-OC12_STM4 {
 base client-signal;
 description
 "Client signal type of OC12 & STM4";
}

identity client-signal-OC48_STM16 {
 base client-signal;
 description
 "Client signal type of OC48 & STM16";
}

identity client-signal-OC192_STM64 {
 base client-signal;
 description
 "Client signal type of OC192 & STM64";
}

identity client-signal-OC768_STM256 {
 base client-signal;
 description
 "Client signal type of OC768 & STM256";
}

identity client-signal-ODU0 {
 base client-signal;
 description
 "Client signal type of ODU0 (1.24G)";
}

identity client-signal-ODU1 {
 base client-signal;
 description
 "ODU1 protocol (2.49G)";
}

identity client-signal-ODU2 {
 base client-signal;
 description
 "Client signal type of ODU2 (10.03G)";
}

identity client-signal-ODU2e {
 base client-signal;
 description
 "Client signal type of ODU2e (10.39G)";
}
```

```
identity client-signal-ODU3 {
 base client-signal;
 description
 "Client signal type of ODU3 (40.31G)";
}

/*
identity client-signal-ODU3e2 {
 base client-signal;
 description
 "Client signal type of ODU3e2 (41.78G)";
}
*/

identity client-signal-ODU4 {
 base client-signal;
 description
 "Client signal type of ODU4 (104.79G)";
}

identity client-signal-ODUFlex-cbr {
 base client-signal;
 description
 "Client signal type of ODU Flex CBR";
}

identity client-signal-ODUFlex-gfp {
 base client-signal;
 description
 "Client signal type of ODU Flex GFP";
}

identity client-signal-ODUCn {
 base client-signal;
 description
 "Client signal type of ODUCn (beyond 100G).";
}

identity client-signal-FC400 {
 base client-signal;
 description
 "Client signal type of Fibre Channel FC400.";
}

identity client-signal-FC800 {
 base client-signal;
 description
 "Client signal type of Fibre Channel FC800.";
```

```
 }

 identity client-signal-FICON-4G {
 base client-signal;
 description
 "Client signal type of Fibre Connection 4G.";
 }

 identity client-signal-FICON-8G {
 base client-signal;
 description
 "Client signal type of Fibre Connection 8G.";
 }
}
<CODE ENDS>
```

#### 4. Security Considerations

TBD.

#### 5. IANA Considerations

TBD.

#### 6. Acknowledgements

TBD.

#### 7. Contributors

Dieter Beller  
Nokia  
Email: dieter.beller@nokia.com

Yanlei Zheng  
China Unicom  
Email: zhengyl@dimpt.com

Xian Zhang  
Huawei Technologies  
Email: zhang.xian@huawei.com

Lei Wang  
China Mobile  
Email: wangleiyj@chinamobile.com

Oscar Gonzalez de Dios

Telefonica  
Email: oscar.gonzalezdedios@telefonica.com

## 8. Normative References

- [I-D.ietf-netmod-rfc6087bis]  
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-12 (work in progress), March 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

## Authors' Addresses

Haomian Zheng  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: zhenghaomian@huawei.com

Zheyu Fan  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: fanzheyu2@huawei.com

Anurag Sharma  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
USA

Email: ansha@google.com

Rajan Rao  
Infinera  
169 Java Drive  
Sunnyvale, CA 94089  
USA

Email: rrao@infinera.com

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

Victor Lopez  
Telefonica

Email: victor.lopezalvarez@telefonica.com

Yunbo Li  
China Mobile

Email: liyunbo@chinamobile.com

Internet Engineering Task Force  
Internet-Draft  
Intended status: Standards Track  
Expires: May 18, 2018

Yimin Shen  
Minto Jeyananth  
Juniper Networks  
Bruno Decraene  
Orange  
Hannes Gredler  
RtBrick Inc  
Carsten Michel  
Deutsche Telekom  
Huaimo Chen  
Yuanlong Jiang  
Huawei Technologies Co., Ltd.  
November 14, 2017

MPLS Egress Protection Framework  
draft-shen-mpls-egress-protection-framework-07

Abstract

This document specifies a fast reroute framework for protecting IP/MPLS services and MPLS transport tunnels against egress node and egress link failures. In this framework, the penultimate-hop router of an MPLS tunnel acts as the point of local repair (PLR) for egress node failure, and the egress router of the MPLS tunnel acts as the PLR for egress link failure. Each of them pre-establishes a bypass tunnel to a protector. Upon an egress node or link failure, the corresponding PLR performs local failure detection and local repair, by rerouting packets over the corresponding bypass tunnel. The protector in turn performs context label switching or context IP forwarding to send the packets to the ultimate service destination(s). This mechanism can be used to reduce traffic loss before global repair reacts to the failure and control plane protocols converge on the topology changes due to the failure. The framework is applicable to all types of IP/MPLS services and MPLS tunnels. Under the framework, service protocol extensions may be further specified to support service label distribution to the protector.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.



Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 18, 2018.

#### Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                                  | 3  |
| 2. Specification of Requirements . . . . .                                 | 5  |
| 3. Terminology . . . . .                                                   | 5  |
| 4. Requirements . . . . .                                                  | 7  |
| 5. Egress node protection . . . . .                                        | 8  |
| 5.1. Reference topology . . . . .                                          | 8  |
| 5.2. Egress node failure and detection . . . . .                           | 8  |
| 5.3. Protector and PLR . . . . .                                           | 9  |
| 5.4. Protected egress . . . . .                                            | 10 |
| 5.5. Egress-protected tunnel and service . . . . .                         | 11 |
| 5.6. Egress-protection bypass tunnel . . . . .                             | 11 |
| 5.7. Context ID, context label, and context based forwarding . . . . .     | 12 |
| 5.8. Advertisement and path resolution for context ID . . . . .            | 14 |
| 5.9. Egress-protection bypass tunnel establishment . . . . .               | 15 |
| 5.10. Local repair on PLR . . . . .                                        | 15 |
| 5.11. Service label distribution from egress router to protector . . . . . | 16 |
| 5.12. Centralized protector mode . . . . .                                 | 16 |
| 6. Egress link protection . . . . .                                        | 18 |
| 7. Global repair . . . . .                                                 | 21 |
| 8. Example: Layer-3 VPN egress protection . . . . .                        | 21 |
| 8.1. Egress node protection . . . . .                                      | 23 |
| 8.2. Egress link protection . . . . .                                      | 24 |
| 8.3. Global repair . . . . .                                               | 24 |

|                                        |    |
|----------------------------------------|----|
| 9. IANA Considerations . . . . .       | 24 |
| 10. Security Considerations . . . . .  | 24 |
| 11. Acknowledgements . . . . .         | 25 |
| 12. References . . . . .               | 25 |
| 12.1. Normative References . . . . .   | 25 |
| 12.2. Informative References . . . . . | 25 |
| Authors' Addresses . . . . .           | 26 |

## 1. Introduction

In MPLS networks, label switched paths (LSPs) are widely used as transport tunnels to carry IP and MPLS services across MPLS domains. Examples of MPLS services are layer-2 VPNs, layer-3 VPNs, hierarchical LSPs, and others. In general, a tunnel may carry multiple services of one or multiple types, if the tunnel can satisfy both individual and aggregate requirements (e.g. CoS, QoS) of these services. The egress router of the tunnel should host the corresponding service instances of the services. An MPLS service instance is responsible for forwarding service packets via an egress link to the service destination, based on a service label. An IP service instance is responsible for doing the same based on a service IP address. The egress link is often called a PE-CE (provider edge - customer edge) link or attachment circuit (AC).

Today, local repair based fast reroute mechanisms [RFC4090], [RFC5286], [RFC7490], [RFC7812] have been widely deployed to protect MPLS tunnels against transit link/node failures. They can achieve fast restoration of traffic in the order of tens of milliseconds. Local repair refers to the scenario where the router upstream to an anticipated failure (aka. PLR, i.e. point of local repair) pre-establishes a bypass tunnel to the router downstream of the failure (aka. MP, i.e. merge point), and pre-installs the forwarding state of the bypass tunnel in the data plane. The PLR also uses a rapid mechanism (e.g. link layer OAM, BFD, and others) to locally detect the failure in the data plane. When the failure occurs, the PLR reroutes traffic through the bypass tunnel to the MP, allowing the traffic to continue to flow to the tunnel's egress router.

This document describes a fast reroute framework for egress node and egress link protection. Similar to transit link/node protection, this framework relies on a PLR to perform local failure detection and local repair. In egress node protection, the PLR is the penultimate-hop router of a tunnel. In egress link protection, the PLR is the egress router of the tunnel. The framework relies on a so-called "protector" to serve as the tailend of a bypass tunnel. The protector is a router that hosts "protection service instances" and has its own connectivity or paths to service destinations. When a PLR is doing local repair, the protector is responsible for

performing "context label switching" for rerouted MPLS service packets and "context IP forwarding" for rerouted IP service packets. Thus, the service packets can continue to reach service destinations with minimum disruption.

This framework considers an egress node failure as a failure of a tunnel, as well as a failure of all the services carried by the tunnel, because service packets can no longer reach the service instances on the egress router. Therefore, the framework addresses egress node protection at both tunnel level and service level simultaneously. Likewise, the framework considers an egress link failure as a failure of all the services traversing the link, and addresses egress link protection at the service level.

This framework requires that the destination (a CE or site) of a service MUST be dual-homed or have dual paths to an MPLS network, normally via two MPLS edge routers. One of them is the egress router of the service's transport tunnel, and the other is a backup egress router which hosts "backup service instances". In the "co-located" protector mode in this document, the backup egress router serves as a protector, and hence each backup service instance acts as a protection instance. In the "centralized" protector mode (Section 5.12), a protector and a backup egress router are decoupled, and each protection service instance and its corresponding backup service instance are hosted on separate routers.

The framework is described by mainly referring to P2P (point-to-point) tunnels. However, it is equally applicable to P2MP (point-to-multipoint), MP2P (multipoint-to-point) and MP2MP (multipoint-to-multipoint) tunnels, when a sub-LSP can be viewed as a P2P tunnel.

The framework is a multi-service and multi-transport framework. It assumes a generic model where each service is comprised of a common set of components, including a service instance, a service label, and a service label distribution protocol, and the service is transported over an MPLS tunnel of any type. The framework also assumes service labels to be downstream assigned, i.e. assigned by egress routers. Therefore, the framework is generally applicable to most existing and future services. Services which use upstream-assigned service labels are out of scope of this document and left for further study.

The framework does not require extensions for the existing signaling and label distribution protocols (e.g. RSVP, LDP, BGP, etc.) of MPLS tunnels. It expects transport tunnels and bypass tunnels to be established by using the generic mechanisms provided by the protocols. On the other hand, it does not preclude future extensions to the protocols which may facilitate the procedures. One example of such extension is [RSVP-EP]. The framework may need extensions for

IGPs and service label distribution protocols, to support protection establishment and context label switching. This document provides guidelines for these extensions, but the specific details SHOULD be addressed in separate documents.

The framework is intended to complement control-plane convergence and global repair, which are traditionally used to recover networks from egress node and egress link failures. Control-plane convergence relies on control protocols to react on the topology changes due to a failure. Global repair relies on an ingress router to remotely detect a failure and switch traffic to an alternative path. An example of global repair is the BGP Prefix Independent Convergence mechanism [BGP-PIC] for BGP established services. Compared with these mechanisms, this framework is considered as faster in traffic restoration, due to the nature of local failure detection and local repair. However, it is RECOMMENDED that the framework SHOULD be used in conjunction with control-plane convergence or global repair, in order to take the advantages of both approaches to achieve more effective protection. That is, the framework provides fast and temporary repair, and control-plane convergence or global repair provides ultimate and permanent repair.

## 2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

## 3. Terminology

Egress router - A router at the egress endpoint of a tunnel. It hosts service instances for all the services carried by the tunnel, and has connectivity with the destinations of the services.

Egress node failure - A failure of an egress router.

Egress link failure - A failure of the egress link (e.g. PE-CE link, attachment circuit) of a service.

Egress failure - An egress node failure or an egress link failure.

Egress-protected tunnel - A tunnel whose egress router is protected by a mechanism according to this framework. The egress router is hence called a protected egress router.

Egress-protected service - An IP or MPLS service which is carried by an egress-protected tunnel, and hence protected by a mechanism according to this framework.

Backup egress router - Given an egress-protected tunnel and its egress router, this is another router which has connectivity with all or a subset of the destinations of the egress-protected services carried by the egress-protected tunnel.

Backup service instance - A service instance which is hosted by a backup egress router, and corresponding to an egress-protected service on a protected egress router.

Protector - A role acted by a router as an alternate of a protected egress router, to handle service packets in the event of an egress failure. A protector may be physically co-located with or decoupled from a backup egress router, depending on the co-located or centralized protector mode.

Protection service instance - A service instance hosted by a protector, corresponding to the service instance of an egress-protected service on a protected egress router. A protection service instance is a backup service instance, if the protector is co-located with a backup egress router.

PLR - A router at the point of local repair. In egress node protection, it is the penultimate-hop router on an egress-protected tunnel. In egress link protection, it is the egress router of the egress-protected tunnel.

Protected egress {E, P} - A virtual node consisting of an ordered pair of egress router E and protector P. It serves as the virtual destination of an egress-protected tunnel, and as the virtual location of the egress-protected services carried by the tunnel.

Context identifier (ID) - A globally unique IP address assigned to a protected egress {E, P}.

Context label - A non-reserved label assigned to a context ID by a protector.

Egress-protection bypass tunnel - A tunnel used to reroute service packets around an egress failure.

Co-located protector mode - The scenario where a protector and a backup egress router are co-located as one router, and hence each backup service instance serves as a protection service instance.

Centralized protector mode - The scenario where a protector is a dedicated router, and is decoupled from backup egress routers.

Context label switching - Label switching performed by a protector, in the label space of an egress router indicated by a context label.

Context IP forwarding - IP forwarding performed by a protector, in the IP address space of an egress router indicated by a context label.

#### 4. Requirements

This document considers the followings as the design requirements of this egress protection framework.

- o The framework must support P2P tunnels. It should equally support P2MP, MP2P and MP2MP tunnels, by treating each sub-LSP as a P2P tunnel.
- o The framework must support multi-service and multi-transport networks. It must accommodate existing and future signaling and label-distribution protocols of tunnels and bypass tunnels, including RSVP, LDP, BGP, IGP, segment routing, and others. It must also accommodate existing and future IP/MPLS services, including layer-2 VPNs, layer-3 VPNs, hierarchical LSP, and others. It must provide a generic solution for environments where different types of services and tunnels may co-exist.
- o The framework must consider minimizing disruption during deployment. It should only involve routers close to egress, and be transparent to ingress routers and other transit routers.
- o In egress node protection, for scalability and performance reasons, a PLR must be agnostic to services and service labels, like PLRs in transit link/node protection. It must maintain bypass tunnels and bypass forwarding state on a per-transport-tunnel basis, rather than per-service-destination or per-service-label basis. It should also support bypass tunnel sharing between transport tunnels.
- o A PLR must be able to use its local visibility or information of routing and/or TE topology to compute or resolve a path for a bypass tunnel to a protector.
- o A protector must be able to perform context label switching for rerouted MPLS service packets, based on service label(s) assigned by an egress router. It must be able to perform context IP forwarding for rerouted IP service packets, in the public or private IP address space used by an egress router.

- o The framework must be able to work seamlessly with transit link/node protection mechanisms to achieve end-to-end coverage.
- o The framework must be able to work in conjunction with global repair and control plane convergence.

## 5. Egress node protection

### 5.1. Reference topology

This document refers to the following topology when describing the procedures of egress node protection.

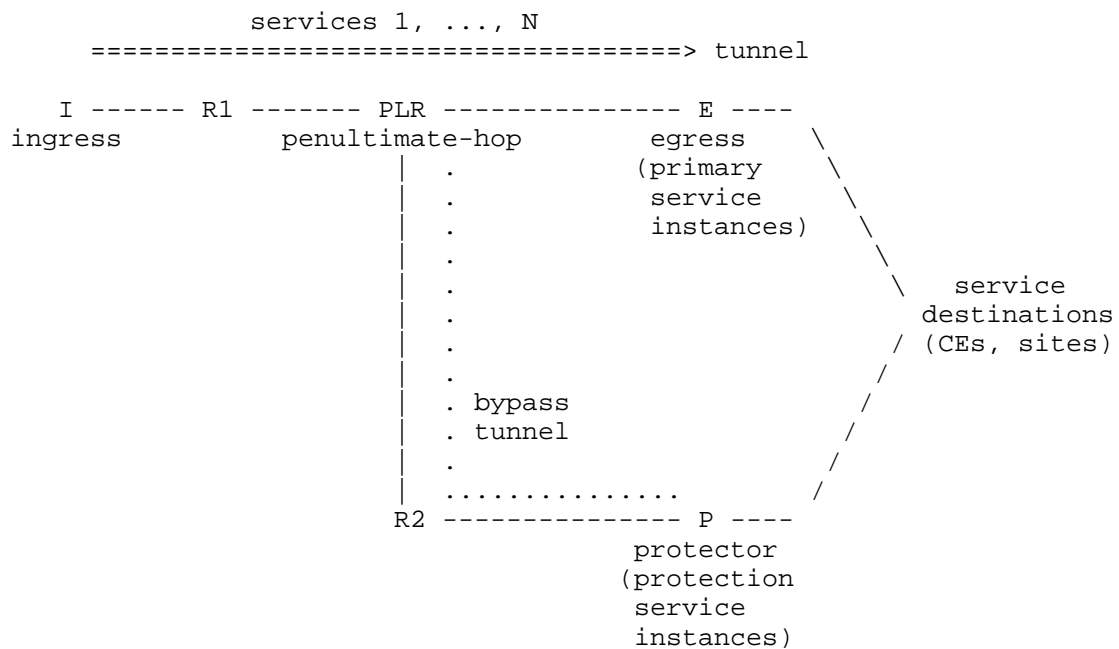


Figure 1

### 5.2. Egress node failure and detection

An egress node failure refers to the failure of an MPLS tunnel's egress router. At the service level, it also means a service instance failure for each IP/MPLS service carried by the tunnel.

Ideally, an egress node failure can be detected by an adjacent router (i.e. PLR in this framework) using a node liveness detection

mechanism, or based on a collective failure of all the links to that node. However, the assumption is that the mechanisms SHOULD be reasonably fast, i.e. faster than control plane failure detection and remote failure detection. Otherwise, local repair will not be able to provide much benefit compared to control plane convergence or global repair. In general, the speed, accuracy, and reliability of a mechanism are the key factors to decide its applicability in egress node protection. This document provides the following guidelines in this regard.

- o If the PLR has a reasonably fast mechanism to detect and differentiate a link failure (of the link between the PLR and the egress node) and an egress node failure, it SHOULD set up both link protection and egress node protection, and trigger one and only one protection upon a corresponding failure.
- o If the PLR has a fast mechanism to detect a link failure and an egress node failure, but cannot distinguish them; Or, if the PLR has a fast mechanism to detect a link failure only, but not an egress node failure, the PLR has two options:
  1. It MAY set up link protection only, and leave the egress node failure to global repair and control plane convergence to handle.
  2. It MAY set up egress node protection only, and treat a link failure as a trigger for the egress node protection. However, the assumption is that treating a link failure as an egress node failure MUST NOT have a negative impact on services. Otherwise, it SHOULD adopt the previous option.

### 5.3. Protector and PLR

A router is assigned to the "protector" role to protect a tunnel and the services carried by the tunnel against an egress node failure. The protector is responsible for hosting a protection service instance for each protected service, serving as the tailend of a bypass tunnel, and performing context label switching and/or context IP forwarding for rerouted service packets.

A tunnel can be protected by only one protector at a given time. Multiple tunnels to a given egress router may be protected by a common protector or different protectors. A protector may protect multiple tunnels with a common egress router or different egress routers.

For each tunnel, its penultimate-hop router acts as a PLR. The PLR pre-establishes a bypass tunnel to the protector, and pre-installs



bypass forwarding state in the data plane. Upon detection of an egress node failure, the PLR reroutes all the service packets received on the tunnel through the bypass tunnel to the protector. For MPLS service packets, the PLR keeps service labels intact in the packets. The protector in turn forwards the rerouted service packets towards the ultimate service destinations. Specifically, it performs context label switching for MPLS service packets, based on service labels assigned by the protected egress router; It performs context IP forwarding for IP service packets, based on their destination addresses.

The protector MUST have its own connectivity with each service destination, via a direct link or a multi-hop path, which MUST NOT traverse the protected egress router or be affected by the egress node failure. This also requires that each service destination MUST be dual-homed or have dual paths to the egress router and a backup egress router which serves as the protector. Each protection service instance on the protector relies on such connectivity to set up forwarding state for context label switching and/or context IP forwarding.

#### 5.4. Protected egress

This document introduces the notion of "protected egress" as a virtual node consisting of the egress router E of a tunnel and a protector P. It is denoted by an ordered pair of {E, P}, indicating the primary-and-protector relationship between the two routers. It serves as the virtual destination of the tunnel, and the virtual location of service instances for the services carried by the tunnel. The tunnel and services are considered as being "associated" with the protected egress {E, P}.

A given egress router E may be the tailend of multiple tunnels. In general, the tunnels may be protected by multiple protectors, e.g. P1, P2, and so on, with each Pi protecting a subset of the tunnels. Thus, these routers form multiple protected egresses, i.e. {E, P1}, {E, P2}, and so on. Each tunnel is associated with one and only one protected egress {E, Pi}. All the services carried by the tunnel are then automatically associated with the same protected egress {E, Pi}. Conversely, a service associated with a protected egress {E, Pi} MUST be carried by a tunnel associated with the protected egress {E, Pi}. This mapping MUST be ensured by the ingress router of the tunnel and the service (Section 5.5).

Two routers X and Y may be protectors for each other. In this case, they form two distinct protected egresses {X, Y} and {Y, X}.

### 5.5. Egress-protected tunnel and service

A tunnel, which is associated with a protected egress {E, P}, is called an egress-protected tunnel. It is associated with one and only one protected egress {E, P}. Multiple egress-protected tunnels may be associated with a given protected egress {E, P}. In this case, they share the common egress router and protector, but may or may not share a common ingress router, or a common PLR (i.e. penultimate-hop router).

An egress-protected tunnel is considered as logically "destined" for its protected egress {E, P}. However, its path MUST be resolved and established with E as the physical tailend.

A service, which is associated with a protected egress {E, P}, is called an egress-protected service. The egress router E hosts the primary instance of the service, and the protector P hosts the protection instance of the service.

An egress-protected service is associated with one and only one protected egress {E, P}. Multiple egress-protected services may be associated with a given protected egress {E, P}. In this case, these services share the common egress router and protector, but may or may not share a common egress-protected tunnel or a common ingress router.

An egress-protected service MUST be mapped to an egress-protected tunnel by its ingress router, based on the common protected egress {E, P} of the service and the tunnel. This is achieved by introducing the notion of "context ID" for protected egress {E, P}, as described in (Section 5.7).

### 5.6. Egress-protection bypass tunnel

An egress-protected tunnel destined for a protected egress {E, P} MUST have a bypass tunnel from its PLR to the protector P. This bypass tunnel is called an egress-protection bypass tunnel. The bypass tunnel is considered as logically "destined" for the protected egress {E, P}. However, due to its bypass nature, it MUST be resolved and established with P as the physical tailend and E as the node to avoid. The bypass tunnel MUST have the property that it MUST NOT be affected by any topology change caused by an egress node failure.

An egress-protection bypass tunnel is associated with one and only one protected egress {E, P}. A PLR may share an egress-protection bypass tunnel for multiple egress-protected tunnels associated with a common protected egress {E, P}. For multiple egress-protected tunnels associated with a common protected egress {E, P}, there may be one or

multiple egress-protection bypass tunnels from one or multiple PLRs to the protector P, depending on the paths of the egress-protected tunnels.

#### 5.7. Context ID, context label, and context based forwarding

In this framework, a globally unique IPv4/v6 address is assigned to a protected egress {E, P} to serve as the identifier of the protected egress {E, P}. It is called a "context ID" due to its specific usage in context label switching and context IP forwarding on the protector. It is an IP address that is logically owned by both the egress router and the protector. For the egress node, it indicates the protector. For the protector, it indicates the egress router, particularly the egress router's forwarding context. For other routers in the network, it is an address reachable via both the egress router and the protector in the routing domain and the TE domain (Section 5.8), similar to an anycast address.

The main purpose of a context ID is to coordinate ingress router, egress router, PLR and protector in setting up egress protection. Given an egress-protected service associated with a protected egress {E, P}, its context ID is used as below:

- o If the service is an MPLS service, when E distributes a service label binding message to the ingress router, E attaches the context ID to the message. If the service is an IP service, when E advertises the service destination address to the ingress router, E also attaches the context ID to the advertisement message. How the context ID is encoded in the messages is a choice of the service protocol, and may need protocol extensions to define a "context ID" object.
- o The ingress router uses the context ID as destination to establish or resolve an egress-protected tunnel. The ingress router then maps the service to the tunnel for transportation. In this process, the special semantics of the context ID is transparent to the ingress router. The ingress router only treats the context ID as an IP address of E, and behaves in the same manner as in establishing or resolving a regular transport tunnel, although the end result is an egress-protected tunnel.
- o The context ID is conveyed to the PLR by the signaling protocol of the egress-protected tunnel, or learned by the PLR via an IGP (i.e. OSPF or ISIS) or a topology-driven label distribution protocol (e.g. LDP). The PLR uses the context ID as destination to establish or resolve an egress-protection bypass tunnel to P while avoiding E.

- o P maintains a dedicated label space or a dedicated IP address space for E, depending on whether the service is MPLS or IP. This is referred to as "E's label space" or "E's IP address space", respectively. P uses the context ID to identify the space.
- o If the service is an MPLS service, E also distributes the service label binding message to P. This is the same label binding message that E advertises to the ingress router, attached with the context ID. Based on the context ID, P installs the service label in an MPLS forwarding table corresponding to E's label space. If the service is an IP service, P installs an IP route in an IP forwarding table corresponding to E's IP address space. In either case, the protection service instance on P interprets the service and constructs forwarding state for the route based on P's own connectivity to the service's destination.
- o P assigns a non-reserved label to the context ID. In the data plane, this label represents the context ID and indicates E's label space and IP address space. Therefore, it is called a "context label".
- o The PLR may establish the egress-protection bypass tunnel to P in several manners. If the bypass tunnel is established by RSVP, the PLR signals the bypass tunnel with the context ID as destination, and P binds the context label to the bypass tunnel. If the bypass tunnel is established by LDP, P advertises the context label for the context ID as an IP prefix FEC. If the bypass tunnel is established by the PLR in a hierarchical manner, the PLR treats the context label as a one-hop LSP over a regular bypass tunnel to P (e.g. a bypass tunnel to P's loopback IP address). If the bypass tunnel is constructed by using segment routing, the bypass tunnel is represented by a stack of SID labels with the context label as the inner-most SID label (Section 5.9). In any case, the bypass tunnel is a UHP tunnel whose incoming label at P is the context label.
- o During local repair, all the service packets received by P on the bypass tunnel have the context label as top label. P first pops the context label. For an MPLS service packet, P further looks up the service label in E's label space indicated by the context label, which is called context label switching. For an IP service packet, P looks up the IP destination address in E's IP address space indicated by the context label, which is called context IP forwarding.

## 5.8. Advertisement and path resolution for context ID

Path resolution and computation for a context ID are done on ingress routers for egress-protected tunnels, and on PLRs for egress-protection bypass tunnels. Therefore, given a protected egress {E, P} and its context ID, E and P MUST coordinate the context ID in the routing domain and the TE domain via IGP advertisement. The context ID MUST be advertised in such a manner that all egress-protected tunnels MUST have E as tailend, and all egress-protection bypass tunnels MUST have P as tailend while avoiding E.

This document suggests two approaches:

1. The first approach is called "proxy mode". It requires E and P, but not the PLR, to have the knowledge of the egress protection schema. E and P advertise the context ID as a virtual proxy node (i.e. a logical node) connected to the two routers, with the link between the proxy node and E having more preferable IGP and TE metrics than the link between the proxy node and P. Therefore, all egress-protected tunnels destined for the context ID should automatically follow the shortest IGP or TE paths to E. Each PLR will no longer view itself as a penultimate-hop, but rather two hops away from the proxy node, via E. The PLR will be able to find a bypass path via P to the proxy node, while the bypass tunnel should actually be terminated by P.
2. The second approach is called "alias mode". It requires P and the PLR, but not E, to have the knowledge of the egress protection schema. E simply advertises the context ID as a regular IP address. P advertises the context ID and the context label by using a "context ID label binding" advertisement. The advertisement MUST be understood by the PLR. In both routing domain and TE domain, the context ID is only reachable via E. This ensures that all egress-protected tunnels destined for the context ID should have E as tailend. Based on the "context ID label binding" advertisement, the PLR can establish an egress-protection bypass tunnel in several manners (Section 5.9). The "context ID label binding" advertisement is defined as IGP mirroring context segment in [SR-ARCH], [SR-OSPF] and [SR-ISIS]. These IGP extensions are generic in nature, and hence can be used for egress protection purposes.

In a scenario where an egress-protected tunnel is an inter-area or inter-AS tunnel, its associated context ID MUST be propagated from the residing area/AS to the other areas/AS' via IGP or BGP, so that the ingress router of the tunnel can have the reachability to the context ID. The propagation process of the context ID SHOULD be the same as that of a regular IP address in an inter-area/AS environment.

### 5.9. Egress-protection bypass tunnel establishment

A PLR MUST know the context ID of a protected egress {E, P} in order to establish an egress-protection bypass tunnel. The information is obtained from the signaling or label distribution protocol of the egress-protected tunnel. The PLR may or may not need to have the knowledge of the egress protection schema. All it does is to set up a bypass tunnel to a context ID while avoiding the next-hop router (i.e. egress router). This is achievable by using a constraint-based computation algorithm similar to those which are commonly used in the computation of traffic engineering paths and loop-free alternate (LFA) paths. Since the context ID is advertised in the routing domain and the TE domain by IGP according to Section 5.8, the PLR should be able to resolve or establish such a bypass path with the protector as tailend. In some cases like the proxy mode, the PLR may do so in the same manner as transit node protection.

An egress-protection bypass tunnel may be established via several methods:

- (1) It may be established by a signaling protocol (e.g. RSVP), with the context ID as destination. The protector binds the context label to the bypass tunnel.
- (2) It may be formed by a topology driven protocol (e.g. LDP with various LFA mechanisms). The protector advertises the context ID as an IP prefix FEC, with the context label bound to it.
- (3) It may be constructed as a hierarchical tunnel. When the protector uses the alias mode (Section 5.8), the PLR will have the knowledge of the context ID, context label, and protector (i.e. the advertiser). The PLR can then establish the bypass tunnel in a hierarchical manner, with the context label as a one-hop LSP over a regular bypass tunnel to the protector's IP address (e.g. loopback address). This regular bypass tunnel may be established by RSVP, LDP, segment routing, and others.

### 5.10. Local repair on PLR

In this framework, a PLR is agnostic to services and service labels. This obviates the need to maintain bypass forwarding state on a per-service basis, and allows bypass tunnel sharing between egress-protected tunnels. The PLR may share an egress-protection bypass tunnel for multiple egress-protected tunnels associated with a common protected egress {E, P}. During local repair, the PLR reroutes all service packets received on the egress-protected tunnels via the egress-protection bypass tunnel. Service labels remain intact in MPLS service packets.

Label operation during the rerouting depends on the bypass tunnel's characteristics. If the bypass tunnel is a single level tunnel, the rerouting will involve swapping the incoming label of an egress-protected tunnel to the outgoing label of the bypass tunnel. If the bypass tunnel is a hierarchical tunnel, the rerouting will involve swapping the incoming label of an egress-protected tunnel to a context label, and pushing the outgoing label of a regular bypass tunnel. If the bypass tunnel is constructed by segment routing, the rerouting will involve swapping the incoming label of an egress-protected tunnel to a context label, and pushing a stack of SID labels of the bypass tunnel.

#### 5.11. Service label distribution from egress router to protector

As mentioned in previous sections, when a protector receives a rerouted MPLS service packet, it performs context label switching based on the packet's service label which is assigned by the corresponding egress router. In order to achieve this, the protector MUST maintain such kind of service labels in dedicated label spaces on a per protected egress {E, P} basis, i.e. one label space for each egress router that it protects.

Also, there MUST be a service label distribution protocol session between each egress router and the protector. Through this protocol, the protector learns the label binding of each egress-protected service. This is the same label binding that the egress router advertises to the corresponding ingress router, attached with a context ID. The corresponding protection service instance on the protector recognizes the service, and resolves forwarding state based on its own connectivity with the service's destination. It then installs the service label with the forwarding state in the label space of the egress router, which is indicated by the context ID (i.e. context label).

Different service protocols may use different mechanisms for such kind of label distribution. Specific protocol extensions may be needed on a per-protocol basis or per-service-type basis. The details of the extensions SHOULD be specified in separate documents. As an example, RFC 8104 specifies the LDP extensions for pseudowire services.

#### 5.12. Centralized protector mode

In this framework, it is assumed that the service destination of an egress-protected service MUST be dual-homed to two edge routers of an MPLS network. One of them is the protected egress router, and the other is a backup egress router. So far in this document, the discussion has been focusing on the scenario where a protector and a

backup egress router are co-located as one router. Therefore, the number of protectors in a network is equal to the number of backup egress routers. As another scenario, a network may assign a small number of routers to serve as dedicated protectors, each protecting a subset of egress routers. These protectors are called centralized protectors.

Topologically, a centralized protector may be decoupled from all backup egress routers, or it may be co-located with one backup egress router while decoupled from the other backup egress routers. The procedures in this section assume the scenario where a protector and a backup egress router are decoupled.

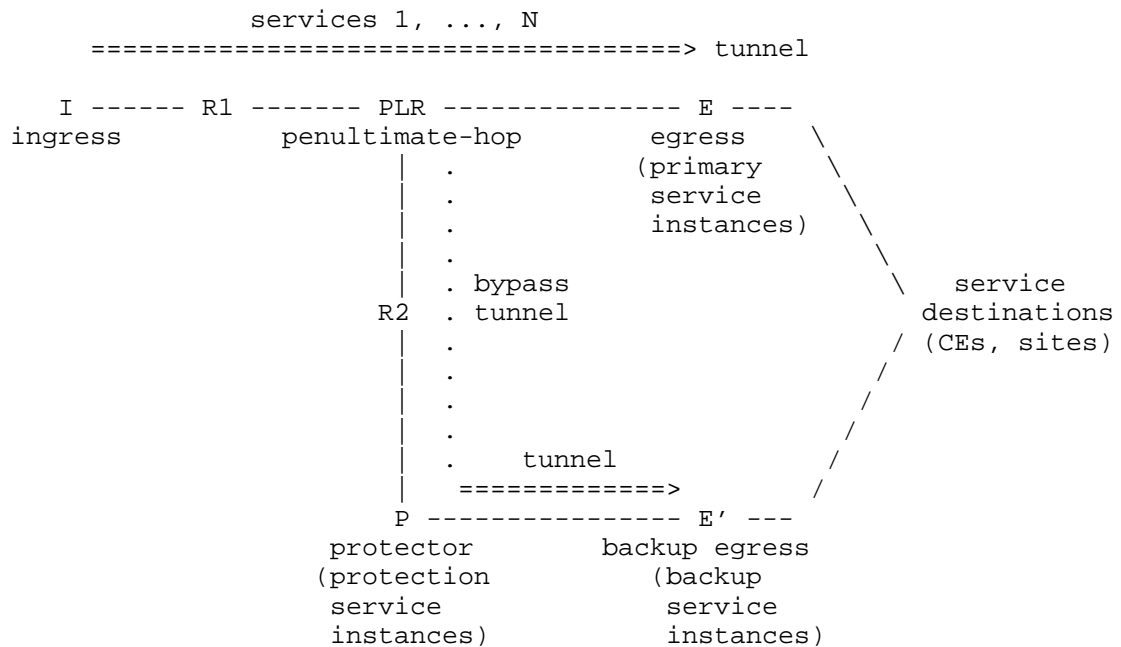


Figure 2

Like a co-located protector, a centralized protector hosts protection service instances, receives rerouted service packets from PLRs, and performs context label switching and/or context IP forwarding. For each service, instead of sending service packets directly to the service destination, the protector MUST send them via another transport tunnel to the corresponding backup service instance on a backup egress router. The backup service instance in turn forwards



them to the service destination. Specifically, in the case of an MPLS service, the protector MUST swap the service label in each received service packet to the label of the backup service advertised by the backup egress router, and then push the label (or label stack) of the transport tunnel.

In order for a centralized protector to map an egress-protected MPLS service to a service hosted on a backup egress router, there MUST be a service label distribution protocol session between the backup egress router and the protector. Through this session, the backup egress router advertises the service label of the backup service, attached with the FEC of the egress-protected service and the context ID of the protected egress {E, P}. Based on this information, the protector associates the egress-protected service with the backup service, resolves or establishes a transport tunnel to the backup egress router, and accordingly sets up forwarding state for the label of the egress-protected service in the label space of the egress router.

The service label which the backup egress router advertises to the protector can be the same as the label which the backup egress router advertises to the ingress router(s), if and only if the forwarding state of the label does not direct service packets towards the protected egress router. Otherwise, the label is not usable for egress protection, because it will create a loop, which MUST be avoided. In this case, the backup egress router MUST advertise a unique service label for egress protection, and set its forwarding state to use the backup egress router's connectivity with the service destination.

## 6. Egress link protection

Egress link protection is achievable through procedures similar to that of egress node protection. In normal situations, an egress router forwards service packets to a service destination based on a service label, whose forwarding state points to an egress link. In egress link protection, the egress router acts as PLR, by performing local failure detection and local repair. Specifically, the egress router pre-establishes an egress-protection bypass tunnel to a protector, and installs bypass forwarding state for the service label, pointing to the bypass tunnel. During local repair, the egress router reroutes service packets via the bypass tunnel to the protector. The protector in turn forwards the packets to the service destination (in the co-located protector mode, as shown in Figure-3), or forwards the packets to a backup egress router (in the centralized protector mode, as shown in Figure-4).

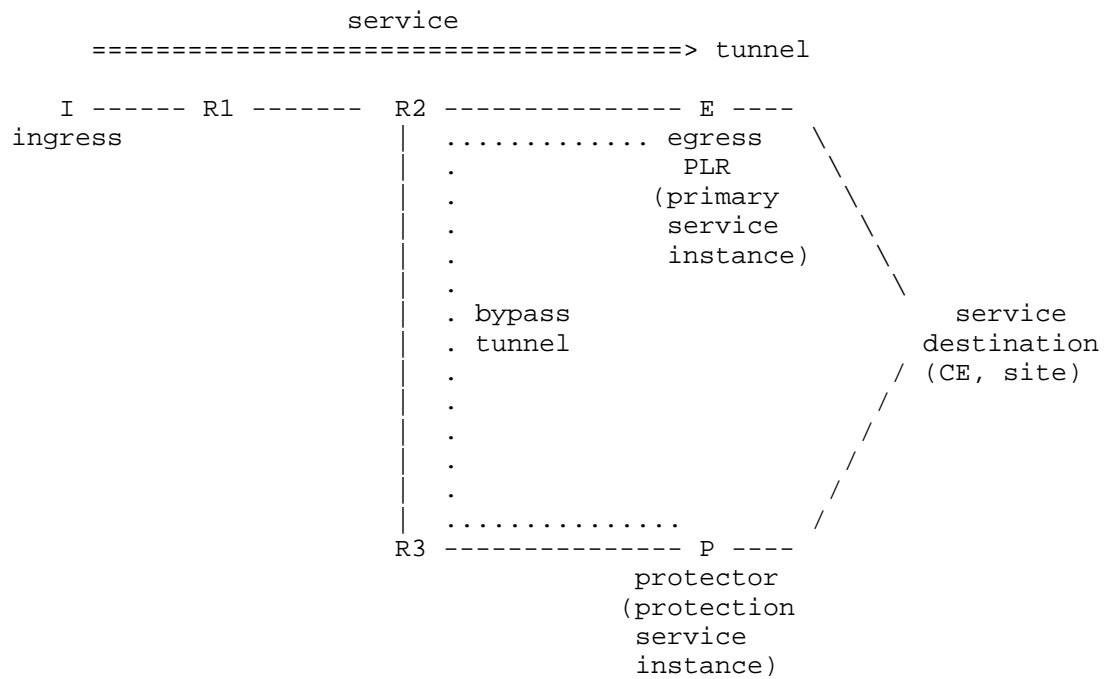


Figure 3

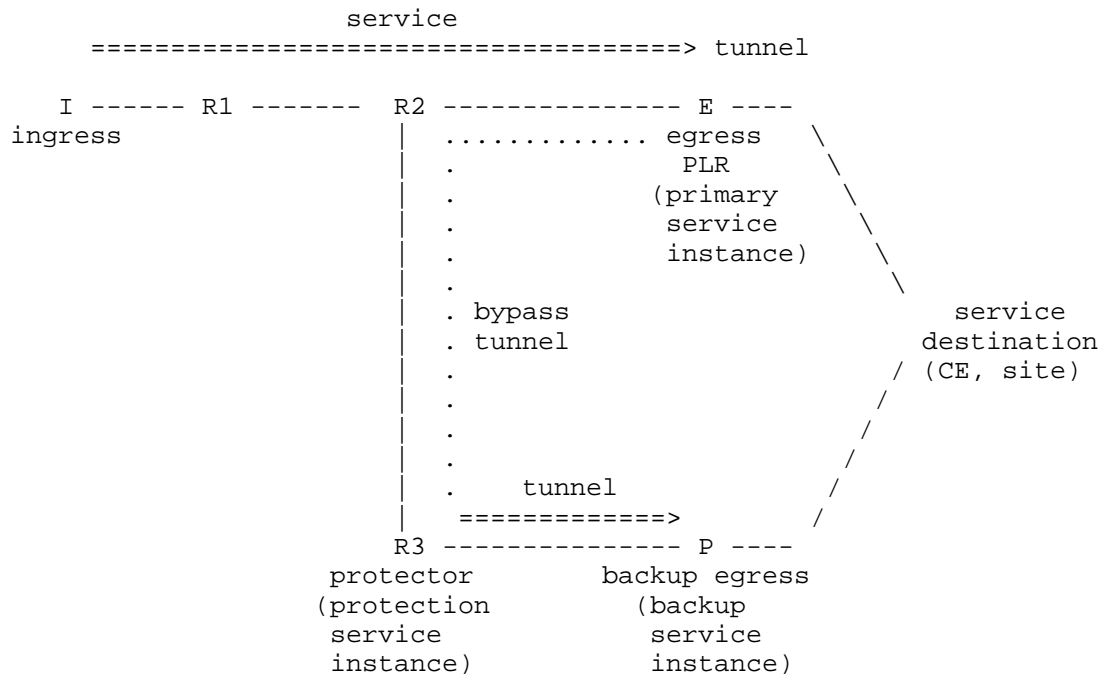


Figure 4

There are two approaches to set up the bypass forwarding state on the egress router, depending on whether the egress router knows the service label advertised by the backup egress router. The difference is that one approach requires the protector to perform context label switching, and the other one does not. Both approaches are equally supported by this framework, and may be used in parallel.

(1) The first approach applies when the egress router does not know the service label advertised by the backup egress router. In this case, the egress router sets up the bypass forwarding state as a label push with the outgoing label of the egress-protection bypass tunnel. Rerouted packets will have the egress router's service label intact. Therefore, the protector MUST perform context label switching, and the bypass tunnel MUST be destined for the context ID of the {E, P} and established as described in Section 5.9. This approach is consistent with egress node protection. Hence, a protector can serve in egress node and egress link protection in a consistent manner, and both the co-located protector mode and the centralized protector mode may be used (Figure-3 and Figure-4).

(2) The second approach applies when the egress router knows the service label advertised by the backup egress route, via a label distribution protocol session. In this case, the backup egress router serves as the protector for egress link protection, regardless of the protector of egress node protection, which should be the same router in the co-located protector mode but may be a different router in the centralized protector mode. The egress router sets up the bypass forwarding state as a label swap from the incoming service label to the service label of the protector, followed by a push with the outgoing label (or label stack) of the egress link protection bypass tunnel. The bypass tunnel is a regular tunnel destined for an IP address of the protector, instead of the context ID of the {E, P}. The protector simply forwards rerouted service packets based on its own service label, rather than performing context label switching. With this approach, only the co-located protector mode is applicable.

Note that for a bidirectional service, the physical link of an egress link may carry service traffic bi-directionally. Therefore, an egress link failure may simultaneously be an ingress link failure for the traffic in the opposite direction. However, protection for ingress link failure SHOULD be provided by a separate mechanism, and hence is out of the scope of this document.

#### 7. Global repair

This framework provides a fast but temporary repair for egress node and egress link failures. For permanent repair, it is RECOMMENDED that the traffic SHOULD be moved to an alternative tunnel or alternative services which are fully functional. This is referred to as global repair. Possible triggers of global repair include control plane notifications of tunnel and service status, end-to-end OAM and fault detection at tunnel or service levels, and others. The alternative tunnel and services may be pre-established as standby, or dynamically established as a result of the triggers or network protocol convergence.

#### 8. Example: Layer-3 VPN egress protection

This section shows an example of egress protection for a layer-3 VPN.

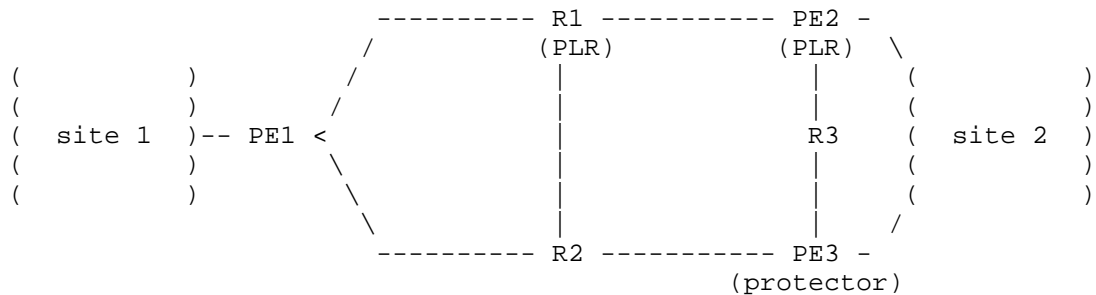


Figure 5

In this example, the site 1 (subnet 203.0.113.192/26) of a given VPN is attached to PE1, and site 2 (subnet 203.0.113.128/26) is dual-homed to PE2 and PE3. PE2 is the primary PE for site 2, and PE3 is the backup PE. Each PE hosts a VPN instance. R1 and R2 are transit routers in the MPLS network. The network uses OSPF as routing protocol, and RSVP-TE as tunnel signaling protocol. The PEs use BGP to exchange VPN prefixes and VPN labels between each other.

Using the framework in this document, the network assigns PE3 to be a protector for PE2 to protect the VPN traffic in the direction from site 1 to site 2. This is the co-located protector mode. Hence, PE2 and PE3 form a protected egress {PE2, PE3}. A context ID 198.51.100.1 is assigned to the protected egress {PE2, PE3}. The VPN instance on PE3 serves as a protection instance for the VPN instance on PE2. On PE3, a context label 100 is assigned to the context ID, and a label table pe2.mpls is created to represent PE2's label space. PE3 installs the label 100 in its default MPLS forwarding table, with nexthop pointing to the label table pe2.mpls. PE2 and PE3 are coordinated to use the proxy mode to advertise the context ID in the routing domain and the TE domain.

PE2 uses per-VRF VPN label allocation mode. It assigns a single label 9000 to the VRF of the VPN. For a given VPN prefix 203.0.113.128/26 in site 2, PE2 advertises it along with the label 9000 and other attributes to PE1 and PE3 via BGP. In particular, the NEXT\_HOP attribute is set to the context ID 198.51.100.1.

Similarly, PE3 also uses per-VRF VPN label allocation mode. It assigns a single label 10000 to the VRF of the VPN. For the VPN prefix 203.0.113.128/26 in site 2, PE3 advertises it along with the label 10000 and other attributes to PE1 and PE2 via BGP. In particular, the NEXT\_HOP attribute is set to an IP address of PE3.

Upon receipt and acceptance of the BGP advertisement, PE1 uses the context ID 198.51.100.1 as destination to compute a TE path for an egress-protected tunnel. The resulted path is PE1->R1->PE2. PE1 then uses RSVP to signal the tunnel, with the context ID 198.51.100.1 as destination, and with the "node protection desired" flag set in the SESSION\_ATTRIBUTE of RSVP Path message. Once the tunnel comes up, PE1 maps the VPN prefix 203.0.113.128/26 to the tunnel and installs a route for the prefix in the corresponding VRF. The route's nexthop is a push with the VPN label 9000, followed by a push with the outgoing label of the egress-protected tunnel.

Upon receipt of the above BGP advertisement from PE2, PE3 (i.e. the protector) recognizes the context ID 198.51.100.1 in the NEXT\_HOP attribute, and installs a route for label 9000 in the label table pe2.mpls. PE3 sets the route's nexthop to a "protection VRF". This protection VRF contains IP routes corresponding to the IP prefixes in the dual-homed site 2, including 203.0.113.128/26. The nexthops of these routes MUST be based on PE3's connectivity with site 2, even if this connectivity is not the best path in PE3's VRF due to metrics (e.g. MED, local preference, etc.), and MUST NOT use any path traversing PE2. Note that the protection VRF is a logical concept, and it may simply be PE3's own VRF if the VRF satisfies the requirement.

#### 8.1. Egress node protection

R1, i.e. the penultimate-hop router of the egress-protected tunnel, serves as the PLR for egress node protection. Based on the "node protection desired" flag and the destination address (i.e. context ID 198.51.100.1) of the tunnel, R1 computes a bypass path to 198.51.100.1 while avoiding PE2. The resulted bypass path is R1->R2->PE3. R1 then signals the path (i.e. egress-protection bypass tunnel), with 198.51.100.1 as destination.

Upon receipt of an RSVP Path message of the egress-protection bypass tunnel, PE3 recognizes the context ID 198.51.100.1 as the destination, and hence responds with the context label 100 in an RSVP Resv message.

After the egress-protection bypass tunnel comes up, R1 installs a bypass nexthop for the egress-protected tunnel. The bypass nexthop is a swap from the incoming label of the egress-protected tunnel to the outgoing label of the egress-protection bypass tunnel.

When R1 detects a failure of PE2, it will invoke the above bypass nexthop to reroute VPN service packets. The packets will have the label of the bypass tunnel as outer label, and the VPN label 9000 as inner label. When the packets arrive at PE3, they will have the

context label 100 as outer label, and the VPN label 9000 as inner label. The context label will first be popped, and then the VPN label will be looked up in the label table pe2.mpls. The lookup will cause the VPN label to be popped, and the IP packets will finally be forwarded to site 2 based on the protection VRF.

## 8.2. Egress link protection

PE2 serves as the PLR for egress link protection. It has already learned the VPN label 10000 from PE3, and hence it uses the approach (2) described in Section 6 to set up bypass forwarding state. It signals an egress-protection bypass tunnel to PE3, by using the path PE2->R3->PE3, and PE3's IP address as destination. After the bypass tunnel comes up, PE2 installs a bypass nexthop for the VPN label 9000. The bypass nexthop is a label swap from the incoming label 9000 to the VPN label 10000 of PE3, followed by a label push with the outgoing label of the bypass tunnel.

When PE3 detects a failure of the egress link, it will invoke the above bypass nexthop to reroute VPN service packets. The packets will have the label of the bypass tunnel as outer label, and the VPN label 10000 as inner label. When the packets arrive at PE3, the VPN label 10000 will be popped, and the IP packets will be forwarded based on the VRF indicated by on the VPN label 10000.

## 8.3. Global repair

Eventually, global repair will take effect, as control plane protocols converge on the new topology. PE1 will choose PE3 as new entrance to site 2. Before that happens, the VPN traffic has been protected by the above local repair.

## 9. IANA Considerations

This document has no request for new IANA allocation.

## 10. Security Considerations

The framework in this document relies on fast reroute around a network failure. Specifically, service traffic is temporarily rerouted from a PLR to a protector. In the centralized protector mode, the traffic is further rerouted from the protector to a backup egress router. Such kind of fast reroute is planned and anticipated, and hence it should not be viewed as a new security threat.

The framework requires a service label distribution protocol to run between an egress router and a protector. The available security

measures of the protocol MAY be used to achieve a secured session between the two routers.

## 11. Acknowledgements

This document leverages work done by Yakov Rekhter, Kevin Wang and Zhaohui Zhang on MPLS egress protection. Thanks to Alexander Vainshtein, Rolf Winter, and Lizhong Jin for their valuable comments that helped shape this document and improve its clarity.

## 12. References

### 12.1. Normative References

- [SR-ARCH] Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing (work in progress), 2017.
- [SR-OSPF] Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions (work in progress), 2017.
- [SR-ISIS] Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions (work in progress), 2017.

### 12.2. Informative References

- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <<https://www.rfc-editor.org/info/rfc5286>>.
- [RFC7490] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <<https://www.rfc-editor.org/info/rfc7490>>.



- [RFC7812] Atlas, A., Bowers, C., and G. Enyedi, "An Architecture for IP/LDP Fast Reroute Using Maximally Redundant Trees (MRT-FRR)", RFC 7812, DOI 10.17487/RFC7812, June 2016, <<https://www.rfc-editor.org/info/rfc7812>>.
- [RFC8104] Shen, Y., Aggarwal, R., Henderickx, W., and Y. Jiang, "Pseudowire (PW) Endpoint Fast Failure Protection", RFC 8104, DOI 10.17487/RFC8104, March 2017, <<https://www.rfc-editor.org/info/rfc8104>>.
- [BGP-PIC] Bashandy, P., Filsfils, C., and P. Mohapatra, "BGP Prefix Independent Convergence", draft-ietf-rtgwg-bgp-pic-05.txt (work in progress), 2017.
- [RSVP-EP] Chen, H., Liu, A., Saad, T., Xu, F., Huang, L., and N. So, "Extensions to RSVP-TE for LSP Egress Local Protection", draft-ietf-teas-rsvp-egress-protection (work in progress), 2017.

## Authors' Addresses

Yimin Shen  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
USA

Phone: +1 9785890722  
Email: [yshen@juniper.net](mailto:yshen@juniper.net)

Minto Jeyananth  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
USA

Phone: +1 4089367563  
Email: [minto@juniper.net](mailto:minto@juniper.net)

Bruno Decraene  
Orange

Email: [bruno.decraene@orange.com](mailto:bruno.decraene@orange.com)

Hannes Gredler  
RtBrick Inc

Email: hannes@rtbrick.com

Carsten Michel  
Deutsche Telekom

Email: c.michel@telekom.de

Huaimo Chen  
Huawei Technologies Co., Ltd.

Email: huaimo.chen@huawei.com

Yuanlong Jiang  
Huawei Technologies Co., Ltd.  
Bantian, Longgang district  
Shenzhen 518129  
China

Email: jiangyuanlong@huawei.com

MPLS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 11, 2017

H. Sitaraman  
V. Beeram  
Juniper Networks  
T. Parikh  
Verizon  
March 10, 2017

Signaling RSVP-TE tunnels on a shared MPLS forwarding plane  
draft-sitaraman-mpls-rsvp-shared-labels-00.txt

Abstract

As the scale of MPLS RSVP-TE LSPs has grown, various implementation recommendations have been proposed to manage control plane state. However, the forwarding plane footprint of labels at a transit LSR has remained proportional to the total LSP state in the control plane. This draft defines a mechanism to prevent the label space limit on an LSR from being a constraint to control plane scaling on that node. It introduces the notion of pre-installed per TE link 'pop labels' that are shared by MPLS RSVP-TE LSPs that traverse these links and thus significantly reducing the forwarding plane state required. This couples the feature benefits of the RSVP-TE control plane with the simplicity of the Segment Routing MPLS forwarding plane. This document also introduces the ability to mix different types of label operations along the path of the LSP, thereby allowing the ingress or an external controller to influence how to optimally place a LSP.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 11, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                |    |
|----------------------------------------------------------------|----|
| 1. Introduction . . . . .                                      | 3  |
| 2. Conventions used in this document . . . . .                 | 4  |
| 3. Terminology . . . . .                                       | 4  |
| 4. Allocation of pop labels . . . . .                          | 4  |
| 5. RSVP-TE pop and forward tunnel setup . . . . .              | 4  |
| 6. Mixing pop and swap labels in a RSVP-TE tunnel . . . . .    | 6  |
| 7. Distributing label stack imposition . . . . .               | 7  |
| 8. Facility backup protection . . . . .                        | 7  |
| 8.1. Link Protection . . . . .                                 | 7  |
| 8.2. Node Protection . . . . .                                 | 8  |
| 9. Quantifying pop labels . . . . .                            | 8  |
| 10. Protocol Extensions . . . . .                              | 9  |
| 10.1. Requirements . . . . .                                   | 9  |
| 10.2. Attributes Flags TLV: Pop Label . . . . .                | 9  |
| 10.3. RRO Label Subobject Flag: Pop Label . . . . .            | 10 |
| 10.4. Attributes TLV: Label Stack Imposition TLV . . . . .     | 10 |
| 11. OAM considerations . . . . .                               | 11 |
| 12. Acknowledgements . . . . .                                 | 11 |
| 13. Contributors . . . . .                                     | 11 |
| 14. IANA Considerations . . . . .                              | 11 |
| 14.1. Attribute Flags: Pop Label . . . . .                     | 11 |
| 14.2. Attribute TLV: Label Stack Imposition TLV . . . . .      | 11 |
| 14.3. Record Route Label Sub-object Flags: Pop Label . . . . . | 12 |
| 15. Security Considerations . . . . .                          | 12 |
| 16. References . . . . .                                       | 12 |
| 16.1. Normative References . . . . .                           | 12 |
| 16.2. Informative References . . . . .                         | 13 |
| Authors' Addresses . . . . .                                   | 14 |

## 1. Introduction

Various RSVP-TE scaling recommendations [RFC2961] [I-D.ietf-teas-rsvp-te-scaling-rec] have been proposed for implementations to adopt guidelines that would allow the RSVP-TE [RFC3209] control plane to scale better. The forwarding plane state required to handle the equivalent control plane state remains unchanged and is proportional to the total LSP state in the control plane. The motivation of this draft is to prevent the platform specific label space limit on an LSR from being a constraint to pushing the limits of control plane scaling on that node.

This document proposes the allocation of a 'pop label' by a LSR for each of its TE links. The label is installed in the MPLS forwarding plane with a pop label operation and to forward the received packet over the TE link. This label is sent normally by the LSR in the Label object in the Resv message as LSPs are setup. The ingress LER SHOULD construct and push a stack of labels [RFC3031] as received in the Record Route object (RRO) in the Resv message.

This pop and forward data plane behavior is similar to that used by Segment Routing (SR) [I-D.ietf-spring-segment-routing] using a MPLS forwarding plane and a series of adjacency segments. The RSVP-TE pop and forward tunnels can co-exist with SR LSPs as described in [I-D.sitaraman-sr-rsvp-coexistence-rec].

RSVP-TE using a pop and forward data plane offers the following benefits:

1. Shared forwarding plane: The transit label on a TE link is shared among RSVP-TE tunnels traversing the link and is used independent of the ingress and egress of the LSPs.
2. Faster LSP setup time: The forwarding plane state is not programmed during LSP setup and teardown resulting in faster LSP setup time.
3. Hitless routes: New transit labels are not required on complete path overlap during make-before-break (MBB) resulting in a faster MBB event. This avoids the ingress LER and the services that might be using the tunnel from needing to update its forwarding plane with new tunnel labels. Periodic MBB events are relatively common in networks that deploy auto-bandwidth on RSVP-TE LSPs to monitor bandwidth utilization and periodically adjust LSP bandwidth.
4. Mix and match labels: Both 'pop' and 'swap' labels can be mixed across transit hops for a single RSVP-TE tunnel (see Section 6).

This allows local policy at an ingress or path computation engine to influence RSVP-TE to mix and match different types of labels across a LSP path.

No additional extensions are required to IGP-TE in order to support this pop and forward data plane. Functionalities such as bandwidth admission control, LSP priorities, preemption, auto-bandwidth and Fast Reroute continue to work with this forwarding plane.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 3. Terminology

Pop label: An incoming label at a LSR that will be popped and forwarded over a specific TE link to a neighbor.

Swap label: An incoming label at a LSR that will be swapped to an outgoing label and forwarded over a specific downstream TE link.

Pop and forward data plane: A forwarding plane where every LSR along the path uses a pop label.

RSVP-TE pop and forward tunnel: A MPLS RSVP-TE tunnel that uses a pop and forward data plane.

## 4. Allocation of pop labels

A LSR SHOULD allocate a unique pop label for each TE link. The forwarding action for the pop label should it appear on top of the label stack MUST be to pop the label and forward the packet over the TE link to the downstream neighbor of the RSVP-TE tunnel. Multiple labels MAY be allocated for the TE link to accommodate tunnels requesting no protection, link-protection and node-protection over the specific TE link.

## 5. RSVP-TE pop and forward tunnel setup

This section provides an example of how the RSVP-TE signaling procedure works to setup a tunnel utilizing a pop and forward data plane. The sample topology below will be used to explain the setup.

Labels shown at each node are pop labels for that neighbor

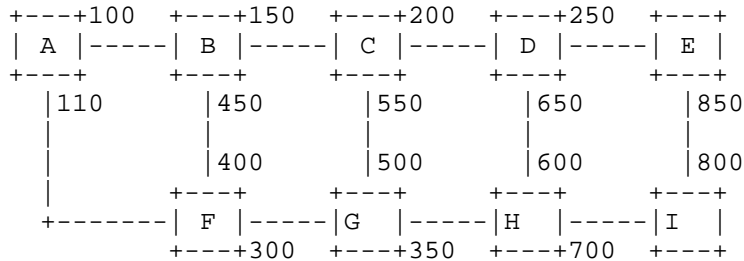


Figure 1: Pop and forward label topology

RSVP-TE tunnel T1: From A to E on path A-B-C-D-E

RSVP-TE tunnel T2: From F to E on path F-B-C-D-E

Both tunnels share the TE links B-C, C-D and D-E.

As RSVP-TE signals the setup (using the pop label attributes flag defined in Section 10.2) of tunnel T1, when LSR D receives the Resv message from the egress E, it checks the next-hop TE link (D-E) and provides the pop label (250) in the Resv message for the tunnel. The label is sent in the Label object and is also recorded in the Label sub-object (using the pop label bit defined in Section 10.3) carried in the RRO. Similarly, C provides the pop label (200) for the next-hop TE link C-D and B provides the pop label (150) for the next-hop TE link B-C. For the tunnel T2, the transit LSRs provide the same pop labels as described for tunnel T1.

Both LER A and F will push the same stack of labels {150(top), 200, 250} for tunnels T1 and T2 respectively. It should be noted that a transit LSR does not use the pop label provided in the label object by its downstream LSR in the NHLFE as the outgoing label. The recorded labels in the RRO are of interest to the ingress LER in order to construct a stack of labels.

If there were another RSVP-TE tunnel T3 from F to I on path F-B-C-D-E-I, then this would also share the TE links B-C, C-D and D-E and additionally traverse link E-I. The label stack used by F would be {150(top), 200, 250, 850}. Hence, regardless of the ingress and egress LERs from where the LSPs start and end, they will share LSR labels at shared hops in the pop and forward data plane.

There MAY be local operator policy at the ingress LER that influences the maximum depth of the label stack that can be pushed for a RSVP-TE pop and forward tunnel. Prior to signaling the LSP, if the ingress LER decides that it would be unable to push the entire label stack

should every transit hop provide a pop label, then the LER can choose to either not signal a RSVP-TE pop and forward tunnel or can adopt techniques mentioned in Section 6 or Section 7.

## 6. Mixing pop and swap labels in a RSVP-TE tunnel

Labels can be mixed across transit hops in a single MPLS RSVP-TE LSP. Certain LSRs can use pop labels and others can use swap labels. The ingress can construct a label stack appropriately based on what type of label is recorded from every transit LSR.

Labels shown at each node are pop labels for that TE link. (#) are swap labels.

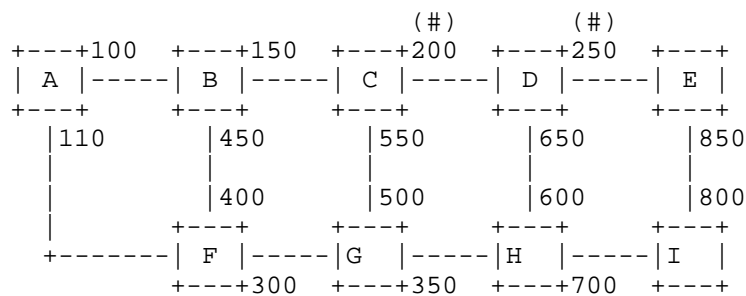


Figure 2: Mix pop and swap label topology

If the transit LSR is allocating a swap label to be sent upstream in the Resv, then the label operation in the NHLFE MUST be a swap to any label received from the downstream LSR. If the transit LSR is using a pop label to be sent upstream in the Resv, then the label operation in the NHLFE MUST be a pop and forward regardless of any label received from the downstream LSR.

The ingress LER MUST check the type of label received from each transit hop as recorded in the RRO in the Resv message and generate the appropriate label stack to use for the RSVP-TE tunnel.

The following logic could be used by the ingress LER while constructing the label stack:

Each RRO label sub-object SHOULD be processed starting with the label sub-object from the first downstream hop. Any label provided by the first downstream hop MUST always be pushed on the label stack regardless of the label type. If the label type is a pop label, then any label from the next downstream hop MUST also be pushed on the constructed label stack. If the label type is a swap label, then any label from the next downstream hop MUST NOT be pushed on the



constructed label stack. For example, the LSP from A to I using path A-B-C-D-E-I will use a label stack of {150(top), 200}.

Signaling extensions for the ingress LER to request a certain type of label from a particular hop is defined in Section 10.2. A Hop-Count value of 1 (Label Stack Imposition Attribute) SHOULD be used for the specific hops to allocate a swap label.

## 7. Distributing label stack imposition

One or more transit LSRs can assist the ingress LER by imposing part of the label stack required for the path. From Figure 1, ingress LER A can use the assistance of transit LSRs to push labels downstream of that LSR. For example, LER A can push label 150 and LSR C can push {200(top), 250} for the LSP taking path A-B-C-D-E.

The ingress LER can request one or more specific transit hops to handle pushing labels for N of its downstream hops. To achieve this request properly, the ingress can learn the label stack depth push limit of the transit LSRs. The mechanism by which the ingress or controller (hosting the path computation element) learns this information is outside the scope of this document. The particular transit hops SHOULD allocate a swap label that will result in that label being replaced and a set of labels pushed to accommodate N downstream hops.

Signaling extensions for the ingress LER to request one or more transit LSRs to handle label stack imposition for N downstream hops or for the transit hop to indicate to the ingress that it can handle label stack imposition for N downstream hops is defined in Section 10.2. The Hop-Count field (Label Stack Imposition Attribute) can be used to indicate the value of N.

## 8. Facility backup protection

The following section describe how link and node protection works with facility backup protection [RFC4090] for the RSVP-TE pop and forward tunnels.

### 8.1. Link Protection

To provide link protection at a PLR with a pop and forward data plane, the LSR SHOULD allocate a separate pop label for the TE link that will be used for RSVP-TE tunnels that request link-protection from the ingress. No signaling extensions are required to support link protection for RSVP-TE tunnels over the pop and forward data plane.

(\*) are pop labels to offer link protection for that TE link

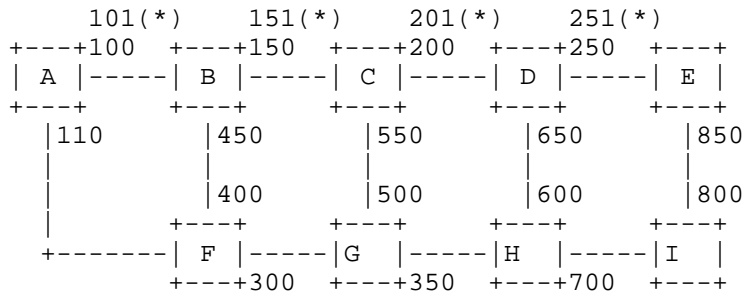


Figure 3: Link protection topology

At each LSR, link protected pop labels can be allocated for each TE link and a link protecting facility backup LSP can be created to protect the TE link. This label can be sent by the LSR for LSPs requesting link-protection over the specific TE link. Since the facility backup terminates at the next-hop (merge point), the incoming label on the packet will be what the merge point expects.

As an example, LSR B can install a facility backup LSP for the link protected pop label 151. When the TE link B-C is up, LSR B will pop 151 and send the packet to C. If the TE link B-C is down, the LSR can pop 151 and send the packet via the facility backup to C.

## 8.2. Node Protection

The solutions for the PLR to provide node-protection for the pop and forward RSVP-TE tunnel will be explained in the next version of the document.

## 9. Quantifying pop labels

This section attempts to quantify the number of labels required in the forwarding plane to provide sharing of labels across RSVP-TE pop and forward tunnels. A MPLS RSVP-TE tunnel offers either no protection, link protection or node protection and only one of these labels is required per tunnel during signaling. The scale of the number of pop labels required per LSR can be deduced as follows:

- o For a LSR having X neighbors reachable across Y interfaces, the number of unprotected pop labels = X

- o For a PLR having X neighbors reachable across Y interfaces, number of link protected pop labels = X
- o For a PLR having X neighbors, each having Nx neighbors (i.e. next-nexthop for PLR), number of node protected pop labels = SUM\_OF\_ALL(Nx)

Total number of pop labels = Unprotected pop labels + link protected pop labels + node protected pop labels = 2X + SUM(Nm)

## 10. Protocol Extensions

### 10.1. Requirements

The functionality discussed in this document imposes the following requirements on the signaling protocol.

- o The Ingress of the LSP SHOULD have the ability to mandate/request the use and recording of pop labels at all hops along the path of the LSP.
- o When the use of pop labels is mandated/requested for the entire path,

the node recording the pop label SHOULD have the ability to indicate if the recorded label is a pop label.

the ingress SHOULD have the ability to override this path specific behavior by

explicitly mandating specific hops to not use pop labels (or)

mandating specific hops to share the onus of imposing the label stack (and also specifying the desired number of hops that need to be accounted for at that node)

the node which was mandated to share the onus of imposing the label stack SHOULD have the ability to indicate the actual number of hops that it can account for.

### 10.2. Attributes Flags TLV: Pop Label

Bit Number (TBD1): Pop Label

The presence of this in the LSP\_ATTRIBUTES/LSP\_REQUIRED\_ATTRIBUTES object of a Path message indicates that the ingress has requested/mandated the use and recording of pop labels at all hops along the

path of this LSP. When a node that does not cater to the request/mandate receives a Path message carrying the LSP\_REQUIRED\_ATTRIBUTES object with this flag set, it MUST send a PathErr message with an error code of 'routing problem' and an error value of 'pop label usage failure'.

### 10.3. RRO Label Subobject Flag: Pop Label

Bit Number (TBD2): Pop Label

The presence of this flag indicates that the recorded label is a pop label. This flag SHOULD be used by a node only if the use and recording of pop labels is requested/mandated for this LSP.

### 10.4. Attributes TLV: Label Stack Imposition TLV

| 0        |   |   |   |   |   |   |   |   |   | 1 |   |   |   |   |   |   |   |   |   | 2         |   |   |   |   |   |   |   |   |   | 3 |   |  |  |  |  |  |  |  |  |
|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|
| 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |  |  |  |  |  |  |  |  |
| Reserved |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   | Hop-Count |   |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |  |  |  |

Attribute TLV Type: TBD3

The presence of this in the HOP\_ATTRIBUTES subobject [RFC7570] of an ERO object in the Path message mandates the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject to share the onus of imposing the label stack. This attribute MUST be used only if the use and recording of pop labels is requested/mandated for this LSP (only if Pop Label flag is present in the LSP\_ATTRIBUTES/LSP\_REQUIRED\_ATTRIBUTES object). If the node is not able to comply with this mandate, it MUST send a PathErr message with an error code of 'routing problem' and an error value of 'label stack imposition failure'.

The Hop-Count field specifies the desired number of hops that this node needs to account for. A Hop-Count value of 0 is considered invalid and a value of 1 implies that this hop perform a normal swap or pop (if this hop is PHP) operation towards the next downstream hop.

The presence of this in the HOP\_ATTRIBUTES subobject of an RRO object in the RESV message indicates that the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject is sharing the onus of imposing the label stack. The Hop-Count field specifies the actual number of hops that this node can account for. This should not be included in the RESV message unless this TLV is also present in the corresponding Path message for this hop.

## 11. OAM considerations

Any extensions necessary for MPLS LSP traceroute for the RSVP-TE pop and forward tunnel will be explained in the next version of the document.

## 12. Acknowledgements

The authors would like to thank Adrian Farrel, Kireeti Kompella, Markus Jork and Ross Callon for their input from discussions.

## 13. Contributors

The following individuals contributed to this document:

Raveendra Torvi  
Juniper Networks  
Email: rtorvi@juniper.net

Chandra Ramachandran  
Juniper Networks  
Email: csekar@juniper.net

## 14. IANA Considerations

### 14.1. Attribute Flags: Pop Label

IANA manages the 'Attribute Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This document introduces a new Attribute Flag.

| Bit No. | Name      | Attribute Flags | Attribute Path | Attribute Resv | RRO | ERO | Reference                 |
|---------|-----------|-----------------|----------------|----------------|-----|-----|---------------------------|
| TBD1    | Pop Label | Yes             | No             | No             | No  | No  | This document (Section 5) |

### 14.2. Attribute TLV: Label Stack Imposition TLV

IANA manages the "Attribute TLV Space" registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This document introduces a new Attribute TLV.

| Type | Name                                | Allowed on<br>LSP<br>ATTRIBUTES | Allowed on<br>LSP REQUIRED<br>ATTRIBUTES | Allowed on<br>LSP Hop<br>Attributes | Reference                    |
|------|-------------------------------------|---------------------------------|------------------------------------------|-------------------------------------|------------------------------|
| TBD3 | Label<br>Stack<br>Imposition<br>TLV | No                              | No                                       | Yes                                 | This document<br>(Section 7) |

#### 14.3. Record Route Label Sub-object Flags: Pop Label

IANA manages the 'Record Route Object Sub-object Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This registry currently does not include Label Sub-object Flags. This document proposes the addition of a new sub-registry for Label Sub-object Flags as shown below.

| Flag | Name         | Reference                 |
|------|--------------|---------------------------|
| 0x1  | Global Label | RFC 3209                  |
| TBD2 | Pop Label    | This document (Section 5) |

#### 15. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

#### 16. References

##### 16.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<http://www.rfc-editor.org/info/rfc3209>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<http://www.rfc-editor.org/info/rfc4090>>.
- [RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <<http://www.rfc-editor.org/info/rfc7570>>.

## 16.2. Informative References

- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-11 (work in progress), February 2017.
- [I-D.ietf-teas-rsvp-te-scaling-rec]  
Beeram, V., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Implementation Recommendations to Improve the Scalability of RSVP-TE Deployments", draft-ietf-teas-rsvp-te-scaling-rec-03 (work in progress), October 2016.
- [I-D.sitaraman-sr-rsvp-coexistence-rec]  
Sitaraman, H., Beeram, V., Minei, I., and S. Sivabalan, "Recommendations for RSVP-TE and Segment Routing LSP co-existence", draft-sitaraman-sr-rsvp-coexistence-rec-02 (work in progress), February 2017.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<http://www.rfc-editor.org/info/rfc2961>>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<http://www.rfc-editor.org/info/rfc5920>>.

Authors' Addresses

Harish Sitaraman  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
US

Email: [hsitaraman@juniper.net](mailto:hsitaraman@juniper.net)

Vishnu Pavan Beeram  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
US

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Tejal Parikh  
Verizon  
400 International Parkway  
Richardson, TX 75081  
US

Email: [tejal.parikh@verizon.com](mailto:tejal.parikh@verizon.com)



MPLS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 12, 2018

H. Sitaraman  
V. Beeram  
Juniper Networks  
T. Parikh  
Verizon  
T. Saad  
Cisco Systems  
December 9, 2017

Signaling RSVP-TE tunnels on a shared MPLS forwarding plane  
draft-sitaraman-mpls-rsvp-shared-labels-03.txt

Abstract

As the scale of MPLS RSVP-TE networks has grown, so the number of Label Switched Paths (LSPs) supported by individual network elements has increased. Various implementation recommendations have been proposed to manage the resulting increase in control plane state.

However, those changes have had no effect on the number of labels that a transit Label Switching Router (LSR) has to support in the forwarding plane. That number is governed by the number of LSPs transiting or terminated at the LSR and is directly related to the total LSP state in the control plane.

This document defines a mechanism to prevent the maximum size of the label space limit on an LSR from being a constraint to control plane scaling on that node. That is, it allows many more LSPs to be supported than there are forwarding plane labels available.

This work introduces the notion of pre-installed 'per Traffic Engineering (TE) link labels' that can be shared by MPLS RSVP-TE LSPs that traverse these TE links. This approach significantly reduces the forwarding plane state required to support a large number of LSPs. This couples the feature benefits of the RSVP-TE control plane with the simplicity of the Segment Routing MPLS forwarding plane.

This document also introduces the ability to mix different types of label operations along the path of an LSP, thereby allowing the ingress router or an external controller to influence how to optimally place a LSP in the network.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

#### Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 12, 2018.

#### Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|                                                               |    |
|---------------------------------------------------------------|----|
| 1. Introduction . . . . .                                     | 3  |
| 2. Terminology . . . . .                                      | 5  |
| 3. Allocation of TE Link Labels . . . . .                     | 5  |
| 4. Segment Routed RSVP-TE Tunnel Setup . . . . .              | 5  |
| 5. Delegating Label Stack Imposition . . . . .                | 7  |
| 5.1. Stacking at the Ingress . . . . .                        | 8  |
| 5.1.1. Stack to Reach Delegation Hop . . . . .                | 8  |
| 5.1.2. Stack to Reach Egress . . . . .                        | 9  |
| 5.2. Explicit Delegation . . . . .                            | 10 |
| 5.3. Automatic Delegation . . . . .                           | 10 |
| 5.3.1. Effective Transport Label-Stack Depth (ETLD) . . . . . | 10 |

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| 6. Mixing TE Link Labels and Regular Labels in an RSVP-TE Tunnel           | 11 |
| 7. Construction of Label Stacks                                            | 12 |
| 8. Facility Backup Protection                                              | 13 |
| 8.1. Link Protection                                                       | 13 |
| 8.2. Node Protection                                                       | 14 |
| 9. Quantifying TE Link Labels                                              | 14 |
| 10. Protocol Extensions                                                    | 14 |
| 10.1. Requirements                                                         | 14 |
| 10.2. Attribute Flags TLV: TE Link Label                                   | 15 |
| 10.3. RRO Label Subobject Flag: TE Link Label                              | 15 |
| 10.4. Attribute Flags TLV: LSI-D                                           | 15 |
| 10.5. RRO Label Subobject Flag: Delegation Label                           | 16 |
| 10.6. Attributes Flags TLV: LSI-D-S2E                                      | 16 |
| 10.7. Attributes TLV: ETLD                                                 | 16 |
| 11. OAM Considerations                                                     | 17 |
| 12. Acknowledgements                                                       | 17 |
| 13. Contributors                                                           | 17 |
| 14. IANA Considerations                                                    | 18 |
| 14.1. Attribute Flags: TE Link Label, LSI-D, LSI-D-S2E                     | 18 |
| 14.2. Attribute TLV: ETLD                                                  | 18 |
| 14.3. Record Route Label Sub-object Flags: TE Link Label, Delegation Label | 18 |
| 15. Security Considerations                                                | 19 |
| 16. References                                                             | 19 |
| 16.1. Normative References                                                 | 19 |
| 16.2. Informative References                                               | 20 |
| Authors' Addresses                                                         | 21 |

## 1. Introduction

The scaling of RSVP-TE [RFC3209] control plane implementations can be improved by adopting the guidelines and mechanisms described in [RFC2961] and [I-D.ietf-teas-rsvp-te-scaling-rec]. These documents do not make any difference to the forwarding plane state required to handle the control plane state. The forwarding plane state remains unchanged and is directly proportional to the total number of Label Switching Paths (LSPs) supported by the control plane.

This document describes a mechanism that prevents the size of the platform specific label space on a Label Switching Router (LSR) from being a constraint to pushing the limits of control plane scaling on that node.

This work introduces the notion of pre-installed 'per Traffic Engineering (TE) link labels' that are allocated by an LSR. Each such label is installed in the MPLS forwarding plane with a 'pop' operation and the instruction to forward the received packet over the TE link. An LSR advertises this label in the Label object of a Resv

message as LSPs are set up and they are recorded hop by hop in the Record Route object (RRO) of the Resv message as it traverses the network. To make use of this feature, the ingress Label Edge Router (LER) pushes a stack of labels [RFC3031] as received in the RRO. These 'TE link labels' can be shared by MPLS RSVP-TE LSPs that traverse the same TE link.

This forwarding plane behavior fits in the MPLS architecture [RFC3031] and is same as that exhibited by Segment Routing (SR) [I-D.ietf-spring-segment-routing] when using an MPLS forwarding plane and a series of adjacency segments. This work couples the feature benefits of the RSVP-TE control plane with the simplicity of the Segment Routing MPLS forwarding plane. The RSVP-TE tunnels that use this shared forwarding plane can co-exist with MPLS-SR LSPs [I-D.ietf-spring-segment-routing-mpls] as described in [I-D.ietf-teas-sr-rsvp-coexistence-rec].

RSVP-TE using a shared MPLS forwarding plane offers the following benefits:

1. Shared Labels: The transit label on a TE link is shared among RSVP-TE tunnels traversing the link and is used independent of the ingress and egress of the LSPs.
2. Faster LSP setup time: No forwarding plane state needs to be programmed during LSP setup and teardown resulting in faster time for provisioning and deprovisioning LSPs.
3. Hitless re-routing: New transit labels are not required during make-before-break (MBB) in scenarios where the new LSP instance traverses the exact same path as the old LSP instance. This saves the ingress LER and the services that use the tunnel from needing to update the forwarding plane with new tunnel labels and so makes MBB events faster. Periodic MBB events are relatively common in networks that deploy the 'auto-bandwidth' feature on RSVP-TE LSPs to monitor bandwidth utilization and periodically adjust LSP bandwidth.
4. Mix and match labels: Both 'TE link labels' and regular labels can be used on transit hops for a single RSVP-TE tunnel (see Section 6). This allows backward compatibility with transit LSRs that provide regular labels in Resv messages.

No additional extensions are required to routing protocols (IGP-TE) in order to support this shared MPLS forwarding plane. Functionalities such as bandwidth admission control, LSP priorities, preemption, auto-bandwidth and Fast Reroute continue to work with this forwarding plane.

The signaling procedures and extensions discussed in this document do not apply to Point to Multipoint (P2MP) RSVP-TE Tunnels.

## 2. Terminology

The following terms are used in this document:

**TE link label:** An incoming label at an LSR that will be popped by the LSR with the packet being forwarded over a specific outgoing TE link to a neighbor.

**Shared MPLS forwarding plane:** An MPLS forwarding plane where every participating LSR uses TE link labels on every LSP.

**Segment Routed RSVP-TE tunnel:** An MPLS RSVP-TE tunnel that requests the use of a shared MPLS forwarding plane at every hop of the LSP.

## 3. Allocation of TE Link Labels

An LSR that participates in a shared MPLS forwarding plane **MUST** allocate a unique TE link label for each TE link. When an LSR encounters a TE link label at the top of the label stack it **MUST** pop the label and forward the packet over the TE link to the downstream neighbor on the RSVP-TE tunnel.

Multiple TE link labels **MAY** be allocated for the TE link to accommodate tunnels requesting no protection, link-protection and node-protection over the specific TE link.

Implementations that maintain per label bandwidth accounting at each hop must aggregate the reservations made for all the LSPs using the shared TE link label.

## 4. Segment Routed RSVP-TE Tunnel Setup

This section provides an example of how the RSVP-TE signaling procedure works to set up a tunnel utilizing a shared MPLS forwarding plane. The sample topology below is used to explain the example. Labels shown at each node are TE link labels that, when present at the top of the label stack, indicate that they should be popped and that the packet should be forwarded on the TE link to the neighbor.

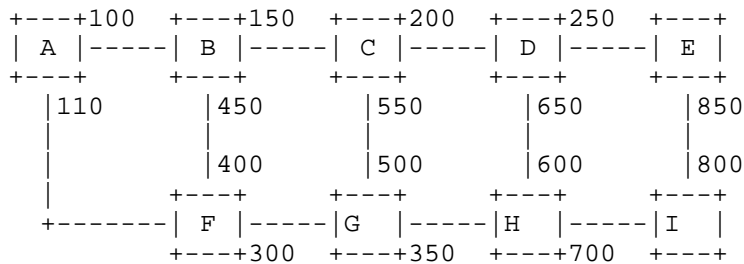


Figure 1: Sample Topology - TE Link Labels

Consider two tunnels:

RSVP-TE tunnel T1: From A to E on path A-B-C-D-E

RSVP-TE tunnel T2: From F to E on path F-B-C-D-E

Both tunnels share the TE links B-C, C-D, and D-E.

RSVP-TE is used to signal the setup of tunnel T1 (using the TE link label attributes flag defined in Section 10.2). When LSR D receives the Resv message from the egress LER E, it checks the next-hop TE link (D-E) and provides the TE link label (250) in the Resv message for the tunnel placing the label value in the Label object and also in the Label subobject carried in the RRO and setting the TE link label flag as defined in Section 10.3.

Similarly, LSR C provides the TE link label (200) for the TE link C-D, and LSR B provides the TE link label (150) for the TE link B-C.

For tunnel T2, the transit LSRs provide the same TE link labels as described for tunnel T1 as the links B-C, C-D, and D-E are common between the two LSPs.

The ingress LERs (A and F) will push the same stack of labels (from top of stack to bottom of stack) {150, 200, 250} for tunnels T1 and T2 respectively.

It should be noted that a transit LSR does not swap the top TE link label on an incoming packet (the label that it advertised in the Resv message it sent). All it has to do is pop the top label and forward the packet.

The values in the Label subobjects in the RRO are of interest to the ingress LERs in order to construct the stack of labels to impose on the packets.

If, in this example, there was another RSVP-TE tunnel T3 from F to I on path F-B-C-D-E-I, then this would also share the TE links B-C, C-D, and D-E and additionally traverse link E-I. The label stack used by F would be {150, 200, 250, 850}. Hence, regardless of the ingress and egress LERs from where the LSPs start and end, they will share LSR labels at shared hops in the shared MPLS forwarding plane.

There MAY be local operator policy at the ingress LER that influences the maximum depth of the label stack that can be pushed for a Segment Routed RSVP-TE tunnel. Prior to signaling the LSP, the ingress LER may decide that it would be unable to push a label stack containing one label for each hop along the path. In this case the LER can choose either to not signal a Segment Routed RSVP-TE tunnel (using normal LSP signaling instead), or can adopt the techniques described in Section 5 or Section 6.

## 5. Delegating Label Stack Imposition

One or more transit LSRs can assist the ingress LER by imposing part of the label stack required for the path. Consider the example in Figure 2 with an RSVP-TE tunnel from A to L on path A-B-C-D-E-F-G-H-I-J-K-L. In this case, the LSP is too long for LER A to impose the full label stack, so it uses the assistance of delegation hops LSR D and LSR I to impose parts of the label stack.

Each delegation hop allocates a delegation label to represent a set of labels that will be pushed at this hop. When a packet arrives at a delegation hop LSR with a delegation label, the LSR pops the label and pushes a set of labels before forwarding the packet.

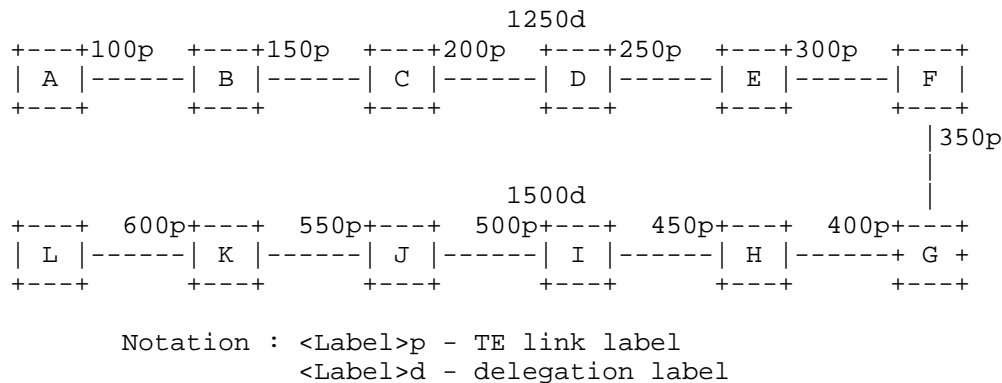


Figure 2: Delegating Label Stack Imposition

### 5.1. Stacking at the Ingress

When delegation labels come into play, there are two stacking approaches that the ingress can choose from. Section 7 explains how the label stack can be constructed.

#### 5.1.1. Stack to Reach Delegation Hop

In this approach, the stack pushed by the ingress carries a set of labels that will take the packet to the first delegation hop. When this approach is employed, the set of labels represented by a delegation label at a given delegation hop will include the corresponding delegation label from the next delegation hop. As a result, this delegation label can only be shared among LSPs that are destined to the same egress and traverse the same downstream path.

This approach is shown in Figure 3. The delegation label 1250 represents the stack {300, 350, 400, 450, 1500} and the delegation label 1500 represents the label stack {550, 600}.

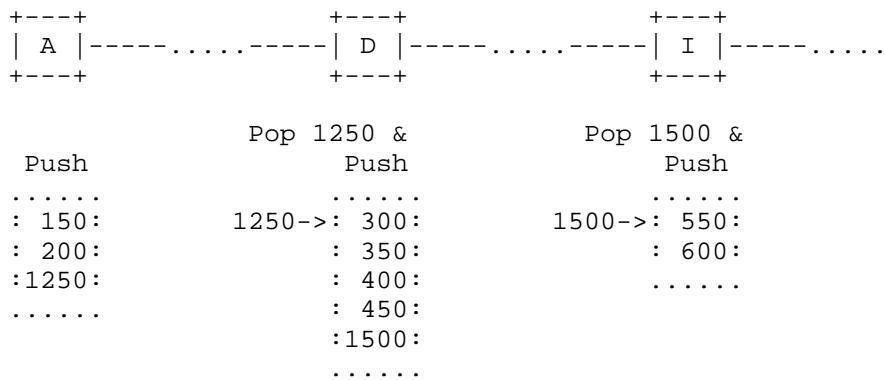


Figure 3: Stack to Reach Delegation Hop

With this approach, the ingress LER A will push {150, 200, 1250} for the tunnel in Figure 2. At LSR D, the delegation label 1250 will get popped and {300, 350, 400, 450, 1500} will get pushed. And at LSR I, the delegation label 1500 will get popped and the remaining set of labels {550, 600} will get pushed.



### 5.1.1.2. Stack to Reach Egress

In this approach, the stack pushed by the ingress carries a set of labels that will take the packet all the way to the egress so that all the delegation labels are part of the stack. When this approach is employed, the set of labels represented by a delegation label at a given delegation hop will not include the corresponding delegation label from the next delegation hop. As a result, this delegation label can be shared among all LSPs traversing the segment between the two delegation hops.

The downside of this approach is that the number of hops that the LSP can traverse is dictated by the label stack push limit of the ingress.

This approach is shown in Figure 4. The delegation label 1250 represents the stack {300, 350, 400, 450} and the delegation label 1500 represents the label stack {550, 600}.

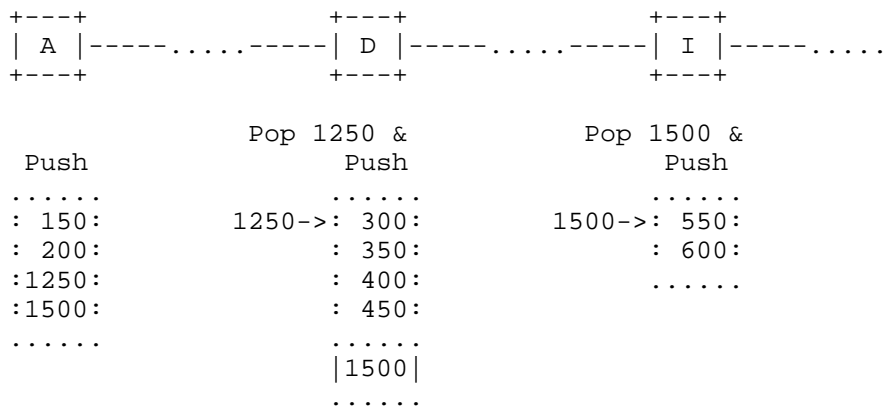


Figure 4: Stack to reach egress

With this approach, the ingress LER A will push {150, 200, 1250, 1500} for the tunnel in Figure 2. At LSR D, the delegation label 1250 will get popped and {300, 350, 400, 450} will get pushed. And at LSR I, the delegation label 1500 will get popped and the remaining set of labels {550, 600} will get pushed. The signaling extension required for the ingress to indicate the chosen stacking approach is defined in Section 10.6.

## 5.2. Explicit Delegation

In this delegation option, the ingress LER can explicitly delegate one or more specific transit LSRs to handle pushing labels for a certain number of their downstream hops. In order to accurately pick the delegation hops, the ingress needs to be aware of the label stack depth push limit of each of the transit LSRs prior to initiating the signaling sequence. The mechanism by which the ingress or controller (hosting the path computation element) learns this information is outside the scope of this document.

The signaling extension required for the ingress LER to explicitly delegate one or more specific transit hops is defined in Section 10.4. The extension required for the delegation hop to indicate that the recorded label is a delegation label is defined in Section 10.5.

## 5.3. Automatic Delegation

In this approach, the ingress LER lets the downstream LSRs automatically pick suitable delegation hops during the initial signaling sequence. The ingress does not need to be aware up front of the label stack depth push limit of each of the transit LSRs. The delegation hops are picked based on a per-hop signaled attribute called the Effective Transport Label-Stack Depth (ETLD) as described in the next section.

### 5.3.1. Effective Transport Label-Stack Depth (ETLD)

The ETLD is signaled as a per-hop attribute in the Path message [RFC7570]. When automatic delegation is requested, the ingress MUST populate the ETLD with the maximum number of transport labels that it can potentially send to its downstream hop. This value is then decremented at each successive hop. If a node is reached where the ETLD set from the previous hop is 1, then that node MUST select itself as the delegation hop. If a node is reached and it is determined that this hop cannot receive more than one transport label, then that node MUST select itself as the delegation hop. If there is a node or a sequence of nodes along the path of the LSP that do not support ETLD, then the immediate hop that supports ETLD MUST select itself as the delegation hop. The ETLD MUST be decremented at each non-delegation transit hop by either 1 or some appropriate number based on the limitations at that hop. At each delegation hop, the ETLD MUST be reset to the maximum number of transport labels that the hop can send and the ETLD decrements start again at each successive hop until either a new delegation hop is selected or the egress is reached. The net result is that by the time the Path message reaches the egress, all delegation hops are selected. During

the Resv processing, at each delegation hop, a suitable delegation label is selected (either an existing label is reused or a new label is allocated) and recorded in the Resv message.

Consider the example shown in Figure 5. Let's assume ingress LER A can push up to 3 transport labels while the remaining nodes can push up to 5 transport labels. The ingress LER A signals the initial Path message with ETLD set to 3. The ETLD value is adjusted at each successive hop and signaled downstream as shown. By the time the Path message reaches the egress LER L, LSRs D and I are automatically selected as delegation hops.

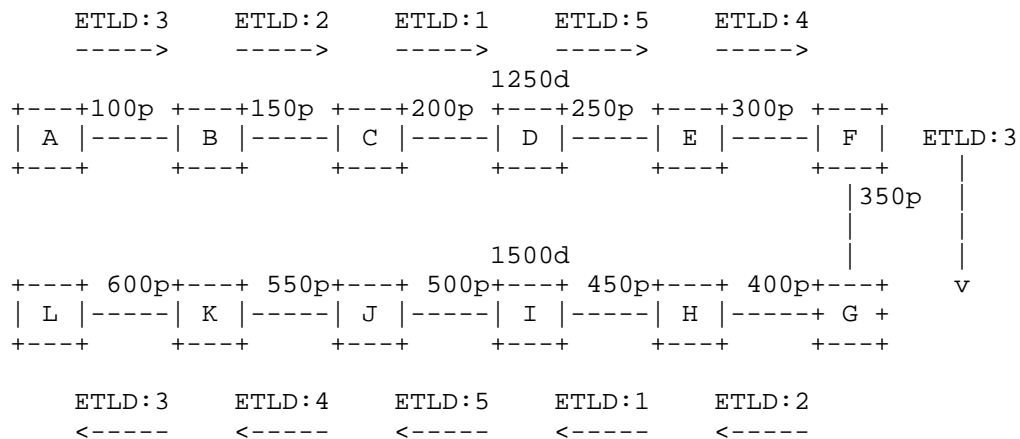
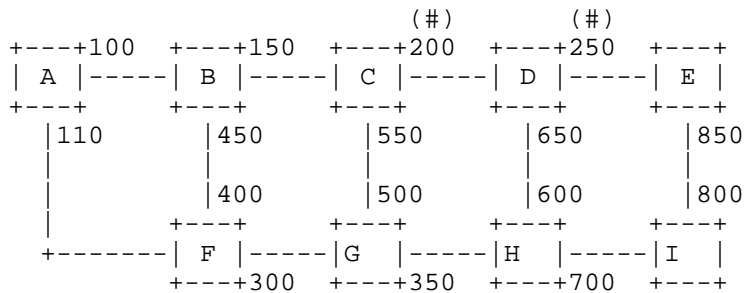


Figure 5: ETLD

Signaling extension for the ingress LER to request automatic delegation is defined in Section 10.4. The extension for signaling the ETLD is defined in Section 10.7. The extension required for the delegation hop to indicate that the recorded label is a delegation label is defined in Section 10.5.

## 6. Mixing TE Link Labels and Regular Labels in an RSVP-TE Tunnel

Labels can be mixed across transit hops in a single MPLS RSVP-TE LSP. Certain LSRs can use TE link labels and others can use regular labels. The ingress can construct a label stack appropriately based on what type of label is recorded from every transit LSR.



Notation : (#) denotes regular labels  
 Other labels are TE link labels

Figure 6: Sample Topology - TE Link Labels and Regular Labels

If the transit LSR allocates a regular label to be sent upstream in the Resv, then the label operation at the LSR is a swap to the label received from the downstream LSR. If the transit LSR is using a TE link label to be sent upstream in the Resv, then the label operation at the LSR is a pop and forward regardless of any label received from the downstream LSR. There is no change in the behavior of a penultimate hop popping (PHP) LSR [RFC3031].

Section 7 explains how the label stack can be constructed. For example, the LSP from A to I using path A-B-C-D-E-I will use a label stack of {150, 200}.

## 7. Construction of Label Stacks

The ingress LER or delegation hop MUST check the type of label received from each transit hop as recorded in the RRO in the Resv message and generate the appropriate label stack to reach the next delegation hop or the egress.

The following logic could be used by the node constructing the label stack:

Each RRO label sub-object SHOULD be processed starting with the label sub-object from the first downstream hop. Any label provided by the first downstream hop MUST always be pushed on the label stack regardless of the label type. If the label type is a TE link label, then any label from the next downstream hop MUST also be pushed on the constructed label stack. If the label type is a regular label, then any label from the next downstream hop MUST NOT be pushed on the constructed label stack. If the label type is a delegation label, then the stacking procedure stops at

that delegation hop. Approaches in Section 5.1 SHOULD be used to determine how the delegation labels are pushed in the label stack.

## 8. Facility Backup Protection

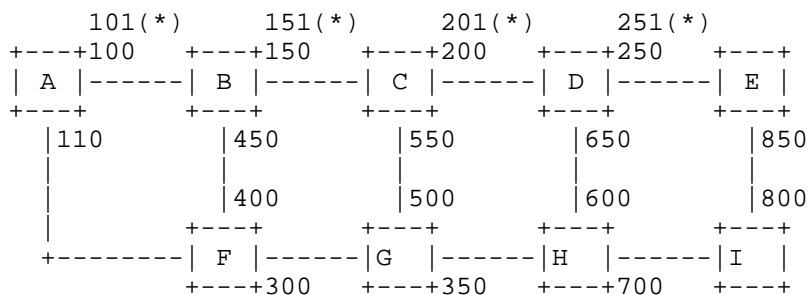
The following section describe how link and node protection works with facility backup protection [RFC4090] for the Segment Routed RSVP-TE tunnels.

### 8.1. Link Protection

To provide link protection at a Point of Local Repair (PLR) with a shared MPLS forwarding plane, the LSR SHOULD allocate a separate TE link label for the TE link that will be used for RSVP-TE tunnels that request link-protection from the ingress. No signaling extensions are required to support link protection for RSVP-TE tunnels over the shared MPLS forwarding plane.

At each LSR, link protected TE link labels can be allocated for each TE link and a link protecting facility backup LSP can be created to protect the TE link. The link protected TE link label can be sent by the LSR for LSPs requesting link-protection over the specific TE link. Since the facility backup terminates at the next-hop (merge point), the incoming label on the packet will be what the merge point expects.

Consider the network shown in Figure 7. LSR B can install a facility backup LSP for the link protected TE link label 151. When the TE link B-C is up, LSR B will pop 151 and send the packet to C. If the TE link B-C is down, the LSR can pop 151 and send the packet via the facility backup to C.



Notation : (\*) denotes link protection TE link labels

Figure 7: Link Protection Topology

## 8.2. Node Protection

The solutions for the PLR to provide node-protection for the Segment Routed RSVP-TE tunnel will be explained in a future version of this document.

## 9. Quantifying TE Link Labels

This section quantifies the number of labels required in the forwarding plane to provide sharing of labels across Segment Routed RSVP-TE tunnels. An MPLS RSVP-TE tunnel offers either no protection, link protection, or node protection and only one of these labels is required per tunnel during signaling. The scale of the number of TE link labels required per LSR can be deduced as follows:

- o For an LSR having X neighbors reachable across Y interfaces, the number of unprotected TE link labels is X.
- o For a PLR having X neighbors reachable across Y interfaces, the number of link protected TE link labels is X.
- o For a PLR having X neighbors, each having Nx neighbors (i.e. next-nexthops for the PLR), number of node protected TE link labels is SUM\_OF\_ALL(Nx).

The total number of TE link labels is given by:

Unprotected TE link labels +  
link protected TE link labels +  
node protected TE link labels =  $2X + \text{SUM\_OF\_ALL}(Nx)$

## 10. Protocol Extensions

### 10.1. Requirements

The functionality discussed in this document imposes the following requirements on the signaling protocol.

- o The Ingress of the LSP SHOULD have the ability to mandate/request the use and recording of TE link labels at all hops along the path of the LSP.
- o When the use of TE link labels is mandated/requested for the path:
  - \* the node recording the TE link label SHOULD have the ability to indicate if the recorded label is a TE link label.

- \* the ingress SHOULD have the ability to delegate label stack imposition by:
  - + explicitly mandating specific hops to be delegation hops (or)
  - + requesting automatic delegation.
- \* When explicit delegation is mandated or automatic delegation is requested:
  - + the ingress SHOULD have the ability to indicate the chosen stacking approach (and)
  - + the delegation hop SHOULD have the ability to indicate that the recorded label is a delegation label.

#### 10.2. Attribute Flags TLV: TE Link Label

Bit Number (TBD1): TE Link Label

The presence of this in the LSP\_ATTRIBUTES/LSP\_REQUIRED\_ATTRIBUTES object of a Path message indicates that the ingress has requested/mandated the use and recording of TE link labels at all hops along the path of this LSP. When a node that does not cater to the mandate receives a Path message carrying the LSP\_REQUIRED\_ATTRIBUTES object with this flag set, it MUST send a PathErr message with an error code of 'routing problem' and an error value of 'TE link label usage failure'.

#### 10.3. RRO Label Subobject Flag: TE Link Label

Bit Number (TBD2): TE Link Label

The presence of this flag indicates that the recorded label is a TE link label. This flag MUST be used by a node only if the use and recording of TE link labels is requested/mandated for the LSP.

#### 10.4. Attribute Flags TLV: LSI-D

Bit Number (TBD3): Label Stack Imposition - Delegation (LSI-D)

Automatic Delegation: The presence of this flag in the LSP\_ATTRIBUTES object of a Path message indicates that the ingress has requested automatic delegation of label stack imposition. This flag MUST be set in the LSP\_ATTRIBUTES object of a Path message only if the use and recording of TE link labels is requested/mandated for this LSP.

Explicit Delegation: The presence of this flag in the HOP\_ATTRIBUTES subobject [RFC7570] of an ERO object in the Path message indicates that the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject has been picked as an explicit delegation hop. The HOP\_ATTRIBUTES subobject carrying this flag MUST have the R (Required) bit set. This flag MUST be set in the HOP\_ATTRIBUTES subobject of an ERO object in the Path message only if the use and recording of TE link labels is requested/mandated for this LSP. If the hop is not able to comply with this mandate, it MUST send a PathErr message with an error code of 'routing problem' and an error value of 'label stack imposition failure'.

#### 10.5. RRO Label Subobject Flag: Delegation Label

Bit Number (TBD4): Delegation Label

The presence of this flag indicates that the recorded label is a delegation label. This flag MUST be used by a node only if the use and recording of TE link labels and delegation are requested/mandated for the LSP.

#### 10.6. Attributes Flags TLV: LSI-D-S2E

Bit Number (TBD5): Label Stack Imposition - Delegation - Stack to reach egress (LSI-D-S2E)

The presence of this flag in the LSP\_ATTRIBUTES object of a Path message indicates that the ingress has chosen to use the "Stack to reach egress" approach for stacking. The absence of this flag in the LSP\_ATTRIBUTES object of a Path message indicates that the ingress has chosen to use the "Stack to reach delegation hop" approach for stacking. This flag MUST be set in the LSP\_ATTRIBUTES object of a Path message only if the use and recording of TE link labels and delegation are requested/mandated for this LSP.

#### 10.7. Attributes TLV: ETLD

The format of the ETLD Attributes TLV is shown in Figure 8. The Attribute TLV Type is TBD6.



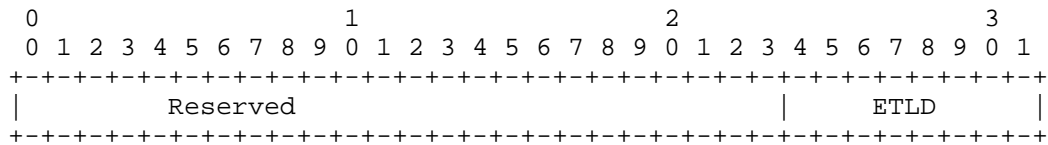


Figure 8: The ETLD Attributes TLV

The presence of this TLV in the HOP\_ATTRIBUTES subobject of an RRO object in the Path message indicates that the hop identified by the preceding IPv4 or IPv6 or Unnumbered Interface ID subobject supports automatic delegation. This attribute MUST be used only if the use and recording of TE link labels is requested/mandated and automatic delegation is requested for the LSP. The ETLID field specifies the maximum number of transport labels that this hop can potentially send to its downstream hop.

## 11. OAM Considerations

MPLS LSP ping and traceroute [RFC8029] are applicable for Segment Routed RSVP-TE tunnels. The existing procedures allow for the label stack imposed at a delegation hop to be reported back in the Label Stack Sub-TLV in the MPLS echo reply for traceroute.

## 12. Acknowledgements

The authors would like to thank Adrian Farrel, Kireeti Kompella, Markus Jork and Ross Callon for their input from discussions.

Adrian Farrel provided a review and text suggestion for clarity and readability.

### 13. Contributors

The following individuals contributed to this document:

Raveendra Torvi  
Juniper Networks  
Email: [rtorvi@juniper.net](mailto:rtorvi@juniper.net)

Chandra Ramachandran  
Juniper Networks  
Email: [csekar@juniper.net](mailto:csekar@juniper.net)

George Swallow  
Email: [swallow.ietf@gmail.com](mailto:swallow.ietf@gmail.com)

## 14. IANA Considerations

### 14.1. Attribute Flags: TE Link Label, LSI-D, LSI-D-S2E

IANA manages the 'Attribute Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This document introduces three new Attribute Flags.

| Bit No. | Name          | Attribute Flags | Attribute Path | Attribute Resv | RRO | ERO | Reference                    |
|---------|---------------|-----------------|----------------|----------------|-----|-----|------------------------------|
| TBD1    | TE Link Label | Yes             | No             | No             | No  | No  | This document (Section 11.2) |
| TBD3    | LSI-D         | Yes             | No             | No             | No  | Yes | This document (Section 11.4) |
| TBD5    | LSI-D-S2E     | Yes             | No             | No             | No  | No  | This document (Section 11.6) |

### 14.2. Attribute TLV: ETLD

IANA manages the "Attribute TLV Space" registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This document introduces a new Attribute TLV.

| Type | Name | Allowed on LSP ATTRIBUTES | Allowed on LSP REQUIRED ATTRIBUTES | Allowed on LSP Hop Attributes | Reference                    |
|------|------|---------------------------|------------------------------------|-------------------------------|------------------------------|
| TBD6 | ETLD | No                        | No                                 | Yes                           | This document (Section 11.7) |

### 14.3. Record Route Label Sub-object Flags: TE Link Label, Delegation Label

IANA manages the 'Record Route Object Sub-object Flags' registry as part of the 'Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters' registry located at <http://www.iana.org/assignments/rsvp-te-parameters>. This registry currently does not include Label Sub-object Flags. This document requests the addition of a new sub-registry for Label Sub-object Flags as shown below.

| Flag | Name             | Reference                    |
|------|------------------|------------------------------|
| 0x1  | Global Label     | RFC 3209                     |
| TBD2 | TE Link Label    | This document (Section 11.3) |
| TBD4 | Delegation Label | This document (Section 11.5) |

## 15. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

## 16. References

### 16.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<https://www.rfc-editor.org/info/rfc2205>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <<https://www.rfc-editor.org/info/rfc7570>>.

- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 16.2. Informative References

- [I-D.ietf-spring-segment-routing]  
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-13 (work in progress), October 2017.
- [I-D.ietf-spring-segment-routing-mpls]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-11 (work in progress), October 2017.
- [I-D.ietf-teas-rsvp-te-scaling-rec]  
Beeram, V., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP Traffic Engineering Deployments", draft-ietf-teas-rsvp-te-scaling-rec-08 (work in progress), October 2017.
- [I-D.ietf-teas-sr-rsvp-coexistence-rec]  
Sitaraman, H., Beeram, V., Minei, I., and S. Sivabalan, "Recommendations for RSVP-TE and Segment Routing LSP co-existence", draft-ietf-teas-sr-rsvp-coexistence-rec-01 (work in progress), June 2017.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <<https://www.rfc-editor.org/info/rfc2961>>.
- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<https://www.rfc-editor.org/info/rfc5920>>.

Authors' Addresses

Harish Sitaraman  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, CA 94089  
US

Email: [hsitaraman@juniper.net](mailto:hsitaraman@juniper.net)

Vishnu Pavan Beeram  
Juniper Networks  
10 Technology Park Drive  
Westford, MA 01886  
US

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Tejal Parikh  
Verizon  
400 International Parkway  
Richardson, TX 75081  
US

Email: [tejal.parikh@verizon.com](mailto:tejal.parikh@verizon.com)

Tarek Saad  
Cisco Systems  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
Canada

Email: [tsaad@cisco.com](mailto:tsaad@cisco.com)

CCAMP Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: September 7, 2017

J.E. Lopez de Vergara  
Daniel Perdices  
Universidad Autonoma de Madrid  
V. Lopez  
O. Gonzalez de Dios  
Telefonica I+D/GCTO  
D. King  
Lancaster University  
Y. Lee  
Huawei  
G. Galimberti  
Cisco Photonics Srl  
March 6, 2017

YANG data model for Flexi-Grid Optical Networks  
draft-vergara-ccamp-flexigrid-yang-04.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on September 7, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

## Abstract

This document defines a YANG model for managing flexi-grid optical Networks. The model described in this document is composed of two submodels: one to define a flexi-grid traffic engineering database, and other one to describe the flexi-grid paths or media channels. It is grounded on other defined YANG abstract models.

## Table of Contents

|                                                     |    |
|-----------------------------------------------------|----|
| 1. Introduction .....                               | 2  |
| 2. Conventions used in this document .....          | 3  |
| 3. Flexi-grid network topology model overview ..... | 3  |
| 4. Main building blocks.....                        | 4  |
| 4.1. flexi-grid TED .....                           | 4  |
| 4.2. Media-channel/network-media-channel .....      | 8  |
| 5. Example of use .....                             | 11 |
| 6. Formal Syntax .....                              | 12 |
| 7. Security Considerations .....                    | 12 |
| 8. IANA Considerations .....                        | 12 |
| 9. References .....                                 | 12 |
| 9.1. Normative References .....                     | 12 |
| 9.2. Informative References .....                   | 13 |
| 10. Contributors .....                              | 13 |
| 11. Acknowledgments .....                           | 14 |
| Appendix A. YANG models.....                        | 14 |
| A.1. Flexi-grid TED YANG Model .....                | 14 |
| A.1.1. YANG Model - Tree .....                      | 14 |
| A.1.2. YANG Model - Code .....                      | 16 |
| A.2. Media Channel YANG Model .....                 | 26 |
| A.2.1. YANG Model - Tree .....                      | 26 |
| A.2.2. YANG Model - Code .....                      | 27 |
| A.3. License .....                                  | 31 |
| Authors' Addresses .....                            | 32 |

## 1. Introduction

Internet-based traffic is dramatically increasing every year. Moreover, such traffic is also becoming more dynamic. Thus, transport networks need to evolve from current DWDM systems towards elastic optical networks, based on flexi-grid transmission and switching technologies. This technology aims at increasing both transport network scalability and flexibility, allowing the optimization of bandwidth usage.

This document presents a YANG model for flexi-grid objects in the dynamic optical network, including the nodes, transponders and links between them, as well as how such links interconnect nodes and transponders.

The YANG model for flexi-grid [RFC7698] networks allows the representation of the flexi-grid optical layer of a network, combined with the underlying physical layer. The model is defined in two YANG modules:

- o Flexi-grid-TED (Traffic Engineering Database): This module defines all the information needed to represent the flexi-grid optical node, transponder and link.
- o Media-channel: This module defines the whole path from a source transponder to the destination through a number of intermediate nodes in the flexi-grid optical network.

This document identifies the flexi-grid components, parameters and their values, characterizes the features and the performances of the flexi-grid elements. An application example is provided towards the end of the document to better understand their utility.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

## 3. Flexi-grid network topology model overview

YANG is a data modeling language used to model configuration data manipulated by the NETCONF protocol. Several YANG models have already been specified for network configurations. For instance, the work in [I-D.draft-ietf-ietf-i2rs-yang-network-topo] has proposed a generic YANG model for network/service topologies and inventories. The work in [I-D.draft-ietf-teas-yang-te-topo] presents a data model to represent, retrieve and manipulate Traffic Engineering (TE) Topologies. These models serve as base models that other technology specific models can augment. A YANG model has also been proposed in [I-D.draft-dharini-ccamp-dwdm-if-yang] to manage single channel optical interface parameters of DWDM applications, and in



Internet-Draft     A YANG data model for Flexi-Grid     March 2017  
[I-D.draft-ietf-ccamp-wson-yang] another model has been specified for the routing and wavelength assignment TE topology in wavelength switched optical networks (WSONs). None of them are specific for flexi-grid technology.

Then, as stated before, we propose a model to describe a flexi-grid topology that is split in two YANG sub-modules:

- o Flexi-grid-TED: In order to be compatible with existing proposals, we augment the definitions contained in [I-D.draft-ietf-i2rs-yang-network-topo] and [I-D.draft-ietf-teas-yang-te-topo], by defining the different elements we can find in a flexi-grid network: a node, a transponder and a link. For that, each of those elements is defined as a container that includes a group of attributes. References to the elements are provided to be later used in the definition of a media channel. It also includes the data types for the type of modulation, the flexi-grid technology, the FEC, etc.
- o Media-channel: This module defines the whole path from a source transponder to the destination through a number of intermediate nodes and links. For this, it takes the information defined before in the flexi-grid TED.

The following section provides a detailed view of each module.

#### 4. Main building blocks

Subsections below detail each of the defined YANG modules. They are listed in Appendix A.

##### 4.1. Flexi-grid TED

The description of the three main components, flexi-grid-node, flexi-grid-transponder and flexi-grid-link is provided below. flexi-grid-sliceable-transponders are also defined.

```
<flexi-grid-node> ::= <config> <state>
```

<flexi-grid-node>: This element designates a node in the network.

```
<config> ::= <flexi-grid-node-attributes-config>
```

<config>: Contains the configuration of a node.  
<flexi-grid-node-attributes-config> ::= <list-interface>  
<connectivity\_matrix>

<flexi-grid-node-attributes-config>: Contains all the attributes related to the node configuration, such as its interfaces or its management addresses.

```
<list-interface> ::= <name> <port-number>
<input-port> <output-port> <description>
<interface-type>
[<numbered-interface> / <unnumbered-interface>]
```

<list-interface>: The list containing all the information of the interfaces.

<name>: Determines the interface name.

<port-number>: Port number of the interface.

<input-port>: Boolean value that defines whether the interface is input or not.

<output-port>: Boolean value that defines whether the interface is output or not.

<description>: Description of the usage of the interface.

<interface-type>: Determines if the interface is numbered or unnumbered.

```
<numbered-interface> ::= <n-i-ip-address>
 <numbered-interface>: An interface with
 its own IP address.
```

```
 <n-i-ip-address>: Only available if
 <interface-type> is "numbered-interface".
 Determines the IP address of the interface.
```

```
<unnumbered-interface> ::= <u-i-ip-address>
<label>
```

```
 <unnumbered-interface>: A interface that
 needs a label to be unique.
```

```
 <u-i-ip-address>: Only available if
 <interface-type> is "numbered-interface".
 Determines the node IP address, which with
 the label defines the interface.
```

```
 <label>: Label that determines the
 interface, joint with the node IP address.
```

```
<connectivity-matrix> ::= <connections>
```

<connectivity-matrix>: Determines whether a connection port in/port out exists.

```
<connections> ::= <input-port-id>
<output-port-id>
```

<flexi-grid-transponder>: This item designates a transponder of a node.

<transponder-type>: Contains the type of the transponder.

<config> ::= <flexi-grid-transponder-attributes-config>

<config>: Contains the configuration of a transponder.

<flexi-grid-transponder-attributes-config> ::=  
<available-modulation> <modulation-type>  
<available-FEC> <FEC-enabled> [<FEC-type>]

<flexi-grid-transponder-attributes>: Contains all the attributes related to the transponder, such as whether it has FEC enabled or not, or its modulation type.

<available-modulation>: It provides a list of the modulations available at this transponder.

<modulation-type>: Determines the type of modulation in use: QPSK, QAM16, QAM64...

<available-FEC>: It provides a list of the FEC algorithms available at this transponder.

<FEC-enabled>: Boolean value that determines whether is the FEC enabled or not.

<FEC-type>: Determines the type of FEC in use: reed-solomon, hamming-code, enum golay, BCH...

<state> ::= <flexi-grid-transponder-attributes-config>  
<flexi-grid-transponder-attributes-state>

<state>: Contains the state of a transponder.

<flexi-grid-transponder-attributes-config>: See above.

<flexi-grid-transponder-attributes-state>: Contains the state of a transponder.

Internet-Draft    A YANG data model for Flexi-Grid    March 2017  
<flexi-grid-sliceable-transponder> ::= <transponder-type> <config>  
<state>

<flexi-grid-sliceable-transponder>: A list of transponders.

<transponder-type>: Contains the type of the transponder.

<config> ::= <flexi-grid-transponder-attributes-config>  
<flexi-grid-sliceable-transponder-attributes-config>

<flexi-grid-transponder-attributes-config>: See above.

<flexi-grid-sliceable-transponder-attributes-config> ::=  
<transponder-list>

<flexi-grid-sliceable-transponder-attributes-config>:  
Contains the configuration of a sliceable transponder

<transponder-list> ::= <carrier-id>  
<transponder-list>: A list of transponders.

<carrier-id>: An identifier for each one of the  
transponders in the list.

<state> ::= <flexi-grid-transponder-attributes-state>  
<flexi-grid-sliceable-transponder-attributes-state>  
<flexi-grid-transponder-attributes-config>  
<flexi-grid-sliceable-transponder-attributes-config>

<state>: Contains the state of a sliceable transponder.

<flexi-grid-transponder-attributes-state>: See above.

<flexi-grid-sliceable-transponder-attributes-state>:  
Contains the state attributes of a sliceable transponders.

<flexi-grid-transponder-attributes-config>: See above.

<flexi-grid-sliceable-transponder-attributes-config>: See  
above.

<link> ::= <config> <state>

<link>: This element describes all the information of a link.

<config> ::= <flexi-grid-link-attributes-config>

<config>: Contains the configuration of a link.

Internet-Draft    A YANG data model for Flexi-Grid    March 2017  
    <flexi-grid-link-attributes-config> ::= <technology-type>  
    <available-label-flexi-grid> <N-max> <base-frequency>  
    <nominal-central-frequency-granularity>  
    <slot-width-granularity>

    <flexi-grid-link-attributes>: Contains all the attributes related to the link, such as its unique id, its N value, its latency, etc.

    <link-id>: Unique id of the link.

    <available-label-flexi-grid>: Array of bits that determines, with each bit, the availability of each interface for flexi-grid technology.

    <N-max>: The max value of N in this link, being N the number of slots.

    <base-frequency>: The default central frequency used in the link.

    <nominal-central-frequency-granularity>: It is the spacing between allowed nominal central frequencies and it is set to 6.25 GHz (note: sometimes referred to as 0.00625 THz).

    <slot-width-granularity>: 12.5 GHz, as defined in G.694.1.

    <state> ::= <flexi-grid-link-attributes-config>  
    <flexi-grid-link-attributes-state>

    <state>: Contains the state of a link.

    <flexi-grid-link-attributes-config>: See above.

    <flexi-grid-link-attributes-state>: Contains all the the information related to the state of a link.

#### 4.2. Media-channel/network-media-channel

The model defines two types of media channels, following the terminology summarized in [RFC7698]:

- o media-channel, which represents a (effective) frequency slot supported by a concatenation of media elements (fibers, amplifiers, filters, switching matrices...);
- o network-media-channel: It is a media channel that transports an Optical Tributary Signal. In the model, the network media channel has as end-points transponders, which are the source and destination of the optical signal.

The description of these components is provided below:

Internet-Draft    A YANG data model for Flexi-Grid    March 2017

```

<media-channel> ::= <source> <destination> <link-channel> <effective-
freq-slot>
 <media-channel>: Determines a media-channel and its components.

 <source > ::= <source-node> <source-port>

 <source>: In a media-channel, the source is a node and a port.

 <source-node>: Reference to the source node of the media
channel.

 <source-port>: Reference to the source port in the source
<node>.

 <destination> ::= <destination-node> <destination-port>

 <destination>: In a media-channel, the destination is a node
and a port.

 <destination-node>: Reference to the destination node of the
media channel.

 <destination-port>: Reference to the destination port in the
destination node.

 <link-channel> ::= <link-id> <N> <M> <source-node> <source-port>
 <destination-node> <destination-port> <link> <bidirectional>
 <link-channel>: Defines a list with each of the links between
elements in the media channel.

 <link-id>: Unique identifier for the link channel

 <N>: N used for this link channel.

 <M>: M used for this link channel.

 <source-node>: Reference to the source node of this link
channel.

 <source-port>: Reference to the source port of this link
channel.

 <destination-node>: Reference to the destination node of this
link channel.

 <destination-port>: Reference to the destination port of this
link channel.

 <link>: Reference to the link of this link channel.

 <bidirectional>: Indicates if this link is bidirectional or
not.

```

<effective-freq-slot> ::= <N> <M>

<effective-freq-slot>: Defines the effective frequency slot of the media channel, which could be different from the one defined in the link channels.

<N>: Defines the effective N for this media channel.

<M>: Defines the effective M for this media channel.

<network-media-channel> ::= <source> <destination> <link-channel>  
<effective-freq-slot>

<network-media-channel>: Determines a network media-channel and its components.

<source > ::= <source-node> <source-transponder>

<source>: In a network media channel, the source is defined by a node and a transponder.

<source-node>: Reference to the source node of the media channel.

<source-transponder>: Reference to the source transponder in the source node.

<destination> ::= <destination-node> <destination-transponder>

<destination>: In a network media channel, the destination is defined by a node and a transponder

<destination-node>: Reference to the destination node of the media channel.

<destination-port>: Reference to the destination port in the destination node.

<link-channel>: See above, the information is reused for both types of media channels.

<effective-freq-slot>: See above, this information is reused for both types of media channels.

## 5. Example of use

In order to explain how this model is used, we provide the following example. An optical network usually has multiple transponders, switches (nodes) and links between them. Figure 1 shows a simple topology, where two physical paths interconnect two optical transponders.

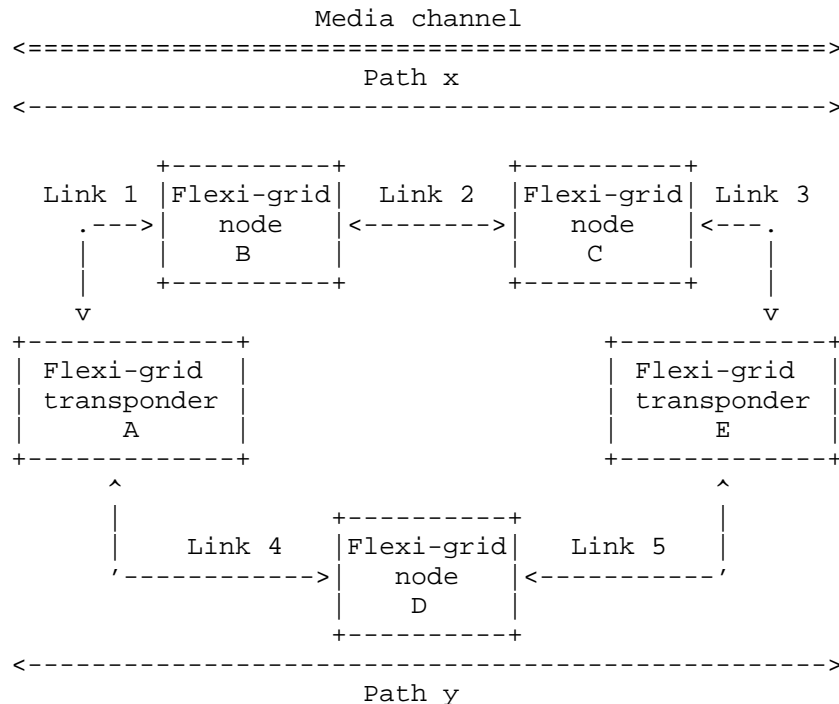


Figure 1. Topology example.

In order to configure a media channel to interconnect transponders A and E, first of all we have to populate the flexi-grid TED YANG model with all elements in the network:

1. We define the transponders A and E, including their FEC type, if enabled, and modulation type. We also provide node identifiers and addresses for the transponders, as well as interfaces included in the transponders. Sliceable transponders can also be defined if needed.
2. We do the same for the nodes B, C and D, providing their identifiers, addresses and interfaces, as well as the internal connectivity matrix between interfaces.
3. Then, we also define the links 1 to 5 that interconnect nodes and transponders, indicating which flexi-grid labels are available. Other information, such as the slot frequency and granularity are also provided.



Next, we can configure the media channel from the information we have stored in the flexi-grid TED, by querying which elements are available, and planning the resources that have to be provided on each situation. Note that every element in the flexi-grid TED has a reference, and this is the way in which they are called in the media channel.

4. Depending on the case, it is possible to define either the source and destination node ports, or the source and destination node and transponder. In our case, we would define a network media channel, with source transponder A and source node B, and destination transponder E and destination node C. Thus, we are going to follow path x.
5. Then, for each link in the path x, we indicate which channel we are going to use, providing information about the slots, and what nodes are connected.

Finally, the flexi-grid TED has to be updated with each element usage status each time a media channel is created or torn down.

## 6. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in [RFC5234].

## 7. Security Considerations

The transport protocol used for sending the managed information MUST support authentication and SHOULD support encryption.

The defined data-model by itself does not create any security implications.

## 8. IANA Considerations

The namespace used in the defined models is currently based on the IDEALIST project URI. Future versions of this document could register a URI in the IETF XML registry [RFC3688], as well as in the YANG Module Names registry [RFC6020].

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.

Internet-Draft    A YANG data model for Flexi-Grid            March 2017  
[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for  
the Network Configuration Protocol (NETCONF)", RFC 6020,  
October 2010.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,  
January 2004.

## 9.2. Informative References

- [RFC7698] Gonzalez de Dios, O., Casellas, R., Eds. "Framework and  
Requirements for GMPLS-Based Control of Flexi-Grid Dense  
Wavelength Division Multiplexing (DWDM) Networks",  
RFC7698, November 2015.
- [I-D.draft-ietf-i2rs-yang-network-topo] Clemm, A., Medved, J.,  
Varga, R., Bahadur, N., Ananthakrishnan, H., Liu, X.,  
"A Data Model for Network Topologies", Internet Draft  
draft-ietf-i2rs-yang-network-topo-12.txt, 2017.
- [I-D.draft-ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Pavan  
Beeram, V., Saad, T., Shah, H., Gonzalez De Dios, O., "YANG  
Data Model for TE Topologies", Internet Draft  
draft-ietf-teas-yang-te-topo-06.txt, 2016
- [I-D.draft-dharini-ccamp-dwdm-if-yang] Galimberti, G., Kunze, R.,  
Lam, K., Hiremagalur, D., Grammel, G., Fang, L., Ratterree, G.,  
Eds., "A YANG model to manage the optical interface parameters  
for an external transponder in a WDM network", Internet Draft,  
draft-dharini-ccamp-dwdm-if-param-yang-00.txt, 2016.
- [I-D.draft-ietf-ccamp-wson-yang] Lee, Y. Dhody, D., Zhang, X., Guo,  
A., Lopez, V., King, D., Yoon, B., "A Yang Data Model for WSON  
Optical Networks", Internet Draft,  
draft-ietf-ccamp-wson-yang-05.txt, 2017.

## 10. Contributors

The model presented in this paper was contributed to by more people  
than can be listed on the author list. Additional contributors  
include:

- o Zafar Ali, Cisco Systems
- o Daniel Michaud Vallinoto, Universidad Autonoma de Madrid

The work presented in this Internet-Draft has been partially funded by the European Commission under the project Industry-Driven Elastic and Adaptive Lambda Infrastructure for Service and Transport Networks (IDEALIST) of the Seventh Framework Program, with Grant Agreement Number: 317999, and by the Spanish Ministry of Economy and Competitiveness under the project TRAFICA (MINECO/FEDER TEC2015-69417-C2-1-R).

## Appendix A. YANG models

### A.1. Flexi-grid TED YANG Model

#### A.1.1. Yang Model - Tree Structure

```
module: ietf-flexi-grid-topology
flexi-grid-network-type
 augment /nd:networks/nd:network/nd:network-types:
 +--rw flexi-grid-network!
flexi-grid-link-attributes-config
 augment /nd:networks/nd:network/lnk:link/tet:te/tet:config:
 +--rw available-label-flexi-grid* bits
 +--rw N-max? int32
 +--rw base-frequency? decimal64
 +--rw nominal-central-frequency-granularity? decimal64
 +--rw slot-width-granularity? decimal64
flexi-grid-link-attributes-state
 augment /nd:networks/nd:network/lnk:link/tet:te/tet:state:
 +--ro available-label-flexi-grid* bits
 +--ro N-max? int32
 +--ro base-frequency? decimal64
 +--ro nominal-central-frequency-granularity? decimal64
 +--ro slot-width-granularity? decimal64
flexi-grid-node-attributes-config
 augment /nd:networks/nd:network/nd:node/tet:te/tet:config:
 +--rw interfaces* [name]
 +--rw name string
 +--rw port-number? uint32
 +--rw input-port? boolean
 +--rw output-port? boolean
 +--rw description? string
 +--rw type? interface-type
 +--rw numbered-interface
 | +--rw n-i-ip-address? inet:ip-address
 +--rw unnumbered-interface
 | +--rw u-i-ip-address? inet:ip-address
 +--rw label? uint32
```

```

flexi-grid-node-attributes-state
 augment /nd:networks/nd:network/nd:node/tet:te/tet:state:
 +--ro interfaces* [name]
 +--ro name string
 +--ro port-number? uint32
 +--ro input-port? boolean
 +--ro output-port? boolean
 +--ro description? string
 +--ro type? interface-type
 +--ro numbered-interface
 | +--ro n-i-ip-address? inet:ip-address
 +--ro unnumbered-interface
 +--ro u-i-ip-address? inet:ip-address
 +--ro label? uint32
flexi-grid-connectivity-matrix-attributes
 augment /nd:networks/nd:network/nd:node/tet:te/tet:config/
 tet:te-node-attributes/tet:connectivity-matrix:
 +--rw connections* [input-port-id]
 +--rw input-port-id flexi-grid-node-port-ref
 +--rw output-port-id? flexi-grid-node-port-ref
flexi-grid-connectivity-matrix-attributes
 augment /nd:networks/nd:network/nd:node/tet:te/tet:state/
 tet:te-node-attributes/tet:connectivity-matrix:
 +--ro connections* [input-port-id]
 +--ro input-port-id flexi-grid-node-port-ref
 +--ro output-port-id? flexi-grid-node-port-ref
flexi-grid-transponder
 augment /nd:networks/nd:network/nd:node/tet:te/
 tet:tunnel-termination-point:
 +--rw transponder-type flexi-grid-transponder-type
 +--rw config
 | +--rw available-modulation* modulation
 | +--rw modulation-type? modulation
 | +--rw available-FEC* FEC
 | +--rw FEC-enabled? boolean
 | +--rw FEC-type? FEC
 +--ro state
 +--ro available-modulation* modulation
 +--ro modulation-type? modulation
 +--ro available-FEC* FEC
 +--ro FEC-enabled? boolean
 +--ro FEC-type? FEC

```

```
augment /nd:networks/nd:network/nd:node/tet:te/
tet:tunnel-termination-point:
 +--rw transponder-type flexi-grid-transponder-type
 +--rw config
 | +--rw available-modulation* modulation
 | +--rw modulation-type? modulation
 | +--rw available-FEC* FEC
 | +--rw FEC-enabled? boolean
 | +--rw FEC-type? FEC
 | +--rw transponder-list* [carrier-id]
 | +--rw carrier-id uint32
 +--ro state
 | +--ro available-modulation* modulation
 | +--ro modulation-type? modulation
 | +--ro available-FEC* FEC
 | +--ro FEC-enabled? boolean
 | +--ro FEC-type? FEC
 | +--ro transponder-list* [carrier-id]
 | +--ro carrier-id uint32
```

#### A.1.2. YANG Model - Code

```
<CODE BEGINS> file "ietf-flexi-grid-ted.yang"
module ietf-flexi-grid-ted {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-flexi-grid-ted";
 prefix "fg-ted";

 import ietf-network {
 prefix "nd";
 }

 import ietf-network-topology {
 prefix "lnk";
 }

 import ietf-te-topology {
 prefix "tet";
 }

 import ietf-inet-types {
 prefix "inet";
 }

 organization
 "IETF CCAMP Working Group";

 contact
 "Editor: Jorge E. Lopez de Vergara
 <jorge.lopez_vergara@uam.es>";
```

description

"This module contains a collection of YANG definitions for a Flexi-Grid Traffic Engineering Database (TED)."

Copyright (c) 2017 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>).";

revision 2017-03-01 {

description

"version 4.";

reference

"RFC XXX: A Yang Data Model for Flexi-Grid Optical Networks ";

}

typedef flexi-grid-trasponder-type {

type enumeration {

enum "flexi-grid-transponder" {

description

"Flexi-grid transponder";

}

enum "flexi-grid-sliceable-transponder" {

description

"Flexi-grid sliceable transponder";

}

}

description "Determines the trasponder type:

flexi-grid-transponder or

flexi-grid-sliceable-transponder";

}

typedef modulation {

type enumeration {

enum QPSK {

description

"QPSK (Quadrature Phase Shift Keying) modulation";

}

enum DP\_QPSK {

description "DP-QPSK (Dual Polarization Quadrature Phase Shift Keying) modulation";

}

enum QAM16 {

description "QAM16 (Quadrature Amplitude Modulation - 4 bits per symbol) modulation";

}

```
 enum DP_QAM16 {
 description "DP-QAM16 (Dual Polarization
 Quadrature Amplitude Modulation - 4 bits per
 symbol) modulation";
 }
 enum DC_DP_QAM16 {
 description "DC DP-QAM16 (Dual Polarization
 Quadrature Amplitude Modulation - 4 bits per
 symbol) modulation";
 }
}
description
 "Enumeration that defines the type of wave modulation";
}

typedef FEC {
 type enumeration {
 enum reed-solomon {
 description "Reed-Solomon error correction";
 }
 enum hamming-code {
 description "Hamming Code error correction";
 }
 enum golay {
 description "Golay error correction";
 }
 }
 description "Enumeration that defines the type of
 Forward Error Correction";
}

typedef interface-type {
 type enumeration {
 enum numbered-interface {
 description "The interface is numbered";
 }
 enum unnumbered-interface {
 description "The interface is unnumbered";
 }
 }
 description
 "Enumeration that defines if an interface is numbered or
 unnumbered";
}
```

```
/*
 * Typedef related to references
 */
typedef flexi-grid-link-ref {
 type leafref {
 path
 "/nd:networks/nd:network/lnk:link/lnk:link-id";
 }

 description
 "This type is used by data models that need to reference
 a flexi-grid optical link.";
}

typedef flexi-grid-node-port-ref {
 type leafref {
 path "/nd:networks/nd:network/nd:node/tet:te/tet:config/"
 +"fg-ted:interfaces/fg-ted:port-number";
 }
 description
 "This type is used by data models that need to reference
 a flexi-grid port.";
}

typedef flexi-grid-transponder-ref {
 type leafref {
 path "/nd:networks/nd:network/nd:node/tet:te/"
 +"tet:tunnel-termination-point/tet:tunnel-tp-id";
 }
 description
 "This type is used by data models that need to reference
 a trasponder.";
}

grouping flexi-grid-network-type {
 container flexi-grid-network {
 presence "indicates a flexi-grid optical network";
 description "flexi-grid optical network";
 }
 description "If present, it indicates a flexi-grid
 optical TED network";
}

grouping flexi-grid-node-attributes-config {
 description "Set of attributes of an optical node.";

 list interfaces {
 key "name";
 unique "port-number";
 description "List of interfaces contained in the node";
 leaf name {
 type string;
 description "Interface name";
 }
 }
}
```



```

 leaf port-number {
 type uint32;
 description "Number of the port used by the interface";
 }
 leaf input-port {
 type boolean;
 description "Determines if the port is an input port";
 }
 leaf output-port {
 type boolean;
 description
 "Determines if the port is an output port";
 }
 leaf description {
 type string;
 description "Description of the interface";
 }
 leaf type {
 type interface-type;
 description "Determines the type of the interface";
 }
 container numbered-interface {
 when "../fg-ted:type =
 'numbered-interface'" {
 description
 "If the interface is a numbered interface";
 }
 description "Container that defines an numbered
 interface with an ip-address";
 leaf n-i-ip-address{
 type inet:ip-address;
 description "IP address of the numbered interface";
 }
 }
 container unnumbered-interface {
 when "../fg-ted:type =
 'unnumbered-interface'" {
 description
 "If the interface is an unnumbered interface";
 }
 description "Container that defines an unnumbered
 interface with an ip-address and a label";
 leaf u-i-ip-address{
 type inet:ip-address;
 description "IP address of the interface";
 }
 leaf label {
 type uint32;
 description "Number as label for the interface";
 }
 }
}

```

```

Internet-Draft A YANG data model for Flexi-Grid March 2017
 grouping flexi-grid-node-attributes-state {
 description "Flexigrid node attributes (state).";
 }
 grouping flexi-grid-link-attributes-config {
 description "Set of attributes of an optical link";
 leaf-list available-label-flexi-grid {
 type bits {
 bit is-available{
 description "Set to 1 when it is available";
 }
 }
 }
 description
 "Array of bits that determines whether a spectral
 slot is available or not.";
 }

 leaf N-max {
 type int32;
 description "Maximum number of channels available.";
 }

 leaf base-frequency {
 type decimal64 {
 fraction-digits 5;
 }
 units THz;
 default 193.1;
 description "Default central frequency";
 reference "rfc7698";
 }
 leaf nominal-central-frequency-granularity {
 type decimal64 {
 fraction-digits 5;
 }
 units GHz;
 default 6.25;
 description
 "It is the spacing between allowed nominal central
 frequencies and it is set to 6.25 GHz";
 reference "rfc7698";
 }

 leaf slot-width-granularity {
 type decimal64 {
 fraction-digits 5;
 }
 units GHz;
 description "Minimum space between slot widths";
 reference "rfc7698";
 }
}

```

```

 grouping flexi-grid-link-attributes-state {
 description "Flexigrid link attributes (state)";
 }
 grouping flexi-grid-transponder-attributes-config {
 description "Configuration of an optical transponder";
 leaf-list available-modulation {
 type modulation;
 description
 "List determining all the available modulations";
 }
 leaf modulation-type {
 type modulation;
 description "Modulation type of the wave";
 }
 leaf-list available-FEC {
 type FEC;
 description "List determining all the available FEC";
 }
 leaf FEC-enabled {
 type boolean;
 description
 "Determines whether the FEC is enabled or not";
 }
 leaf FEC-type {
 type FEC;
 description "FEC type of the transponder";
 }
 }

 grouping flexi-grid-transponder-attributes-state {
 description "State of an optical transponder";
 }

 grouping flexi-grid-sliceable-transponder-attributes-config {
 description
 "Configuration of a sliceable transponder.";
 list transponder-list {
 key "carrier-id";
 description "List of carriers";
 leaf carrier-id {
 type uint32;
 description "Identifier of the carrier";
 }
 }
 }

 grouping flexi-grid-sliceable-transponder-attributes-state {
 description "State of a sliceable transponder.";
 uses flexi-grid-transponder-attributes-state;
 }

```

```

 grouping flexi-grid-connectivity-matrix-attributes {
 description "Connectivity matrix between the input and
 output ports";
 list connections {
 key "input-port-id";
 leaf input-port-id {
 type flexi-grid-node-port-ref;
 description "Identifier of the input port";
 }
 leaf output-port-id {
 type flexi-grid-node-port-ref;
 description "Identifier of the output port";
 }
 }
 description "List of connections between input and
 output ports";
 }
 }

 augment "/nd:networks/nd:network/nd:network-types" {
 uses flexi-grid-network-type;
 description "Augment network-types including flexi-grid
 topology";
 }

 augment "/nd:networks/nd:network/lnk:link/tet:te/tet:config" {
 when "/nd:networks/nd:network/nd:network-types/
 fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 description "Augment link configuration";
 uses flexi-grid-link-attributes-config;
 }

 augment "/nd:networks/nd:network/lnk:link/tet:te/tet:state" {
 when "/nd:networks/nd:network/nd:network-types/
 fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 description "Augment link state";
 uses flexi-grid-link-attributes-config;
 uses flexi-grid-link-attributes-state;
 }

 augment "/nd:networks/nd:network/nd:node/tet:te/tet:config" {
 when "/nd:networks/nd:network/nd:network-types/
 fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 uses flexi-grid-node-attributes-config;
 description "Augment node config with flexi-grid attributes";
 }
}

```

```

Internet-Draft A YANG data model for Flexi-Grid March 2017
augment "/nd:networks/nd:network/nd:node/tet:te/tet:state" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 uses flexi-grid-node-attributes-config;
 uses flexi-grid-node-attributes-state;
 description "Augment node config with flexi-grid attributes";
}

augment "/nd:networks/nd:network/nd:node/tet:te/tet:config"+
"/tet:te-node-attributes/tet:connectivity-matrix" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-connectivity-matrix-attributes;
 description "Augment node connectivity-matrix for node config";
}

augment "/nd:networks/nd:network/nd:node/tet:te/tet:state"+
"/tet:te-node-attributes/tet:connectivity-matrix" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-connectivity-matrix-attributes;
 description "Augment node connectivity-matrix for node config";
}

augment "/nd:networks/nd:network/nd:node/tet:te"+
"/tet:tunnel-termination-point" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 leaf transponder-type {
 type flexi-grid-transponder-type;
 description "Type of flexi-grid transponder";
 }
 container state {
 description "State of the transponder";
 }
 container config {
 description "Configuration of the transponder";
 }
 description "Augment node with configuration and state
for transponder";
}

```

```

 augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:tunnel-termination-point/fg-ted:config" {
 when "../fg-ted:transponder-type" {
 description "When it is either a flexi-grid transponder
 or a sliceable transponder";
 }
 uses flexi-grid-transponder-attributes-config;
 description "Augment node state with transponder attributes";
 }

 augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:tunnel-termination-point/fg-ted:state" {
 when "../fg-ted:transponder-type"{
 description "When it is either a flexi-grid transponder
 or a sliceable transponder";
 }
 uses flexi-grid-transponder-attributes-state;
 uses flexi-grid-transponder-attributes-config;
 description "Augment node state with transponder attributes";
 }

 augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:tunnel-termination-point/fg-ted:config" {
 when "../fg-ted:transponder-type =
 'flexi-grid-sliceable-transponder'"{
 description
 "When it is a flexi-grid sliceable transponder";
 }
 uses flexi-grid-sliceable-transponder-attributes-config;
 description "Augment node with sliceable transponder
 attributes";
 }

 augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:tunnel-termination-point/fg-ted:state" {
 when "../fg-ted:transponder-type =
 'flexi-grid-sliceable-transponder'"{
 description
 "When it is a flexi-grid sliceable transponder";
 }
 uses flexi-grid-sliceable-transponder-attributes-state;
 uses flexi-grid-sliceable-transponder-attributes-config;
 description "Augment node with sliceable transponder
 attributes";
 }
 }

<CODE ENDS>

```

A.2.1. YANG Model - Tree

```
module: ietf-flexi-grid-media-channel
 +--rw media-channel
 | +--rw source
 | | +--rw source-node? te-types:te-node-id
 | | +--rw source-port? fg-ted:flexi-grid-node-port-ref
 | +--rw destination
 | | +--rw destination-node? te-types:te-node-id
 | | +--rw destination-port? fg-ted:flexi-grid-node-port-ref
 | +--rw effective-freq-slot
 | | +--rw N? int32
 | | +--rw M? int32
 | +--rw link-channel* [link-id]
 | | +--rw link-id int32
 | | +--rw N? int32
 | | +--rw M? int32
 | | +--rw source-node? te-types:te-node-id
 | | +--rw source-port? fg-ted:flexi-grid-node-port-ref
 | | +--rw destination-node? te-types:te-node-id
 | | +--rw destination-port? fg-ted:flexi-grid-node-port-ref
 | | +--rw link? fg-ted:flexi-grid-link-ref
 | | +--rw bidireccional? boolean
 | +--rw network-media-channel
 | | +--rw source
 | | | +--rw source-node? te-types:te-node-id
 | | | +--rw source-transponder? fg-ted:flexi-grid-transponder-ref
 | | +--rw destination
 | | | +--rw destination-node? te-types:te-node-id
 | | | +--rw destination-transponder? fg-ted:flexi-grid-transponder-ref
 | | +--rw effective-freq-slot
 | | | +--rw N? int32
 | | | +--rw M? int32
 | | +--rw link-channel* [link-id]
 | | | +--rw link-id int32
 | | | +--rw N? int32
 | | | +--rw M? int32
 | | | +--rw source-node? te-types:te-node-id
 | | | +--rw source-port? fg-ted:flexi-grid-node-port-ref
 | | | +--rw destination-node? te-types:te-node-id
 | | | +--rw destination-port? fg-ted:flexi-grid-node-port-ref
 | | | +--rw link? fg-ted:flexi-grid-link-ref
 | | | +--rw bidireccional? boolean
```

```
<CODE BEGINS> file "ietf-flexi-grid-media-channel.yang"
module ietf-flexi-grid-media-channel {
 yang-version 1.1;

 namespace
 "urn:ietf:params:xml:ns:yang:ietf-flexi-grid-media-channel";
 prefix "fg-mc";

 import ietf-flexi-grid-ted {
 prefix "fg-ted";
 }

 import ietf-te-types {
 prefix "te-types";
 }

 organization
 "IETF CCAMP Working Group";

 contact
 "Editor: Jorge E. Lopez de Vergara
 <jorge.lopez_vergara@uam.es>";

 description
 "This module contains a collection of YANG definitions for
 a Flexi-Grid media channel.

 Copyright (c) 2017 IETF Trust and the persons identified as
 authors of the code. All rights reserved.

 Redistribution and use in source and binary forms, with or
 without modification, is permitted pursuant to, and subject
 to the license terms contained in, the Simplified BSD
 License set forth in Section 4.c of the IETF Trust's Legal
 Provisions Relating to IETF Documents
 (http://trustee.ietf.org/license-info).";

 revision 2017-03-01 {
 description
 "version 4.";

 reference
 "RFC XXX: A Yang Data Model for Flexi-Grid Optical
 Networks ";
 }
}
```



```
container media-channel {
 description
 "Media association that represents both the topology
 (i.e., path through the media) and the resource
 (frequency slot) that it occupies. As a topological
 construct, it represents a (effective) frequency slot
 supported by a concatenation of media elements (fibers,
 amplifiers, filters, switching matrices...). This term
 is used to identify the end-to-end physical layer entity
 with its corresponding (one or more) frequency slots
 local at each link filters.";
 reference "rfc7698";
 container source {
 description "Source of the media channel";
 leaf source-node {
 type te-types:te-node-id;
 description "Source node";
 }
 leaf source-port {
 type fg-ted:flexi-grid-node-port-ref;
 description "Source port";
 }
 }
 container destination {
 description "Destination of the media channel";
 leaf destination-node {
 type te-types:te-node-id;
 description "Destination node";
 }
 leaf destination-port {
 type fg-ted:flexi-grid-node-port-ref;
 description "Destination port";
 }
 }
 uses media-channel-attributes;
}

container network-media-channel {
 description
 "It is a media channel that transports an Optical
 Tributary Signal ";
 reference "rfc7698";
 container source {
 description "Source of the network media channel";
 leaf source-node {
 type te-types:te-node-id;
 description "Source node";
 }
 leaf source-transponder {
 type fg-ted:flexi-grid-transponder-ref;
 description "Source transponder";
 }
 }
}
```

```

 container destination {
 description "Destination of the network media channel";
 leaf destination-node {
 type te-types:te-node-id;
 description "Destination node";
 }
 leaf destination-transponder {
 type fg-ted:flexi-grid-transponder-ref;
 description "Destination transponder";
 }
 }
 }
 uses media-channel-attributes;
}

grouping media-channel-attributes {
 description "Set of attributes of a media channel";
 container effective-freq-slot {
 description "The effective frequency slot is an attribute
 of a media channel and, being a frequency slot, it is
 described by its nominal central frequency and slot
 width";
 reference "rfc7698";
 leaf N {
 type int32;
 description
 "Is used to determine the Nominal Central
 Frequency. The set of nominal central frequencies
 can be built using the following expression:
 f = 193.1 THz + n x 0.00625 THz,
 where 193.1 THz is ITU-T ''anchor frequency'' for
 transmission over the C band, n is a positive or
 negative integer including 0.";
 reference "rfc7698";
 }
 leaf M {
 type int32;
 description
 "Is used to determine the slot width. A slot width
 is constrained to be M x SWG (that is, M x 12.5 GHz),
 where M is an integer greater than or equal to 1.";
 reference "rfc7698";
 }
 }
}
list link-channel {
 key "link-id";
 description
 "A list of the concatenated elements of the media
 channel.";
 leaf link-id {
 type int32;
 description "Identifier of the link";
 }
}

```

```
 uses link-channel-attributes;
 }
}

grouping link-channel-attributes {
 description
 "A link channel is one of the concatenated elements of
 the media channel.";
 leaf N {
 type int32;
 description
 "Is used to determine the Nominal Central Frequency.
 The set of nominal central frequencies can be built
 using the following expression:
 $f = 193.1 \text{ THz} + n \times 0.00625 \text{ THz}$,
 where 193.1 THz is ITU-T ''anchor frequency'' for
 transmission over the C band, n is a positive or
 negative integer including 0.";
 reference "rfc7698";
 }
 leaf M {
 type int32;
 description
 "Is used to determine the slot width. A slot
 width is constrained to be M x SWG (that is,
 M x 12.5 GHz), where M is an integer greater than
 or equal to 1.";
 reference "rfc7698";
 }
 leaf source-node {
 type te-types:te-node-id;
 description "Source node of the link channel";
 }
 leaf source-port {
 type fg-ted:flexi-grid-node-port-ref;
 description "Source port of the link channel";
 }
 leaf destination-node {
 type te-types:te-node-id;
 description "Destination node of the link channel";
 }
 leaf destination-port {
 type fg-ted:flexi-grid-node-port-ref;
 description "Destination port of the link channel";
 }
 leaf link {
 type fg-ted:flexi-grid-link-ref;
 description "Link of the link channel";
 }
}
```

```
 leaf bidireccional {
 type boolean;
 description
 "Determines whether the link is bidireccional or
 not";
 }
 }
}
```

<CODE ENDS>

### A.3. License

Copyright (c) 2017 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Jorge E. Lopez de Vergara  
Universidad Autonoma de Madrid  
Escuela Politecnica Superior  
C/Francisco Tomas y Valiente, 11  
E-28049 Madrid, Spain

Email: [jorge.lopez\\_vergara@uam.es](mailto:jorge.lopez_vergara@uam.es)

Daniel Perdices Burrero  
Universidad Autonoma de Madrid  
Escuela Politecnica Superior  
C/Francisco Tomas y Valiente, 11  
E-28049 Madrid, Spain

Email: [daniel.perdices@estudiante.uam.es](mailto:daniel.perdices@estudiante.uam.es)

Victor Lopez  
Telefonica I+D/GCTO  
Distrito Telefonica  
E-28050 Madrid, Spain

Email: [victor.lopezalvarez@telefonica.com](mailto:victor.lopezalvarez@telefonica.com)

Oscar Gonzalez de Dios  
Telefonica I+D/GCTO  
Distrito Telefonica  
E-28050 Madrid, Spain

Email: [oscar.gonzalezdedios@telefonica.com](mailto:oscar.gonzalezdedios@telefonica.com)

Daniel King  
Lancaster University

Email: [d.king@lancaster.ac.uk](mailto:d.king@lancaster.ac.uk)

Young Lee  
Huawei Technologies

Email: [leeyoung@huawei.com](mailto:leeyoung@huawei.com)

Gabriele Galimberti  
Cisco Photonics Srl

Email: [ggalimbe@cisco.com](mailto:ggalimbe@cisco.com)

CCAMP Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: July 12, 2018

J.E. Lopez de Vergara  
Universidad Autonoma de Madrid  
Daniel Perdices  
Naudit HPCN  
V. Lopez  
O. Gonzalez de Dios  
Telefonica I+D/GCTO  
D. King  
Lancaster University  
Y. Lee  
Huawei  
G. Galimberti  
Cisco Photonics Srl  
January 8, 2018

YANG data model for Flexi-Grid Optical Networks  
draft-vergara-ccamp-flexigrid-yang-06.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on July 12, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

Internet-Draft    A YANG data model for Flexi-Grid    January 2018  
carefully, as they describe your rights and restrictions with  
respect to this document. Code Components extracted from this  
document must include Simplified BSD License text as described in  
Section 4.e of the Trust Legal Provisions and are provided without  
warranty as described in the Simplified BSD License.

## Abstract

This document defines a YANG model for managing flexi-grid optical  
Networks. The model described in this document defines a flexi-grid  
traffic engineering database. A complementary module is referenced  
to detail the flexi-grid media channels.

This module is grounded on other defined YANG abstract models.

## Table of Contents

|                                                     |    |
|-----------------------------------------------------|----|
| 1. Introduction .....                               | 2  |
| 2. Conventions used in this document .....          | 3  |
| 3. Flexi-grid network topology model overview ..... | 3  |
| 4. Main building blocks of the Flexi-grid TED.....  | 4  |
| 4.1 Formal Syntax .....                             | 7  |
| 5. Example of use .....                             | 8  |
| 6. Flexi-grid TED YANG Model.....                   | 9  |
| 6.1. YANG Model - Tree .....                        | 9  |
| 6.2. YANG Model - Code .....                        | 10 |
| 6.3. License .....                                  | 19 |
| 7. Security Considerations .....                    | 20 |
| 8. IANA Considerations .....                        | 20 |
| 9. References .....                                 | 20 |
| 9.1. Normative References .....                     | 20 |
| 9.2. Informative References .....                   | 21 |
| 10. Contributors .....                              | 21 |
| 11. Acknowledgments .....                           | 22 |
| Authors' Addresses .....                            | 22 |

## 1. Introduction

Internet-based traffic is dramatically increasing every year.  
Moreover, such traffic is also becoming more dynamic. Thus,  
transport networks need to evolve from current DWDM systems towards  
elastic optical networks, based on flexi-grid transmission and  
switching technologies [RFC7698]. This technology aims at increasing  
both transport network scalability and flexibility, allowing the  
optimization of bandwidth usage.

This document presents a YANG model for flexi-grid objects in the dynamic optical network, including the nodes, transponders and links between them, as well as how such links interconnect nodes and transponders.

The YANG model for flexi-grid networks allows the representation of the flexi-grid optical layer of a network, combined with the underlying physical layer.

This document identifies the flexi-grid components, parameters and their values, characterizes the features and the performances of the flexi-grid elements. An application example is provided towards the end of the document to better understand their utility.

## 2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

## 3. Flexi-grid network topology model overview

YANG is a data modeling language used to model configuration data manipulated by the NETCONF protocol. Several YANG models have already been specified for network configurations. For instance, the work in [I-D.draft-ietf-i2rs-yang-network-topo] has proposed a generic YANG model for network/service topologies and inventories. The work in [I-D.draft-ietf-teas-yang-te-topo] presents a data model to represent, retrieve and manipulate Traffic Engineering (TE) Topologies. These models serve as base models that other technology specific models can augment. A YANG model has also been proposed in [I-D.draft-dharini-ccamp-dwdm-if-yang] to manage single channel optical interface parameters of DWDM applications, and in



Internet-Draft    A YANG data model for Flexi-Grid    January 2018  
[I-D.draft-ietf-ccamp-wson-yang] another model has been specified for the routing and wavelength assignment TE topology in wavelength switched optical networks (WSONs). None of them are specific for flexi-grid technology.

Then, as stated before, we propose a model to describe a flexi-grid topology that is split in two YANG sub-modules:

- o Flexi-grid-TED: In order to be compatible with existing proposals, we augment the definitions contained in [I-D.draft-ietf-i2rs-yang-network-topo] and [I-D.draft-ietf-teas-yang-te-topo], by defining the different elements we can find in a flexi-grid network: a node, a transponder and a link. For that, each of those elements is defined as a container that includes a group of attributes. References to the elements are provided to be later used in the definition of a media channel. It also includes the data types for the type of modulation, the flexi-grid technology, the FEC, etc.
- o Media-channel: This module defines the whole path from a source transponder to the destination through a number of intermediate nodes and links. For this, it takes the information defined before in the flexi-grid TED. This module is described in [I-D.draft-vergara-ccamp-flexigrid-media-channel-yang]

The following section provides a detailed view of the first module.

#### 4. Main building blocks of the Flexi-grid TED

This section details the defined YANG module. It is listed below in section 6.

The description of the three main components, flexi-grid-node, flexi-grid-transponder and flexi-grid-link is provided below. flexi-grid-sliceable-transponders are also defined.

```
<flexi-grid-node> ::= <config> <state>
```

```
<flexi-grid-node>: This element designates a node in the network.
```

```
<config> ::= <flexi-grid-node-attributes-config>
```

```
<config>: Contains the configuration of a node.
<flexi-grid-node-attributes-config> ::= <list-interface>
<connectivity_matrix>
```

```
<flexi-grid-node-attributes-config>: Contains all the attributes related to the node configuration, such as its interfaces or its management addresses.
```

```
<list-interface> ::= <name> <port-number>
<input-port> <output-port> <description>
<interface-type>
[<numbered-interface> / <unnumbered-interface>]
```

<list-interface>: The list containing all the information of the interfaces.

<name>: Determines the interface name.

<port-number>: Port number of the interface.

<input-port>: Boolean value that defines whether the interface is input or not.

<output-port>: Boolean value that defines whether the interface is output or not.

<description>: Description of the usage of the interface.

<interface-type>: Determines if the interface is numbered or unnumbered.

```
<numbered-interface> ::= <n-i-ip-address>
<numbered-interface>: An interface with its own IP address.
```

<n-i-ip-address>: Only available if <interface-type> is "numbered-interface". Determines the IP address of the interface.

```
<unnumbered-interface> ::= <u-i-ip-address>
<label>
```

<unnumbered-interface>: A interface that needs a label to be unique.

<u-i-ip-address>: Only available if <interface-type> is "numbered-interface". Determines the node IP address, which with the label defines the interface.

<label>: Label that determines the interface, joint with the node IP address.

```
<connectivity-matrix> ::= <connections>
```

<connectivity-matrix>: Determines whether a connection port in/port out exists.

```
<connections> ::= <input-port-id>
<output-port-id>
```

<flexi-grid-transponder>: This item designates a transponder of a node.

<config> ::= <flexi-grid-transponder-attributes-config>

<config>: Contains the configuration of a transponder.

<flexi-grid-transponder-attributes-config> ::=  
<available-operational-mode> <operational-mode>

<flexi-grid-transponder-attributes>: Contains all the attributes related to the transponder.

<available-operational-mode>: It provides a list of the operational modes available at this transponder.

<operational-mode>: Determines the type of operational mode in use.

<state> ::= <flexi-grid-transponder-attributes-config>  
<flexi-grid-transponder-attributes-state>

<state>: Contains the state of a transponder.

<flexi-grid-transponder-attributes-config>: See above.

<flexi-grid-transponder-attributes-state>: Contains the state of a transponder.

<link> ::= <config> <state>

<link>: This element describes all the information of a link.

<config> ::= <flexi-grid-link-attributes-config>

<config>: Contains the configuration of a link.

Internet-Draft    A YANG data model for Flexi-Grid    January 2018  
    <flexi-grid-link-attributes-config> ::= <technology-type>  
    <available-label-flexi-grid> <N-max> <base-frequency>  
    <nominal-central-frequency-granularity>  
    <slot-width-granularity>

    <flexi-grid-link-attributes>: Contains all the attributes related to the link, such as its unique id, its N value, its latency, etc.

    <link-id>: Unique id of the link.

    <available-label-flexi-grid>: Array of bits that determines, with each bit, the availability of each interface for flexi-grid technology.

    <N-max>: The max value of N in this link, being N the number of slots.

    <base-frequency>: The default central frequency used in the link.

    <nominal-central-frequency-granularity>: It is the spacing between allowed nominal central frequencies and it is set to 6.25 GHz (note: sometimes referred to as 0.00625 THz).

    <slot-width-granularity>: 12.5 GHz, as defined in G.694.1.

<state> ::= <flexi-grid-link-attributes-config>  
    <flexi-grid-link-attributes-state>

    <state>: Contains the state of a link.

    <flexi-grid-link-attributes-config>: See above.

    <flexi-grid-link-attributes-state>: Contains all the the information related to the state of a link.

#### 4.1. Formal Syntax

The previous syntax specification uses the augmented Backus-Naur Form (BNF) as described in [RFC5234].

In order to explain how this model is used, we provide the following example. An optical network usually has multiple transponders, switches (nodes) and links between them. Figure 1 shows a simple topology, where two physical paths interconnect two optical transponders.

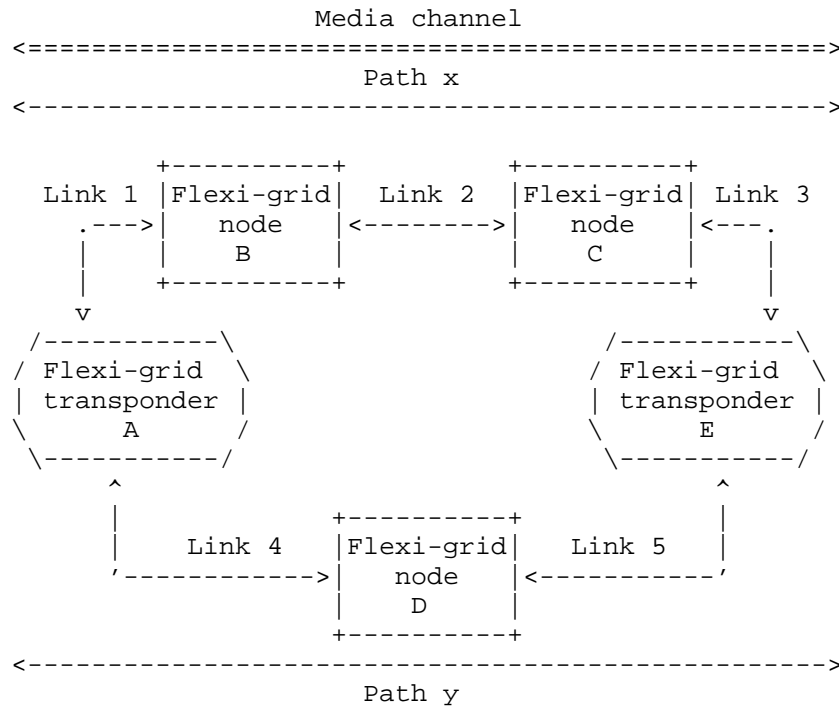


Figure 1. Topology example.

In order to configure a media channel to interconnect transponders A and E, first of all we have to populate the flexi-grid TED YANG model with all elements in the network:

1. We define the transponders A and E, including their FEC type, if enabled, and modulation type. We also provide node identifiers and addresses for the transponders, as well as interfaces included in the transponders. Sliceable transponders can also be defined if needed.
2. We do the same for the nodes B, C and D, providing their identifiers, addresses and interfaces, as well as the internal connectivity matrix between interfaces.
3. Then, we also define the links 1 to 5 that interconnect nodes and transponders, indicating which flexi-grid labels are available. Other information, such as the slot frequency and granularity are also provided.

Next, we can configure the media channel from the information we have stored in the flexi-grid TED, by querying which elements are available, and planning the resources that have to be provided on each situation. Note that every element in the flexi-grid TED has a reference, and this is the way in which they are called in the media channel. We refer to [I-D.draft-vergara-ccamp-flexigrid-media-channel-yang] to complete this example.

## 6. Flexi-grid TED YANG Model

### 6.1. Yang Model - Tree Structure

```

module: ietf-flexi-grid-topology
 augment /nd-s:networks/nd-s:network/nd-s:node/tet-s:te/
 tet-s:te-node-attributes:
 +--ro interfaces* [name]
 +--ro name string
 +--ro port-number? uint32
 +--ro input-port? boolean
 +--ro output-port? boolean
 +--ro description? string
 +--ro type? interface-type
 +--ro numbered-interface
 | +--ro n-i-ip-address? inet:ip-address
 +--ro unnumbered-interface
 | +--ro u-i-ip-address? inet:ip-address
 +--ro label? uint32
 flexi-grid-connectivity-matrix-attributes
 augment /nd:networks/nd:network/nd:node/tet:te/
 tet:te-node-attributes/tet:connectivity-matrices/
 tet:connectivity-matrix:
 +--rw connections* [input-port-id]
 +--rw input-port-id flexi-grid-node-port-ref
 +--rw output-port-id? flexi-grid-node-port-ref
 flexi-grid-connectivity-matrix-attributes
 augment /nd-s:networks/nd-s:network/nd-s:node/tet-s:te/
 tet-s:te-node-attributes/tet-s:connectivity-matrices/
 tet-s:connectivity-matrix:
 +--ro connections* [input-port-id]
 +--ro input-port-id flexi-grid-node-port-ref
 +--ro output-port-id? flexi-grid-node-port-ref
 flexi-grid-transponder
 augment /nd:networks/nd:network/nd:node/tet:te/
 tet:tunnel-termination-point:
 +--rw available-operational-mode* operational-mode
 +--rw operational-mode? operational-mode
 flexi-grid-transponder
 augment /nd-s:networks/nd-s:network/nd-s:node/tet-s:te/
 tet-s:tunnel-termination-point:
 +--ro available-operational-mode* operational-mode
 +--ro operational-mode? operational-mode

```

```
<CODE BEGINS> file "ietf-flexi-grid-ted@2018-01-08.yang"
module ietf-flexi-grid-ted {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-flexi-grid-ted";
 prefix "fg-ted";

 import ietf-network {
 prefix "nd";
 }
 import ietf-network-state {
 prefix "nd-s";
 }
 import ietf-network-topology {
 prefix "lnk";
 }
 import ietf-network-topology-state {
 prefix "lnk-s";
 }
 import ietf-te-topology {
 prefix "tet";
 }
 import ietf-te-topology-state {
 prefix "tet-s";
 }
 import ietf-inet-types {
 prefix "inet";
 }
}

organization
 "IETF CCAMP Working Group";

contact
 "Editor: Jorge Lopez de Vergara
 <jorge.lopez_vergara@uam.es>";
description
 "This module contains a collection of YANG definitions for
 a Flexi-Grid Traffic Engineering Database (TED).

 Copyright (c) 2018 IETF Trust and the persons identified as
 authors of the code. All rights reserved."
```

Internet-Draft    A YANG data model for Flexi-Grid    January 2018  
Redistribution and use in source and binary forms, with or  
without modification, is permitted pursuant to, and subject  
to the license terms contained in, the Simplified BSD  
License set forth in Section 4.c of the IETF Trust's Legal  
Provisions Relating to IETF Documents  
(<http://trustee.ietf.org/license-info>).";

```
revision 2018-01-08 {
 description
 "version 5.";

 reference
 "RFC XXX: A Yang Data Model for
 Flexi-Grid Optical Networks ";
}

/*
 Typedefs
*/

typedef operational-mode {
 type string;
 description
 "Vendor-specific mode that guarantees interoperability.
 It must be an string with the following format:
 B-DScW-ytz(v) where all these attributes are conformant
 to the ITU-T recomendation";
 reference "ITU-T G.698.2 (11/2009) Section 5.3";
}

typedef interface-type {
 type enumeration {
 enum numbered-interface {
 description "The interface is numbered";
 }
 enum unnumbered-interface {
 description "The interface is unnumbered";
 }
 }
 description
 "Enumeration that defines if an interface is numbered or
 unnumbered";
}
```



```

/*
 Typedef related to references
*/
typedef flexi-grid-link-ref {
 type leafref {
 path
 "/nd:networks/nd:network/lnk:link/lnk:link-id";
 }

 description
 "This type is used by data models that need to reference
 a flexi-grid optical link.";
}

typedef flexi-grid-node-port-ref {
 type leafref {
 path "/nd:networks/nd:network/nd:node/tet:te/"
 +"tet:te-node-attributes/fg-ted:interfaces/"
 +"fg-ted:port-number";
 }
 description
 "This type is used by data models that need to reference
 a flexi-grid port.";
}

typedef flexi-grid-transponder-ref {
 type leafref {
 path "/nd:networks/nd:network/nd:node/tet:te/"
 +"tet:tunnel-termination-point/tet:tunnel-tp-id";
 }
 description
 "This type is used by data models that need to reference
 a trasponder.";
}

/*
 Groupings of attributes
*/
grouping flexi-grid-network-type {
 container flexi-grid-network {
 presence "indicates a flexi-grid optical network";
 description "flexi-grid optical network";
 }
 description "If present, it indicates a flexi-grid
 optical TED network";
}

```

```

grouping flexi-grid-node-attributes {
 description "Set of attributes of an optical node.";

 list interfaces {
 key "name";
 unique "port-number"; // TODO Puerto y TP ID
 description "List of interfaces contained in the node";
 leaf name {
 type string;
 description "Interface name";
 }
 leaf port-number {
 type uint32;
 description "Number of the port used by the interface";
 }

 leaf input-port {
 type boolean;
 description "Determines if the port is an input port";
 }
 leaf output-port {
 type boolean;
 description
 "Determines if the port is an output port";
 }
 leaf description {
 type string;
 description "Description of the interface";
 }
 leaf type {
 type interface-type;
 description "Determines the type of the interface";
 }
 container numbered-interface {
 when "../fg-ted:type =
 'numbered-interface'" {
 description
 "If the interface is a numbered interface";
 }
 description "Container that defines an numbered
 interface with an ip-address";
 leaf n-i-ip-address {
 type inet:ip-address;
 description "IP address of the numbered interface";
 }
 }
 }
}

```

```

 container unnumbered-interface {
 when "../fg-ted:type =
 'unnumbered-interface'" {
 description
 "If the interface is an unnumbered interface";
 }
 description "Container that defines an unnumbered
 interface with an ip-address and a label";
 leaf u-i-ip-address{
 type inet:ip-address;
 description "IP address of the interface";
 }
 leaf label {
 type uint32;
 description "Number as label for the interface";
 }
 }
}

grouping flexi-grid-link-attributes {
 description "Set of attributes of an optical link";
 leaf-list available-label-flexi-grid {
 type bits {
 bit is-available{
 description "Set to 1 when it is available";
 }
 }
 description
 "Array of bits that determines whether a spectral
 slot is available or not.";
 }

 leaf N-max {
 type int32;
 description "Maximum number of channels available.";
 }

 leaf base-frequency {
 type decimal64 {
 fraction-digits 5;
 }
 units THz;
 default 193.1;
 description "Default central frequency";
 reference "rfc7698";
 }
}

```

```

Internet-Draft A YANG data model for Flexi-Grid January 2018
 leaf nominal-central-frequency-granularity {
 type decimal64 {
 fraction-digits 5;
 }
 units GHz;
 default 6.25;
 description
 "It is the spacing between allowed nominal central
 frequencies and it is set to 6.25 GHz";
 reference "rfc7698";
 }

 leaf slot-width-granularity {
 type decimal64 {
 fraction-digits 5;
 }
 units GHz;
 default 12.5;
 description "Minimum space between slot widths";
 reference "rfc7698";
 }
}

grouping flexi-grid-transponder-attributes {
 description "Configuration of an optical transponder";
 //TODO Validate attributes
 leaf-list available-operational-mode {
 type operational-mode;
 description "List of all vendor-specific supported
 mode identifiers";
 }

 leaf operational-mode {
 type operational-mode;
 description "Vendor-specific mode identifier";
 }
}

```

```

 grouping flexi-grid-connectivity-matrix-attributes {
 description "Connectivity matrix between the input and
 output ports";
 list connections {
 key "input-port-id";
 leaf input-port-id {
 type flexi-grid-node-port-ref;
 description "Identifier of the input port";
 }
 leaf output-port-id {
 type flexi-grid-node-port-ref;
 description "Identifier of the output port";
 }
 }
 description "List of connections between input and
 output ports";
 }
 }
}

/*
 Augments
*/
augment "/nd:networks/nd:network/nd:network-types" {
 uses flexi-grid-network-type;
 description "Augment network-types including flexi-grid
 topology";
}
augment "/nd-s:networks/nd-s:network/nd-s:network-types" {
 uses flexi-grid-network-type;
 description "Augment network-types including flexi-grid
 topology";
}
augment "/nd:networks/nd:network/lnk:link/tet:te" +
 "/tet:te-link-attributes" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 description "Augment link configuration";
 uses flexi-grid-link-attributes;
}

augment "/nd-s:networks/nd-s:network/lnk-s:link/tet-s:te" +
 "/tet-s:te-link-attributes" {
 when "/nd-s:networks/nd-s:network/nd-s:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 description "Augment link state";
 uses flexi-grid-link-attributes;
}

```

```

Internet-Draft A YANG data model for Flexi-Grid January 2018
augment "/nd:networks/nd:network/nd:node/tet:te" +
 "/tet:te-node-attributes" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }
 uses flexi-grid-node-attributes;
 description "Augment node config with flexi-grid attributes";
 }

augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te" +
 "/tet-s:te-node-attributes" {
 when "/nd-s:networks/nd-s:network/nd-s:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-node-attributes;
 description "Augment node state with flexi-grid attributes";
 }
augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:te-node-attributes/tet:connectivity-matrices/" +
 "tet:connectivity-matrix" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-connectivity-matrix-attributes;
 description "Augment node connectivity-matrix for node config";
 }

augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te"+
 "/tet-s:te-node-attributes/tet-s:connectivity-matrices/" +
 "tet-s:connectivity-matrix" {
 when "/nd-s:networks/nd-s:network/nd-s:network-types/
fg-ted:flexi-grid-network" {
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-connectivity-matrix-attributes;
 description "Augment node connectivity-matrix for node config";
 }

```

```

Internet-Draft A YANG data model for Flexi-Grid January 2018
augment "/nd:networks/nd:network/nd:node/tet:te"+
 "/tet:tunnel-termination-point" {
 when "/nd:networks/nd:network/nd:network-types/
fg-ted:flexi-grid-network"{
 description "Augment only for Flexigrid network.";
 }
 uses flexi-grid-transponder-attributes;
 description "Augment node state with transponder attributes";
 }

augment "/nd-s:networks/nd-s:network/nd-s:node/tet-s:te"+
 "/tet-s:tunnel-termination-point" {
 when "/nd-s:networks/nd-s:network/nd-s:network-types/
fg-ted:flexi-grid-network"{
 description "Augment only for Flexigrid network.";
 }

 uses flexi-grid-transponder-attributes;
 description "Augment node state with transponder attributes";
 }
}

<CODE ENDS>

```

Copyright (c) 2018 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- o Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- o Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



## 7. Security Considerations

The transport protocol used for sending the managed information MUST support authentication and SHOULD support encryption.

The defined data-model by itself does not create any security implications.

## 8. IANA Considerations

The namespace used in the defined models is currently based on the METRO-HAUL project URI. Future versions of this document could register a URI in the IETF XML registry [RFC3688], as well as in the YANG Module Names registry [RFC6020].

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

- [RFC7698] Gonzalez de Dios, O., Casellas, R., Eds. "Framework and Requirements for GMPLS-Based Control of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks", RFC7698, November 2015.
- [I-D.draft-ietf-i2rs-yang-network-topo] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., Liu, X., "A Data Model for Network Topologies", Internet Draft draft-ietf-i2rs-yang-network-topo-20.txt, 2017.
- [I-D.draft-ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Pavan Beeram, V., Saad, T., Shah, H., Gonzalez De Dios, O., "YANG Data Model for TE Topologies", Internet Draft draft-ietf-teas-yang-te-topo-13.txt, 2017
- [I-D.draft-dharini-ccamp-dwdm-if-yang] Galimberti, G., Kunze, R., Lam, K., Hiremagalur, D., Grammel, G., Fang, L., Ratterree, G., Eds., "A YANG model to manage the optical interface parameters for an external transponder in a WDM network", Internet Draft, draft-dharini-ccamp-dwdm-if-param-yang-03.txt, 2016.
- [I-D.draft-ietf-ccamp-wson-yang] Lee, Y. Dhody, D., Zhang, X., Guo, A., Lopez, V., King, D., Yoon, B., "A Yang Data Model for WSON Optical Networks", Internet Draft, draft-ietf-ccamp-wson-yang-09.txt, 2017.
- [I-D.draft-vergara-ccamp-flexigrid-media-channel-yang] Lopez de Vergara, J., Perdices, D., Lopez, V., Gonzalez de Dios, O., King, D., Lee, Y., Galimberti, G., "YANG data model for Flexi-Grid media-channels", Internet Draft, draft-vergara-ccamp-flexigrid-media-channel-yang-01, 2017.

## 10. Contributors

The model presented in this paper was contributed to by more people than can be listed on the author list. Additional contributors include:

- o Zafar Ali, Cisco Systems
- o Daniel Michaud Vallinoto, Universidad Autonoma de Madrid

## 11. Acknowledgments

The work presented in this Internet-Draft has been partially funded by the European Commission under the project H2020 METRO-HAUL (Metro High bandwidth, 5G Application-aware optical network, with edge storage, compUte and low Latency), Grant Agreement number: 761727, and by the Spanish Ministry of Economy and Competitiveness under the project TRAFICA, MINECO/FEDER TEC2015-69417-C2-1-R.

## Authors' Addresses

Jorge E. Lopez de Vergara  
Universidad Autonoma de Madrid  
Escuela Politecnica Superior  
C/Francisco Tomas y Valiente, 11  
E-28049 Madrid, Spain

Email: [jorge.lopez\\_vergara@uam.es](mailto:jorge.lopez_vergara@uam.es)

Daniel Perdices Burrero  
Naudit High Performance Computing and Networking, S.L.  
C/Faraday, 7  
E-28049 Madrid, Spain

Email: [daniel.perdices@naudit.es](mailto:daniel.perdices@naudit.es)

Victor Lopez  
Telefonica I+D/GCTO  
Distrito Telefonica  
E-28050 Madrid, Spain

Email: [victor.lopezalvarez@telefonica.com](mailto:victor.lopezalvarez@telefonica.com)

Oscar Gonzalez de Dios  
Telefonica I+D/GCTO  
Distrito Telefonica  
E-28050 Madrid, Spain

Email: [oscar.gonzalezdedios@telefonica.com](mailto:oscar.gonzalezdedios@telefonica.com)

Daniel King  
Lancaster University

Email: [d.king@lancaster.ac.uk](mailto:d.king@lancaster.ac.uk)

Young Lee  
Huawei Technologies

Email: [leeyoung@huawei.com](mailto:leeyoung@huawei.com)

Gabriele Galimberti  
Cisco Photonics Srl

Email: [ggalimbe@cisco.com](mailto:ggalimbe@cisco.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 10, 2017

X. Xu, Ed.  
S. Bryant  
Huawei  
R. Raszuk  
Bloomberg LP  
U. Chunduri  
Huawei  
L. Contreras  
Telefonica I+D  
L. Jalil  
Verizon  
H. Assarpour  
Broadcom  
March 9, 2017

Unified Source Routing Instruction using MPLS Label Stack  
draft-xu-mpls-unified-source-routing-instruction-00

Abstract

MPLS-SPRING is an MPLS-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. This MPLS -based source routing paradigm could actually be leveraged to realize a unified source routing instruction for both IPv4 and IPv6 underlays.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2017.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                           |   |
|-------------------------------------------|---|
| 1. Introduction . . . . .                 | 2 |
| 1.1. Requirements Language . . . . .      | 3 |
| 2. Terminology . . . . .                  | 3 |
| 3. Use Cases . . . . .                    | 3 |
| 4. Packet Forwarding Procedures . . . . . | 4 |
| 5. Acknowledgements . . . . .             | 7 |
| 6. IANA Considerations . . . . .          | 7 |
| 7. Security Considerations . . . . .      | 7 |
| 8. References . . . . .                   | 7 |
| 8.1. Normative References . . . . .       | 7 |
| 8.2. Informative References . . . . .     | 7 |
| Authors' Addresses . . . . .              | 9 |

## 1. Introduction

MPLS-SPRING [I-D.ietf-spring-segment-routing-mpls] is a MPLS-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. This MPLS-based source routing paradigm could actually be leveraged to realize a unified source routing instruction for both IPv4 and IPv6 underlays. In other words, the source routing instruction information contained in IPv4 and IPv6 source routed packets could be uniformly encoded as an MPLS label stack. As a result, there is no need any more to develop and implement transport-dependent source routing mechanisms for IPv4 and IPv6 respectively.

The traditional IPv4 and IPv6 source routing mechanisms by use of IPv4 Source Routing Options and IPv6 Route Header Type 0 Extension respectively have been deprecated due to their obvious security vulnerabilities. IPv6 SPRING [I-D.ietf-6man-segment-routing-header]

is a newly proposed IPv6 source routing mechanism in which the source route instruction information is encoded as an ordered list of 128-bit long IPv6 addresses and contained in the Source Routing Header (SRH). Although it has overcome the security vulnerability issues associated with the traditional IPv6 source routing mechanism as claimed in [I-D.ietf-6man-segment-routing-header], it still has the following obvious drawbacks at least: 1) the encapsulation overhead is significant especially when the list of the explicit routing hops is very long; 2) for those transit IPv6 routers that don't support the flow label based load-balancing mechanism yet, the ECMP load-balancing effect may be impacted seriously since they could not recognize the SRH and therefore could not obtain the five tuple of the source routed IPv6 packet; 3) it requires a new forwarding logic on basis of the SRH and the forwarding performance associated with the IPv6 SRH may still be a big concern for some hardware platforms.

Section 3 describes various use cases for the unified source routing and Section 4 describes a typical application scenario and how the packet forwarding happens.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Terminology

This memo makes use of the terms defined in [RFC3031] and [I-D.ietf-spring-segment-routing-mpls].

## 3. Use Cases

The unified source routing mechanism across MPLS, IPv4 and IPv6 is useful at least in the following use cases:

- o Incremental deployment of the MPLS-SPRING technology. Since there is no need to run any other label distribution protocol (e.g., LDP, see [I-D.filsfils-spring-segment-routing-ldp-interop] for more details.) on those non-MPLS-SPRING routers, the network provisioning is greatly simplified, which is one of the major claimed benefits of the MPLS-SPRING technology (i.e., running a single protocol).
- o MPLS-based Service Function Chaining (SFC) [I-D.xu-mpls-service-chaining]. Based on the unified source routing mechanism as described in this document, only SFC-related

nodes including Service Function Forwarders (SFF), Service Functions (SF) and classifiers are required to recognize the SFC encapsulation header in the MPLS label stack form, while the intermediate routers just need to support vanilla IP forwarding (either IPv4 or IPv6). In other words, it undoubtedly complies with the transport-independence requirement as listed in the SFC architecture document [RFC7665].

- o Traffic Engineering scenarios where only a few routers (e.g., the entry and exit nodes of each plane in the dual-plane network ) are specified as segments of explicit paths. In this way, only a few routers are required to support the MPLS-SPRING capability while all the other routers just need to support IP forwarding capability, which would significantly reduce the deployment cost of this new technology.
- o A light-weight alternative to IPv6 SPRING technology [I-D.ietf-6man-segment-routing-header]. The Source Routing Header (SRH) [I-D.ietf-6man-segment-routing-header] consisting of an ordered list of 128-bit long IPv6 addresses is now replaced by an ordered list of 20-bit long labels (i.e., label stack). As a result, the encapsulation overhead and forwarding performance issues associated with the IPv6 SPRING are eliminated.
- o A new IPv4 source routing mechanism which has overcome the security vulnerability issues associated with the traditional IPv4 source routing mechanism.

#### 4. Packet Forwarding Procedures

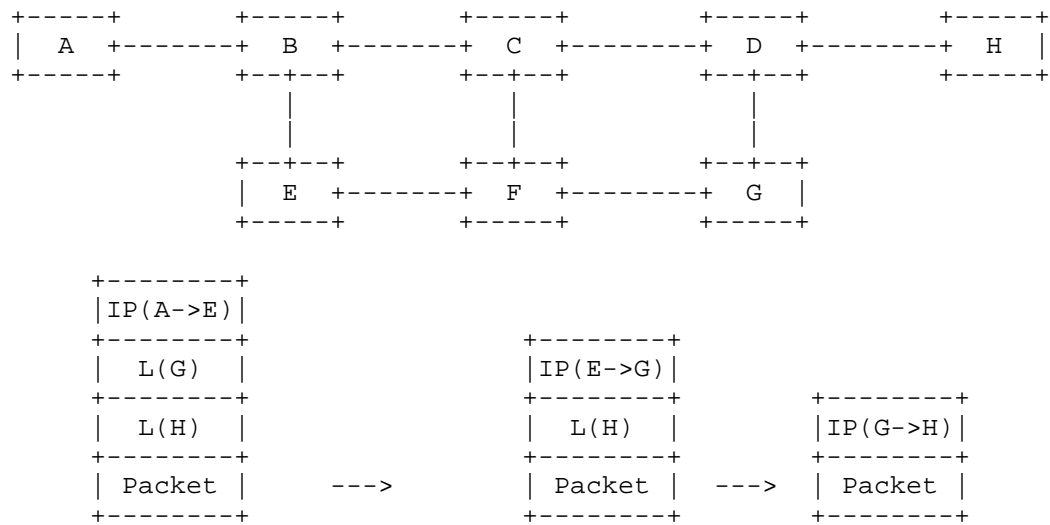


Figure 1

As shown in Figure 1, Assume Router A, E, G and H are MPLS-SPRING-capable routers while the remaining are only capable of forwarding IP packets. Router A, E, G and H advertise their Segment Routing related information via IS-IS or OSPF. Now assume router A wants to send a given IP or MPLS packet via an explicit path of {E->G->H}, router A would impose an MPLS label stack corresponding to that explicit path on the received IP packet. Since there is no Label Switching Path (LSP) towards router E, router A would replace the top label indicating router E with an IP-based tunnel for MPLS (e.g., MPLS-over-UDP [RFC7510] or MPLS-over-GRE [RFC4023]) towards router E and then send it out. In other words, router A would pop the top label and then encapsulate the MPLS packet with an IP-based tunnel towards router E. When the IP-encapsulated MPLS packet arrives at router E, router E would strip the IP-based tunnel header and then process the decapsulated MPLS packet accordingly. Since there is no LSP towards router G which is indicated by the current top label of the decapsulated MPLS packet, router E would replace the current top label with an IP-based tunnel towards router G and send it out. When the packet arrives at router G, router G would strip the IP-based tunnel header and then process the decapsulated MPLS packet. Since there is no LSP towards router H, router G would replace the current top label with an IP-based tunnel towards router H. Now the packet encapsulated with the IP-based tunnel towards router H is exactly the original packet that router A had intended to send towards router H. If the packet is an MPLS packet, router G could use any IP-based tunnel for MPLS (e.g., MPLS-over-UDP [RFC7510] or MPLS-over-GRE [RFC4023]). If the packet is an IP packet, router G could use any IP tunnel for IP (e.g., IP-in-UDP [I-D.xu-intarea-ip-in-udp] or GRE



[RFC2784]). That original IP or MPLS packet would be forwarded towards router H via an IP-based tunnel. When the encapsulated packet arrives at router H, router H would decapsulate it into the original packet and then process it accordingly. Note that in the above description, it's assumed that the label associated with each prefix-SID advertised by the owner of the prefix-SID is a Penultimate Hop Popping (PHP) label (e.g., the NP-flag [I-D.ietf-ospf-segment-routing-extensions] associated with the corresponding prefix SID is not set). Figure 2 demonstrates the packet walk in the case where the label associated with each prefix-SID advertised by the owner of the prefix-SID is not a Penultimate Hop Popping (PHP) label (e.g., the NP-flag [I-D.ietf-ospf-segment-routing-extensions] associated with the corresponding prefix SID is set).

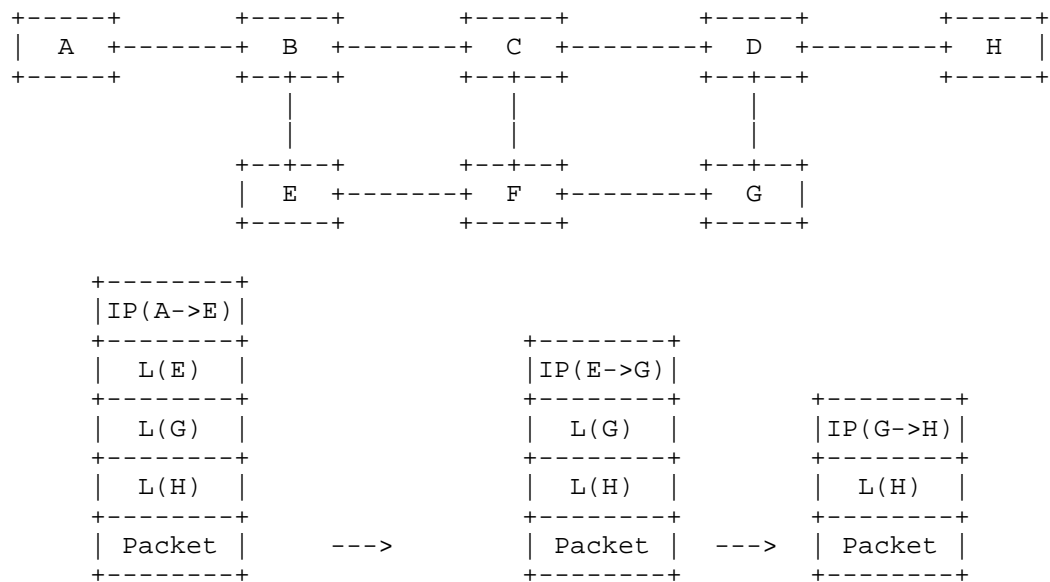


Figure 2

Note that as for which tunnel encapsulation type should be used, it could be manually specified on each tunnel ingress routers or be learnt from the tunnel egress routers' advertisements of its tunnel encapsulation capability. How to advertise the tunnel encapsulation capability using IS-IS or OSPF are specified in [I-D.ietf-isis-encapsulation-cap] and [I-D.ietf-ospf-encapsulation-cap] respectively.

## 5. Acknowledgements

Thanks Joel Halpern, Bruno Decraene and Loa Andersson for their insightful comments on this draft.

## 6. IANA Considerations

No IANA action is required.

## 7. Security Considerations

TBD.

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

### 8.2. Informative References

[I-D.filsfils-spring-segment-routing-ldp-interop]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing interoperability with LDP", draft-filsfils-spring-segment-routing-ldp-interop-03 (work in progress), March 2015.

[I-D.ietf-6man-segment-routing-header]  
Previdi, S., Filsfils, C., Field, B., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., and D. Lebrun, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-05 (work in progress), February 2017.

[I-D.ietf-isis-encapsulation-cap]  
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in IS-IS", draft-ietf-isis-encapsulation-cap-00 (work in progress), October 2016.

- [I-D.ietf-ospf-encapsulation-cap]  
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in OSPF", draft-ietf-ospf-encapsulation-cap-01 (work in progress), October 2016.
- [I-D.ietf-ospf-segment-routing-extensions]  
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-12 (work in progress), March 2017.
- [I-D.ietf-spring-segment-routing-mpls]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., jeffrant@gmail.com, j., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-07 (work in progress), February 2017.
- [I-D.xu-intarea-ip-in-udp]  
Xu, X., Lee, Y., and F. Yongbing, "Encapsulating IP in UDP", draft-xu-intarea-ip-in-udp-04 (work in progress), December 2016.
- [I-D.xu-mpls-service-chaining]  
Xu, X., Bryant, S., Assarpour, H., Shah, H., Contreras, L., and d. daniel.bernier@bell.ca, "Service Chaining using MPLS Source Routing", draft-xu-mpls-service-chaining-00 (work in progress), October 2016.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<http://www.rfc-editor.org/info/rfc2784>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<http://www.rfc-editor.org/info/rfc4023>>.

- [RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <<http://www.rfc-editor.org/info/rfc4817>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<http://www.rfc-editor.org/info/rfc7510>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

Authors' Addresses

Xiaohu Xu (editor)  
Huawei

Email: [xuxiaohu@huawei.com](mailto:xuxiaohu@huawei.com)

Stewart Bryant  
Huawei

Email: [stewart.bryant@gmail.com](mailto:stewart.bryant@gmail.com)

Robert Raszuk  
Bloomberg LP

Email: [robert@raszuk.net](mailto:robert@raszuk.net)

Uma Chunduri  
Huawei

Email: [uma.chunduri@gmail.com](mailto:uma.chunduri@gmail.com)

Luis M. Contreras  
Telefonica I+D

Email: [luismiguel.contrerasmurillo@telefonica.com](mailto:luismiguel.contrerasmurillo@telefonica.com)

Luay Jalil  
Verizon

Email: [luay.jalil@verizon.com](mailto:luay.jalil@verizon.com)

Hamid Assarpour  
Broadcom

Email: [hamid.assarpour@broadcom.com](mailto:hamid.assarpour@broadcom.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 1, 2018

X. Xu  
Huawei  
A. Bashandy  
Cisco  
H. Assarpour  
Broadcom  
S. Ma  
Juniper  
W. Henderickx  
Nokia  
J. Tantsura  
Individual  
September 28, 2017

Unified Source Routing Instructions using MPLS Label Stack  
draft-xu-mpls-unified-source-routing-instruction-04

Abstract

MPLS Segment Routing (SR-MPLS in short) is an MPLS data plane-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. SR-MPLS could be leveraged to realize a unified source routing mechanism across MPLS, IPv4 and IPv6 data planes by using an MPLS label stack as a unified source routing instruction set while preserving backward compatibility with SR-MPLS.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 1, 2018.

## Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                              |    |
|----------------------------------------------|----|
| 1. Introduction . . . . .                    | 2  |
| 1.1. Requirements Language . . . . .         | 3  |
| 2. Terminology . . . . .                     | 3  |
| 3. Use Cases . . . . .                       | 3  |
| 4. Packet Forwarding Procedures . . . . .    | 4  |
| 4.1. Forwarding Entry Construction . . . . . | 5  |
| 4.2. Packet Forwarding Procedures . . . . .  | 6  |
| 5. Contributors . . . . .                    | 9  |
| 6. Acknowledgements . . . . .                | 10 |
| 7. IANA Considerations . . . . .             | 10 |
| 8. Security Considerations . . . . .         | 10 |
| 9. References . . . . .                      | 10 |
| 9.1. Normative References . . . . .          | 10 |
| 9.2. Informative References . . . . .        | 10 |
| Authors' Addresses . . . . .                 | 13 |

## 1. Introduction

MPLS Segment Routing (SR-MPLS in short) [I-D.ietf-spring-segment-routing-mpls] is an MPLS data plane-based source routing paradigm in which a sender of a packet is allowed to partially or completely specify the route the packet takes through the network by imposing stacked MPLS labels to the packet. SR-MPLS could be leveraged to realize a unified source routing mechanism across MPLS, IPv4 and IPv6 data planes by using an MPLS label stack as a unified source routing instruction set while preserving backward compatibility with SR-MPLS. More specifically, the source routing instruction set information contained in a source routed packet could be uniformly encoded as an MPLS label stack no matter the underlay is IPv4, IPv6 or MPLS.

Although the source routing instructions are encoded as MPLS labels, this is a hardware convenience rather than an indication that the whole MPLS protocol stack and in particular the MPLS control protocols need to be deployed. Note that the complexity associated with the whole MPLS protocol stack is largely due to the complex control plane protocols.

Section 3 describes various use cases for the unified source routing instruction mechanism and Section 4 describes a typical application scenario and how the packet forwarding happens.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

## 2. Terminology

This memo makes use of the terms defined in [RFC3031] and [I-D.ietf-spring-segment-routing-mpls].

## 3. Use Cases

The unified source routing mechanism across IPv4, IPv6 and MPLS is useful at least in the following use cases:

- o Incremental deployment of the SR-MPLS technology [I-D.xu-mpls-spring-islands-connection-over-ip]. Since there is no need to run any other label distribution protocol (e.g., LDP, see [I-D.ietf-spring-segment-routing-ldp-interop] for more details.) on those non-SR-MPLS routers for incremental deployment purposes, the network provisioning is greatly simplified, which is one of the major claimed benefits of the SR-MPLS technology (i.e., running a single protocol).
- o Overcome the load-balancing dilemma encountered by SR-MPLS. In fact, this unified source routing mechanism is even useful in a fully upgraded SR-MPLS network since the load-balancing dilemma encountered by SR-MPLS [I-D.ietf-mpls-spring-entropy-label] due to the maximum Readable Label-stack Depth (RLD) hardware limitation [I-D.ietf-ospf-mpls-elc] [I-D.ietf-isis-mpls-elc] [I-D.ietf-idr-bgp-ls-segment-routing-rld] and the Maximum SID Depth (MSD) hardware limitation [I-D.ietf-ospf-segment-routing-msd] [I-D.ietf-isis-segment-routing-msd] [I-D.ietf-idr-bgp-ls-segment-routing-msd] by using the MPLS-in-UDP



encapsulation [RFC7510] where the source port of the UDP tunnel header is used as an entropy field.

- o A poor man's light-weight alternative to SRv6 [I-D.ietf-6man-segment-routing-header]. At least, it could be deployed as an interim until full featured SRv6 is available on more platforms. Since the Source Routing Header (SRH) [I-D.ietf-6man-segment-routing-header] consisting of an ordered list of 128-bit long IPv6 addresses is now replaced by an ordered list of 32-bit long label entries (i.e., label stack), the encapsulation overhead and forwarding performance issues associated with SRv6 are eliminated.
- o A new IPv4 source routing mechanism which has overcome the security vulnerability issues associated with the traditional IPv4 source routing mechanism.
- o Traffic Engineering scenarios where only a few routers (e.g., the entry and exit nodes of each plane in the dual-plane network case or the egress node in the Egress Peer Engineering (EPE) case) are specified as segments of explicit paths. In this way, only a few routers are required to support the SR-MPLS capability while all the other routers just need to support IP forwarding capability, which would significantly reduce the deployment cost of the SR-MPLS technology.
- o MPLS-based Service Function Chaining (SFC) [I-D.xu-mpls-service-chaining]. Based on the unified source routing mechanism as described in this document, only SFC-related nodes including Service Function Forwarders (SFF), Service Functions (SF) and classifiers are required to recognize the SFC encapsulation header in the MPLS label stack form, while the intermediate routers just need to support vanilla IP forwarding (either IPv4 or IPv6). In other words, it undoubtedly complies with the transport-independence requirement for the SFC encapsulation header as listed in the SFC architecture document [RFC7665].

#### 4. Packet Forwarding Procedures

The primary objective of this document is to describe how SR-MPLS capable routers and IP-only routers can seamlessly co-exist and interoperate. This section describes the forwarding information base (FIB) entry and the forwarding behavior that allow the deployment of SR-MPLS when some routers are IPv4 only or IPv6 only. Note that OSPF or ISIS is assumed to be enabled in the following examples as described in Section 4.1 and 4.2, in fact, it's no doubt that BGP could be used as a replacement.

#### 4.1. Forwarding Entry Construction

This sub-section describes the how to construct the forwarding information base (FIB) entry on an SR-MPLS-capable router when some or all of the next-hops along the shortest path towards a prefix-SID are IPv4-only or IPv6-only routers. Consider the router "A" receiving a labeled packet whose top label L(E) corresponds to the prefix-SID is "SID(E)" of prefix "P(E)" advertised by the router "E". Suppose the *i*th next-hop router "NHi" along the shortest path from the router "A" towards the prefix-SID "SID(E)" is not SR-MPLS capable. That is both routers "A" and "E" are SR-MPLS capable but the next hop "NHi" along the shortest path from "A" to "E". The following applies:

- o It is assumed that the router "E" advertises the SR-Capabilities sub-TLV as described in and [I-D.ietf-ospf-segment-routing-extensions], which includes the SRGB because router "E" is SR-MPLS capable.
- o The owning router "E" MUST advertise the encapsulation endpoint and the tunnel type using [I-D.ietf-isis-encapsulation-cap] and/or [I-D.ietf-ospf-encapsulation-cap] .
- o If "A" and "E" are in different areas/levels, then
  - \* The OSPF Tunnel Encapsulation TLV [I-D.ietf-ospf-encapsulation-cap] and/or the ISIS Tunnel Encapsulation sub-TLV [I-D.ietf-isis-encapsulation-cap] are flooded domain-wide.
  - \* The OSPF SID/label range TLV [I-D.ietf-ospf-segment-routing-extensions] and the ISIS SR-Capabilities Sub-TLV [I-D.ietf-isis-segment-routing-extensions] are advertised domain-wide. This way router "A" knows the characteristics of the owning router "E".
  - \* When the owning router "E" is running ISIS and advertises the prefix "P(E) ", the router "E" uses the extended reachability TLV (TLVs 135, 235, 236, 237) and associates the IPv4/IPv6 and/or IPv4/IPv6 source router ID sub-TLV(s) [RFC7794].
  - \* When the owning router "E" is running OSPF and advertises the prefix "P(E)", the router "E" uses the OSPFv2 Extended Prefix Opaque LSA [RFC7684] and sets the flooding scope to AS-wide.
  - \* When the owning router "E" is running ISIS and advertises the ISIS capabilities TLV (TLV 242) [RFC7981], it must set the "router-ID" field to a valid value or include IPV6 TE router-

ID sub-TLV (TLV 12), or do both. The "S" bit (flooding scope) of the ISIS capabilities TLV (TLV 242) MUST be set to "1" .

- o Router "A" programs the FIB entry corresponding to the "SID(E)" as follows:

- \* If NP (OSPF) or P (ISIS) flag is clear,
- \*
  - + pop the outer label.
- \* If NP (OSPF) or P (ISIS) is set,
- \*
  - + the outer label is SID(E) plus the lower bound of the SRGB of "E".
- \* Encapsulate the packet according to the encapsulation advertised in [I-D.ietf-isis-encapsulation-cap] or [I-D.ietf-ospf-encapsulation-cap].
- \* Send the packet towards the next hop "NH<sub>i</sub>".

#### 4.2. Packet Forwarding Procedures

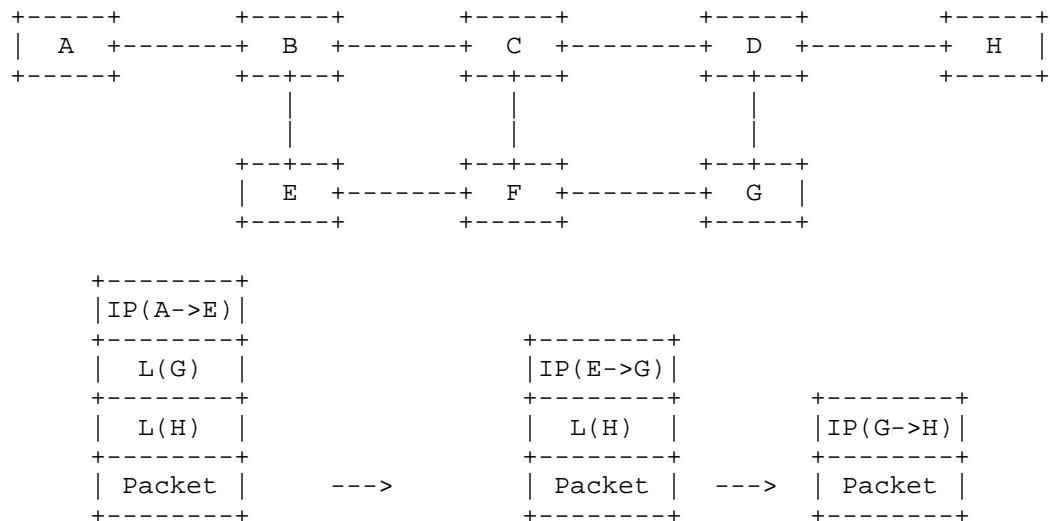


Figure 1

As shown in Figure 1, Assume Router A, E, G and H are SR-MPLS-capable routers while the remaining are only capable of forwarding IP packets. Router A, E, G and H advertise their Segment Routing related information via IS-IS or OSPF. Now assume router A wants to send a given IP or MPLS packet via an explicit path of {E->G->H}, router A would impose an MPLS label stack corresponding to that explicit path on the received IP packet. Since there is no Label Switching Path (LSP) towards router E, router A would replace the top label indicating router E with an IP-based tunnel for MPLS (e.g., MPLS-over-UDP [RFC7510]) towards router E and then send it out. In other words, router A would pop the top label and then encapsulate the MPLS packet with an IP-based tunnel towards router E. When the IP-encapsulated MPLS packet arrives at router E, router E would strip the IP-based tunnel header and then process the decapsulated MPLS packet accordingly. Since there is no LSP towards router G which is indicated by the current top label of the decapsulated MPLS packet, router E would replace the current top label with an IP-based tunnel towards router G and send it out. When the packet arrives at router G, router G would strip the IP-based tunnel header and then process the decapsulated MPLS packet. Since there is no LSP towards router H, router G would replace the current top label with an IP-based tunnel towards router H. Now the packet encapsulated with the IP-based tunnel towards router H is exactly the original packet that router A had intended to send towards router H. If the packet is an MPLS packet, router G could use any IP-based tunnel for MPLS (e.g., MPLS-over-UDP [RFC7510]). If the packet is an IP packet, router G could use any IP tunnel for IP (e.g., IP-in-UDP [I-D.xu-intarea-ip-in-udp]). That original IP or MPLS packet would be forwarded towards router H via an IP-based tunnel. When the encapsulated packet arrives at router H, router H would decapsulate it into the original packet and then process it accordingly.

Note that in the above description, it's assumed that the label associated with each prefix-SID advertised by the owner of the prefix-SID is a Penultimate Hop Popping (PHP) label (e.g., the NP-flag [I-D.ietf-ospf-segment-routing-extensions] associated with the corresponding prefix SID is not set).

Figure 2 demonstrates the packet walk in the case where the label associated with each prefix-SID advertised by the owner of the prefix-SID is not a Penultimate Hop Popping (PHP) label (e.g., the NP-flag [I-D.ietf-ospf-segment-routing-extensions] associated with the corresponding prefix SID is set).

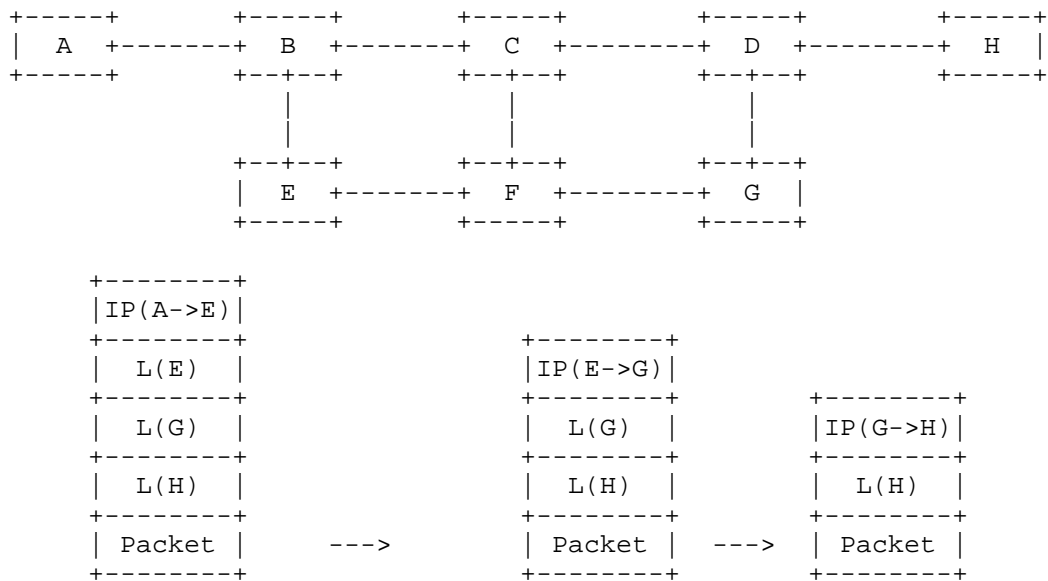


Figure 2

Although the above description is based on the use of prefix-SIDs, the unified source routing instruction approach is actually applicable to the use of adj-SIDs as well. For instance, when the top label of a received MPLS packet indicates an given adj-SID and the corresponding adjacent node to that adj-SID is not MPLS-capable, the top label would be replaced by an IP-based tunnel towards that adjacent node and then forwarded over the corresponding link indicated by that adj-SID.

When encapsulating an MPLS packet with an IP-based tunnel header (e.g., a UDP header as per [RFC7510]), the corresponding entropy field (i.e., the source port in the MPLS-in-UDP case) should be filled with an entropy value that is generated by the encapsulator to uniquely identify a flow. However, what constitutes a flow is locally determined by the encapsulator. For instance, if the MPLS label stack contains at least one entropy label and the encapsulator is capable of reading that entropy label, the entropy label value could be directly copied to the entropy field (e.g., the source port of the UDP header). Otherwise, the encapsulator may have to perform a hash on the whole label stack or the five-tuple of the MPLS payload if the payload is determined as an IP packet. To avoid re-performing hash on the whole packet when re-encapsulating the packet with an IP-based tunnel header (e.g., a UDP tunnel header), especially when the encapsulator could not obtain at least one entropy label due to some reasons (e.g., 1) there is no EL at all in the label stack; 2) the encapsulator couldn't recognize the ELI; 3) the encapsulator could

not read the EL due to the RLD limit), it's RECOMMENDED that the entropy value contained in the packet (e.g., the UDP source port value) is kept when stripping the IP-based tunnel header (e.g., the UDP tunnel header). As such, the entropy value could be directly copied to the entropy field (e.g., the source port of the UDP tunnel header) when re-encapsulating the packet with an IP-based tunnel header (e.g., a UDP tunnel header). As such, the load-balancing dilemma encountered by SR-MPLS as described in [I-D.ietf-mpls-spring-entropy-label] due to the maximum Readable Label-stack Depth (RLD) hardware limitation [I-D.ietf-ospf-mpls-elc] [I-D.ietf-isis-mpls-elc] and the Maximum SID Depth (MSD) hardware limitation [I-D.ietf-ospf-segment-routing-msd] [I-D.ietf-isis-segment-routing-msd] is gone. That's the reason why this unified source routing mechanism is even useful in a fully upgraded SR-MPLS network environment.

## 5. Contributors

Clarence Filsfils  
Cisco  
Email: cfilsfil@cisco.com

Robert Raszuk  
Bloomberg LP  
Email: robert@raszuk.net

Uma Chunduri  
Huawei  
Email: uma.chunduri@gmail.com

Luis M. Contreras  
Telefonica I+D  
Email: luismiguel.contrerasmurillo@telefonica.com

Luay Jalil  
Verizon  
Email: luay.jalil@verizon.com

Gunter Van De Velde  
Nokia  
Email: gunter.van\_de\_velde@nokia.com

Tal Mizrahi  
Marvell  
Email: talmi@marvell.com

## 6. Acknowledgements

Thanks Joel Halpern, Bruno Decraene, Loa Andersson and Stewart Bryant for their insightful comments on this document.

## 7. IANA Considerations

No IANA action is required.

## 8. Security Considerations

TBD.

## 9. References

### 9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

### 9.2. Informative References

[I-D.ietf-6man-segment-routing-header]  
Previdi, S., Filsfils, C., Raza, K., Leddy, J., Field, B., daniel.voyer@bell.ca, d., daniel.bernier@bell.ca, d., Matsushima, S., Leung, I., Linkova, J., Aries, E., Kosugi, T., Vyncke, E., Lebrun, D., Steinberg, D., and R. Raszuk, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-07 (work in progress), July 2017.

[I-D.ietf-idr-bgp-ls-segment-routing-msd]  
Tantsura, J., Chunduri, U., Mirsky, G., and S. Sivabalan, "Signaling Maximum SID Depth using Border Gateway Protocol Link-State", draft-ietf-idr-bgp-ls-segment-routing-msd-00 (work in progress), July 2017.

[I-D.ietf-idr-bgp-ls-segment-routing-rld]  
Velde, G., Henderickx, W., Bocci, M., and K. Patel, "Signalling ERLD using BGP-LS", draft-ietf-idr-bgp-ls-segment-routing-rld-00 (work in progress), July 2017.

[I-D.ietf-isis-encapsulation-cap]  
Xu, X., Decraene, B., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Advertising Tunnelling Capability in IS-IS", draft-ietf-isis-encapsulation-cap-01 (work in progress), April 2017.

[I-D.ietf-isis-mpls-elc]

Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S. Litkowski, "Signaling Entropy Label Capability Using IS-IS", draft-ietf-isis-mpls-elc-02 (work in progress), October 2016.

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and j. jeffrant@gmail.com, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-13 (work in progress), June 2017.

[I-D.ietf-isis-segment-routing-msd]

Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling MSD (Maximum SID Depth) using IS-IS", draft-ietf-isis-segment-routing-msd-04 (work in progress), June 2017.

[I-D.ietf-mpls-spring-entropy-label]

Kini, S., Kompella, K., Sivabalan, S., Litkowski, S., Shakir, R., and j. jeffrant@gmail.com, "Entropy label for SPRING tunnels", draft-ietf-mpls-spring-entropy-label-06 (work in progress), May 2017.

[I-D.ietf-ospf-encapsulation-cap]

Xu, X., Decraene, B., Raszuk, R., Contreras, L., and L. Jalil, "The Tunnel Encapsulations OSPF Router Information", draft-ietf-ospf-encapsulation-cap-08 (work in progress), September 2017.

[I-D.ietf-ospf-mpls-elc]

Xu, X., Kini, S., Sivabalan, S., Filsfils, C., and S. Litkowski, "Signaling Entropy Label Capability Using OSPF", draft-ietf-ospf-mpls-elc-04 (work in progress), November 2016.

[I-D.ietf-ospf-segment-routing-extensions]

Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-ietf-ospf-segment-routing-extensions-19 (work in progress), August 2017.

[I-D.ietf-ospf-segment-routing-msd]

Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling MSD (Maximum SID Depth) using OSPF", draft-ietf-ospf-segment-routing-msd-05 (work in progress), June 2017.



- [I-D.ietf-spring-segment-routing-ldp-interop]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., and S. Litkowski, "Segment Routing interworking with LDP", draft-ietf-spring-segment-routing-ldp-interop-08 (work in progress), June 2017.
- [I-D.ietf-spring-segment-routing-mpls]  
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.
- [I-D.xu-intarea-ip-in-udp]  
Xu, X., Lee, Y., and F. Yongbing, "Encapsulating IP in UDP", draft-xu-intarea-ip-in-udp-04 (work in progress), December 2016.
- [I-D.xu-mpls-service-chaining]  
Xu, X., Bryant, S., Assarpour, H., Shah, H., Contreras, L., daniel.bernier@bell.ca, d., jefftant@gmail.com, j., Ma, S., and M. Vigoureux, "Service Chaining using Unified Source Routing Instructions", draft-xu-mpls-service-chaining-03 (work in progress), June 2017.
- [I-D.xu-mpls-spring-islands-connection-over-ip]  
Xu, X., Raszuk, R., Chunduri, U., Contreras, L., and L. Jalil, "Connecting MPLS-SPRING Islands over IP Networks", draft-xu-mpls-spring-islands-connection-over-ip-00 (work in progress), October 2016.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <<https://www.rfc-editor.org/info/rfc4817>>.

- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black,  
"Encapsulating MPLS in UDP", RFC 7510,  
DOI 10.17487/RFC7510, April 2015,  
<<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function  
Chaining (SFC) Architecture", RFC 7665,  
DOI 10.17487/RFC7665, October 2015,  
<<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC7684] Psenak, P., Gredler, H., Shakir, R., Henderickx, W.,  
Tantsura, J., and A. Lindem, "OSPFv2 Prefix/Link Attribute  
Advertisement", RFC 7684, DOI 10.17487/RFC7684, November  
2015, <<https://www.rfc-editor.org/info/rfc7684>>.
- [RFC7794] Ginsberg, L., Ed., Decraene, B., Previdi, S., Xu, X., and  
U. Chunduri, "IS-IS Prefix Attributes for Extended IPv4  
and IPv6 Reachability", RFC 7794, DOI 10.17487/RFC7794,  
March 2016, <<https://www.rfc-editor.org/info/rfc7794>>.
- [RFC7981] Ginsberg, L., Previdi, S., and M. Chen, "IS-IS Extensions  
for Advertising Router Information", RFC 7981,  
DOI 10.17487/RFC7981, October 2016,  
<<https://www.rfc-editor.org/info/rfc7981>>.

## Authors' Addresses

Xiaohu Xu  
Huawei

Email: [xuxiaohu@huawei.com](mailto:xuxiaohu@huawei.com)

Ahmed Bashandy  
Cisco

Email: [bashandy@cisco.com](mailto:bashandy@cisco.com)

Hamid Assarpour  
Broadcom

Email: [hamid.assarpour@broadcom.com](mailto:hamid.assarpour@broadcom.com)

Shaowen Ma  
Juniper

Email: mashao@juniper.net

Wim Henderickx  
Nokia

Email: wim.henderickx@nokia.com

Jeff Tantsura  
Individual

Email: jefftant@gmail.com

CCAMP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 9, 2017

X. Zhang  
K. Xiang  
Huawei Technologies  
A. Sharma  
Infinera  
X. Liu  
Ericsson  
March 08, 2017

A YANG Data Model for Optical Transport Network Topology  
draft-zhang-ccamp-11-topo-yang-06

Abstract

A transport network is a server-layer network designed to provide connectivity services for a client-layer network to carry the client traffic transparently across the server-layer network resources. A transport network can be constructed from equipments utilizing any of a number of different transport technologies such as the evolving Optical Transport Networks (OTN) or packet transport as provided by the MPLS-Transport Profile (MPLS-TP).

This draft describes a YANG data model to describe the topologies of an Optical Transport Network (OTN). It is independent of control plane protocols and captures topological and resource related information pertaining to OTN. This model enables clients, which interact with a transport domain controller via a REST interface, for OTN topology related operations such as obtaining the relevant topology resource information.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 9, 2017.

#### Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. Introduction . . . . .                                 | 2  |
| 2. Terminology and Notations . . . . .                    | 3  |
| 3. YANG Data Model for OTN Topology . . . . .             | 4  |
| 3.1. the YANG Tree . . . . .                              | 4  |
| 3.2. Explanation of the OTN Topology Data Model . . . . . | 5  |
| 3.3. The YANG Code . . . . .                              | 6  |
| 4. IANA Considerations . . . . .                          | 13 |
| 5. Manageability Considerations . . . . .                 | 13 |
| 6. Security Considerations . . . . .                      | 13 |
| 7. Acknowledgements . . . . .                             | 13 |
| 8. Contributors . . . . .                                 | 13 |
| 9. References . . . . .                                   | 14 |
| 9.1. Normative References . . . . .                       | 14 |
| 9.2. Informative References . . . . .                     | 15 |
| Authors' Addresses . . . . .                              | 15 |

#### 1. Introduction

A transport network is a server-layer network designed to provide connectivity services for a client-layer network to carry the client traffic transparently across the server-layer network resources. A transport network can be constructed of equipments utilizing any of a number of different transport technologies such as the Optical

Transport Networks (OTN) or packet transport as provided by the MPLS-Transport Profile (MPLS-TP).

This document defines a data model of an OTN network topology, using YANG [RFC6020]. The model can be used by an application exposing to a transport controller via a REST interface. Furthermore, it can be used by an application for the following purposes (but not limited to):

- o To obtain a whole view of the network topology information of its interest;
- o To receive notifications with regard to the information change of the OTN topology;
- o To enforce the establishment and update of a network topology with the characteristic specified in the data model, e.g., by a client controller;

The YANG model defined in this draft is independent of control plane protocols and captures topology related information pertaining to an Optical Transport Networks (OTN)-electrical layer, as the scope specified by [RFC7062] and [RFC7138]. Furthermore, it is not a stand-alone model, but augmenting from the TE topology YANG model defined in [I-D.ietf-teas-yang-te-topo].

Optical network technologies, including fixed Dense Wavelength Switched Optical Network (WSON) and flexible optical networks (a.k.a., flexi-grid networks), are covered in [I-D.ietf-ccamp-wson-yang] and [I-D.vergara-ccamp-flexigrid-yang], respectively.

## 2. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [I-D.ietf-netmod-rfc6087bis]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.

- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

### 3. YANG Data Model for OTN Topology

#### 3.1. the YANG Tree

```

module: ietf-otn-topology
augment /nd:networks/nd:network/nd:network-types/tet:te-topology:
 +--rw otn-topology!
augment /nd:networks/nd:network:
 +--rw name? string
augment /nd:networks/nd:network/nd:node:
 +--rw name? string
augment /nd:networks/nd:network/lnk:link/tet:te/tet:config:
 +--rw available-odu-info* [priority]
 | +--rw priority uint8
 | +--rw odulist* [odu-type]
 | +--rw odu-type identityref
 | +--rw number? uint16
 +--rw distance? uint32
augment /nd:networks/nd:network/lnk:link/tet:te/tet:state:
 +--ro available-odu-info* [priority]
 | +--ro priority uint8
 | +--ro odulist* [odu-type]
 | +--ro odu-type identityref
 | +--ro number? uint16
 +--ro distance? uint32
augment /nd:networks/nd:network/nd:node/lnk:termination-point
/tet:te/tet:config:
 +--rw client-facing? empty
 +--rw tpn? uint16
 +--rw tsg? identityref
 +--rw protocol-type? identityref
 +--rw adaptation-type? adaptation-type
 +--rw sink-adapt-active? boolean
 +--rw source-adapt-active? boolean
 +--rw tributary-slots
 | +--rw values* uint8
 +--rw supported-payload-types* [index]
 | +--rw index uint16
 | +--rw payload-type? string
augment /nd:networks/nd:network/nd:node/lnk:termination-point/

```

```

tet:te/tet:state:
 +--ro client-facing? empty
 +--ro tpn? uint16
 +--ro tsng? identityref
 +--ro protocol-type? identityref
 +--ro adaptation-type? adaptation-type
 +--ro sink-adapt-active? boolean
 +--ro source-adapt-active? boolean
 +--ro tributary-slots
 | +--ro values* uint8
 +--ro supported-payload-types* [index]
 +--ro index uint16
 +--ro payload-type? string

```

### 3.2. Explanation of the OTN Topology Data Model

As can be seen, from the data tree shown in Section 3.1, the YANG module presented in this draft augments from a more generic Traffic Engineered (TE) network topology data model, i.e., the `ietf-te-topology.yang` as specified in [I-D.ietf-teas-yang-te-topo]. The entities and their attributes, such as node, termination points and links, are still applicable for describing an OTN topology and the model presented in this draft only specifies with technology-specific attributes/information. For example, if the data plane complies with ITU-T G.709 (2012) standards, the switching-capability and encoding attributes MUST be filled as OTN-TDM and G.709 ODUk(Digital Path) respectively.

Note the model in this draft re-uses some attributes defined in `ietf-transport-types.yang`, which is specified in [I-D.sharma-ccamp-otn-tunnel-model].

One of the main augmentations in this model is that it allows to specify the type of ODU container and the number a link can support per priority level. For example, for a ODU3 link, it may advertise 32\*ODU0, 16\*ODU1, 4\*ODU2 available, assuming only a single priority level is supported. If one of ODU2 resource is taken to establish a ODU path, then the availability of this ODU link is updated as 24\*ODU0, 12\*ODU1, 3\*ODU2 available. If there are equipment hardware limitations, then a subset of potential ODU type SHALL be advertised. For instance, an ODU3 link may only support 4\*ODU2.



### 3.3. The YANG Code

```
<CODE BEGINS> file " ietf-otn-topology@2017-03-08.yang"

module ietf-otn-topology {
 yang-version 1.1;

 namespace "urn:ietf:params:xml:ns:yang:ietf-otn-topology";
 prefix "otntopo";

 import ietf-network {
 prefix "nd";
 }

 import ietf-network-topology {
 prefix "lnk";
 }

 import ietf-te-topology {
 prefix "tet";
 }

 import ietf-transport-types {
 prefix "tran-types";
 }

 organization
 "Internet Engineering Task Force (IETF) CCAMP WG";
 contact
 "
 WG List: <mailto:ccamp@ietf.org>

 ID-draft editor:
 Xian ZHANG (zhang.xian@huawei.com);
 Anurag Sharma (AnSharma@infinera.com);
 ";

 description
 "This module defines a protocol independent Layer 1/ODU
 topology data model.";

 revision 2017-03-08 {
 description
 "Revision 0.2";
 reference
 "draft-zhang-ccamp-l1-topo-yang-05.txt";
 }
}
```

```
/*
typedef
*/

typedef adaptation-type {
 type enumeration {
 enum CBR {
 description "Constant Bit Rate.";
 }
 enum ATMvp {
 description "ATM VP.";
 }
 enum GFP {
 description "Generic Framing Procedure.";
 }
 enum NULL {
 description "NULL";
 }
 enum PRBS {
 description "Pseudo Random Binary Sequence";
 }
 enum RSn {
 description "SDH/SONET section";
 }
 enum ODUj-21 {
 description "ODU payload type 21";
 }
 enum ETHERNET_PP-OS {
 description "ETHERNET_PP-OS, for ODU 2 only";
 }
 enum CBRx {
 description "CBRx(0.. 1.25G), for ODU0 only";
 }
 enum ODU {
 description "Optical Data Unit";
 }
 }
 description
 "Defines a type representing the adaptation type
 on the termination point.";
}

/*
Groupings
*/
```

```
grouping otn-topology-type {
 container otn-topology {
 presence "indicates a topology type of Optical
 Transport Network (OTN)-electrical layer.";
 description "otn topology type";
 }
 description "otn-topology-type";
}

grouping otn-topology-attributes {
 leaf name {
 type string;
 description "the topology name";
 }
 description "name attribute for otn topology";
}

grouping otn-node-attributes {
 description "otn-node-attributes";
 leaf name {
 type string;
 description "a name for this node.";
 }
}

grouping otn-link-attributes {
 description "otn link attributes";

 list available-odu-info{
 key "priority";
 max-elements "8";

 description
 "List of ODU type and number on this link";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "priority";
 }
 }

 list odulist {
 key "odu-type";

 description
 "the list of available ODUs per priority level";

 leaf odu-type {
```

```
 type identityref {
 base tran-types:tributary-protocol-type;
 }
 description "the type of ODU";
 }

 leaf number {
 type uint16;
 description "the number of odu type supported";
 }
}

leaf distance {
 type uint32;
 description "distance in the unit of kilometers";
}

}

grouping otn-tp-attributes {
 description "otn-tp-attributes";

 leaf client-facing {
 type empty;
 description
 "if present, it means this tp is a client-facing tp.
 adding/dropping client signal flow.";
 }

 leaf tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of Evolving
 G.709 Optical Transport Networks.";
 }

 leaf tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description "Tributary slot granularity.";
 reference
 "G.709/Y.1331, February 2012: Interfaces for the Optical
 Transport Network (OTN)";
 }
}
```

```
}

leaf protocol-type {
 type identityref {
 base tran-types:tributary-protocol-type;
 }
 description "Protocol type for the Termination Point.";
}

leaf adaptation-type {
 type adaptation-type;
 description
 "This attribute indicates the type of the supported
 adaptation function at the termination point.";
 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for the
 network element view.";
}

leaf sink-adapt-active {
 type boolean;
 description
 "This attribute allows for activation or deactivation of
 the sink adaptation function. The value of TRUE means active.";

 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for the
 network element view ";
}

leaf source-adapt-active {
 type boolean;
 description
 "This attribute allows for activation or deactivation of
 the sink adaptation function. The value of TRUE
 means active.";
 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for
 the network element view ";
}

container tributary-slots {
 description
 "A list of tributary slots used by the ODU
 Termination Point.";
```

```
 leaf-list values {
 type uint8;
 description
 "Tributary slot value.";
 reference
 "G.709/Y.1331, February 2012: Interfaces for the
 Optical Transport Network (OTN)";
 }
 }

 list supported-payload-types{
 key "index";

 description "supported payload types of a TP";

 leaf index {
 type uint16;
 description "payload type index";
 }

 leaf payload-type {
 type string;
 description "the payload type supported by this client
 tp";
 reference
 "http://www.iana.org/assignments/gmpls-sig-parameters
 /gmpls-sig-parameters.xhtml
 not: the payload type is defined as the generalized PIDs
 in GMPLS.";
 }
 }
}

/*
 * Data nodes
 */
augment "/nd:networks/nd:network/nd:network-types/tet:te-topology" {
 uses otn-topology-type;
 description "augment network types to include otn newtork";
}

augment "/nd:networks/nd:network" {
 when "nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 uses otn-topology-attributes;
 description "Augment network configuration";
}
```

```
}

augment "/nd:networks/nd:network/nd:node" {
 when "../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 description "Augment node configuration";
 uses otn-node-attributes;
}

augment "/nd:networks/nd:network/lnk:link/tet:te/tet:config" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network.";
 }
 description "Augment link configuration";

 uses otn-link-attributes;
}

augment "/nd:networks/nd:network/lnk:link/tet:te/tet:state" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network.";
 }
 description "Augment link state";

 uses otn-link-attributes;
}

augment "/nd:networks/nd:network/nd:node/"
+ "lnk:termination-point/tet:te/tet:config" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 description "OTN TP attributes config in a ODU topology.";
 uses otn-tp-attributes;
}

augment "/nd:networks/nd:network/nd:node/"
+ "lnk:termination-point/tet:te/tet:state" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 description "OTN TP attributes state in a ODU topology.";
 uses otn-tp-attributes;
}
}

<CODE ENDS>
```

#### 4. IANA Considerations

TBD.

#### 5. Manageability Considerations

TBD.

#### 6. Security Considerations

The data following the model defined in this draft is exchanged via, for example, the interface between an orchestrator and a transport network controller. The security concerns mentioned in [I-D.ietf-teas-yang-te-topo] for using ietf-te-topology.yang model also applies to this draft.

The YANG module defined in this document can be accessed via the RESTCONF protocol defined in [I-D.ietf-netconf-restconf], or maybe via the NETCONF protocol [RFC6241].

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., POST) to these data nodes without proper protection can have a negative effect on network operations.

Editors note: to list specific subtrees and data nodes and their sensitivity/vulnerability.

#### 7. Acknowledgements

We would like to thank Igor Bryskin, Zhe Liu, Dieter Beller and Daniele Ceccarelli for their comments and discussions.

#### 8. Contributors

Baoquan Rao  
Huawei Technologies  
Email: raobaoquan@huawei.com

Sergio Belotti  
Alcatel Lucent  
Email: Sergio.belotti@alcatel-lucent.com

Huub van Helvoort  
Hai Gaoming BV  
the Netherlands



Email: huubatwork@gmail.com

## 9. References

### 9.1. Normative References

- [I-D.ietf-netconf-restconf]  
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-18 (work in progress), October 2016.
- [I-D.ietf-netmod-rfc6087bis]  
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-12 (work in progress), March 2017.
- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-06 (work in progress), October 2016.
- [I-D.sharma-ccamp-otn-tunnel-model]  
ansharma@infinera.com, a., Rao, R., and X. Zhang, "OTN Tunnel YANG Model", draft-sharma-ccamp-otn-tunnel-model-00 (work in progress), October 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC7062] Zhang, F., Ed., Li, D., Li, H., Belotti, S., and D. Ceccarelli, "Framework for GMPLS and PCE Control of G.709 Optical Transport Networks", RFC 7062, DOI 10.17487/RFC7062, November 2013, <<http://www.rfc-editor.org/info/rfc7062>>.

[RFC7138] Ceccarelli, D., Ed., Zhang, F., Belotti, S., Rao, R., and J. Drake, "Traffic Engineering Extensions to OSPF for GMPLS Control of Evolving G.709 Optical Transport Networks", RFC 7138, DOI 10.17487/RFC7138, March 2014, <<http://www.rfc-editor.org/info/rfc7138>>.

## 9.2. Informative References

[I-D.ietf-ccamp-wson-yang]  
Lee, Y., Dhody, D., Zhang, X., Guo, A., Lopezalvarez, V., King, D., and B. Yoon, "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang-05 (work in progress), February 2017.

[I-D.vergara-ccamp-flexigrid-yang]  
Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid Optical Networks", draft-vergara-ccamp-flexigrid-yang-04 (work in progress), March 2017.

## Authors' Addresses

Xian Zhang  
Huawei Technologies  
F3-5-B R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China  
  
Email: [zhang.xian@huawei.com](mailto:zhang.xian@huawei.com)

Kun Xiang  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China  
  
Email: [xiangkun@huawei.com](mailto:xiangkun@huawei.com)

Anurag Sharma  
Infinera  
  
Email: [ansharma@infinera.com](mailto:ansharma@infinera.com)

Xufeng Liu  
Ericsson

Email: xufeng.liu@ericsson.com

CCAMP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: October 27, 2017

H. Zheng  
Z. Fan  
Huawei Technologies  
A. Sharma  
Google  
X. Liu  
Jabil  
April 25, 2017

A YANG Data Model for Optical Transport Network Topology  
draft-zhang-ccamp-11-topo-yang-07

Abstract

A transport network is a server-layer network designed to provide connectivity services for a client-layer network to carry the client traffic transparently across the server-layer network resources. A transport network can be constructed from equipments utilizing any of a number of different transport technologies such as the evolving Optical Transport Networks (OTN) or packet transport as provided by the MPLS-Transport Profile (MPLS-TP).

This draft describes a YANG data model to describe the topologies of an Optical Transport Network (OTN). It is independent of control plane protocols and captures topological and resource related information pertaining to OTN. This model enables clients, which interact with a transport domain controller via a REST interface, for OTN topology related operations such as obtaining the relevant topology resource information.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 27, 2017.

#### Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Table of Contents

1. Introduction . . . . .	2
2. Terminology and Notations . . . . .	3
3. YANG Data Model for OTN Topology . . . . .	4
3.1. the YANG Tree . . . . .	4
3.2. Explanation of the OTN Topology Data Model . . . . .	5
3.3. The YANG Code . . . . .	5
4. IANA Considerations . . . . .	13
5. Manageability Considerations . . . . .	13
6. Security Considerations . . . . .	13
7. Acknowledgements . . . . .	13
8. Contributors . . . . .	13
9. References . . . . .	14
9.1. Normative References . . . . .	14
9.2. Informative References . . . . .	15
Authors' Addresses . . . . .	15

#### 1. Introduction

A transport network is a server-layer network designed to provide connectivity services for a client-layer network to carry the client traffic transparently across the server-layer network resources. A transport network can be constructed of equipments utilizing any of a number of different transport technologies such as the Optical

Transport Networks (OTN) or packet transport as provided by the MPLS-Transport Profile (MPLS-TP).

This document defines a data model of an OTN network topology, using YANG [RFC6020]. The model can be used by an application exposing to a transport controller via a REST interface. Furthermore, it can be used by an application for the following purposes (but not limited to):

- o To obtain a whole view of the network topology information of its interest;
- o To receive notifications with regard to the information change of the OTN topology;
- o To enforce the establishment and update of a network topology with the characteristic specified in the data model, e.g., by a client controller;

The YANG model defined in this draft is independent of control plane protocols and captures topology related information pertaining to an Optical Transport Networks (OTN)-electrical layer, as the scope specified by [RFC7062] and [RFC7138]. Furthermore, it is not a stand-alone model, but augmenting from the TE topology YANG model defined in [I-D.ietf-teas-yang-te-topo].

Optical network technologies, including fixed Dense Wavelength Switched Optical Network (WSON) and flexible optical networks (a.k.a., flexi-grid networks), are covered in [I-D.ietf-ccamp-wson-yang] and [I-D.vergara-ccamp-flexigrid-yang], respectively.

## 2. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [I-D.ietf-netmod-rfc6087bis]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.

- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

### 3. YANG Data Model for OTN Topology

#### 3.1. the YANG Tree

```

module: ietf-otn-topology
augment /nd:networks/nd:network/nd:network-types/tet:te-topology:
 +--rw otn-topology!
augment /nd:networks/nd:network:
 +--rw name? string
augment /nd:networks/nd:network/nd:node:
 +--rw name? string
augment /nd:networks/nd:network/lnk:link/tet:te/tet:config:
 +--rw available-odu-info* [priority]
 | +--rw priority uint8
 | +--rw odulist* [odu-type]
 | +--rw odu-type identityref
 | +--rw number? uint16
 +--rw distance? uint32
augment /nd:networks/nd:network/lnk:link/tet:te/tet:state:
 +--ro available-odu-info* [priority]
 | +--ro priority uint8
 | +--ro odulist* [odu-type]
 | +--ro odu-type identityref
 | +--ro number? uint16
 +--ro distance? uint32
augment /nd:networks/nd:network/nd:node/lnk:termination-point/
tet:te/tet:config:
 +--rw client-facing? empty
 +--rw tpn? uint16
 +--rw tsg? identityref
 +--rw protocol-type? identityref
 +--rw adaptation-type? adaptation-type
 +--rw sink-adapt-active? boolean
 +--rw source-adapt-active? boolean
 +--rw tributary-slots
 | +--rw values* uint8
 +--rw supported-payload-types* [index]
 | +--rw index uint16
 | +--rw payload-type? string
augment /nd:networks/nd:network/nd:node/lnk:termination-point/
tet:te/tet:state:

```

```

+--ro client-facing? empty
+--ro tpn? uint16
+--ro tsq? identityref
+--ro protocol-type? identityref
+--ro adaptation-type? adaptation-type
+--ro sink-adapt-active? boolean
+--ro source-adapt-active? boolean
+--ro tributary-slots
| +--ro values* uint8
+--ro supported-payload-types* [index]
 +--ro index uint16
 +--ro payload-type? string

```

### 3.2. Explanation of the OTN Topology Data Model

As can be seen, from the data tree shown in Section 3.1, the YANG module presented in this draft augments from a more generic Traffic Engineered (TE) network topology data model, i.e., the `ietf-te-topology.yang` as specified in [I-D.ietf-teas-yang-te-topol]. The entities and their attributes, such as node, termination points and links, are still applicable for describing an OTN topology and the model presented in this draft only specifies with technology-specific attributes/information. For example, if the data plane complies with ITU-T G.709 (2012) standards, the switching-capability and encoding attributes MUST be filled as OTN-TDM and G.709 ODUk(Digital Path) respectively.

Note the model in this draft re-uses some attributes defined in `ietf-transport-types.yang`, which is specified in [I-D.sharma-ccamp-otn-tunnel-model].

One of the main augmentations in this model is that it allows to specify the type of ODU container and the number a link can support per priority level. For example, for a ODU3 link, it may advertise 32\*ODU0, 16\*ODU1, 4\*ODU2 available, assuming only a single priority level is supported. If one of ODU2 resource is taken to establish a ODU path, then the availability of this ODU link is updated as 24\*ODU0, 12\*ODU1, 3\*ODU2 available. If there are equipment hardware limitations, then a subset of potential ODU type SHALL be advertised. For instance, an ODU3 link may only support 4\*ODU2.

### 3.3. The YANG Code

```

<CODE BEGINS> file "ietf-otn-topology@2017-04-25.yang"

module ietf-otn-topology {

```



```
yang-version 1.1;

namespace "urn:ietf:params:xml:ns:yang:ietf-otn-topology";
prefix "otntopo";

import ietf-network {
 prefix "nd";
}

import ietf-network-topology {
 prefix "lnk";
}

import ietf-te-topology {
 prefix "tet";
}

import ietf-transport-types {
 prefix "tran-types";
}

organization
 "Internet Engineering Task Force (IETF) CCAMP WG";
contact
 "
 WG List: <mailto:ccamp@ietf.org>

 ID-draft editor:
 Haomian Zheng (zhenghaomian@huawei.com);
 Zheyu Fan (fanzheyu2@huawei.com);
 Anurag Sharma (ansha@google.com);
 Xufeng Liu (Xufeng_Liu@jabil.com)
 ";

description
 "This module defines a protocol independent Layer 1/ODU
 topology data model.";

revision 2017-04-25 {
 description
 "Revision 0.3";
 reference
 "draft-zhang-ccamp-l1-topo-yang-07.txt";
}

/*
typedef
*/
```

```
typedef adaptation-type {
 type enumeration {
 enum CBR {
 description "Constant Bit Rate.";
 }
 enum ATMvp {
 description "ATM VP.";
 }
 enum GFP {
 description "Generic Framing Procedure.";
 }
 enum NULL {
 description "NULL";
 }
 enum PRBS {
 description "Pseudo Random Binary Sequence";
 }
 enum RSn {
 description "SDH/SONET section";
 }
 enum ODUj-21 {
 description "ODU payload type 21";
 }
 enum ETHERNET_PP-OS {
 description "ETHERNET_PP-OS, for ODU 2 only";
 }
 enum CBRx {
 description "CBRx(0.. 1.25G), for ODU0 only";
 }
 enum ODU {
 description "Optical Data Unit";
 }
 }

 description
 "Defines a type representing the adaptation type
 on the termination point.";
}

/*
Groupings
*/

grouping otn-topology-type {
 container otn-topology {
 presence "indicates a topology type of Optical
 Transport Network (OTN)-electrical layer.";
 }
}
```

```
 description "otn topology type";
 }
 description "otn-topology-type";
}

grouping otn-topology-attributes {
 leaf name {
 type string;
 description "the topology name";
 }
 description "name attribute for otn topology";
}

grouping otn-node-attributes {
 description "otn-node-attributes";
 leaf name {
 type string;
 description "a name for this node.";
 }
}

grouping otn-link-attributes {
 description "otn link attributes";

 list available-odu-info{
 key "priority";
 max-elements "8";

 description
 "List of ODU type and number on this link";
 leaf priority {
 type uint8 {
 range "0..7";
 }
 description "priority";
 }
 }

 list odulist {
 key "odu-type";

 description
 "the list of available ODUs per priority level";

 leaf odu-type {
 type identityref{
 base tran-types:tributary-protocol-type;
 }
 description "the type of ODU";
 }
 }
}
```

```
 }

 leaf number {
 type uint16;
 description "the number of odu type supported";
 }
 }

 leaf distance {
 type uint32;
 description "distance in the unit of kilometers";
 }
}

grouping otn-tp-attributes {
 description "otn-tp-attributes";

 leaf client-facing {
 type empty;
 description
 "if present, it means this tp is a client-facing tp.
 adding/dropping client signal flow.";
 }

 leaf tpn {
 type uint16 {
 range "0..4095";
 }
 description
 "Tributary Port Number. Applicable in case of mux services.";
 reference
 "RFC7139: GMPLS Signaling Extensions for Control of Evolving
 G.709 Optical Transport Networks.";
 }

 leaf tsg {
 type identityref {
 base tran-types:tributary-slot-granularity;
 }
 description "Tributary slot granularity.";
 reference
 "G.709/Y.1331, February 2012: Interfaces for the Optical
 Transport Network (OTN)";
 }

 leaf protocol-type {
 type identityref {
```

```
 base tran-types:tributary-protocol-type;
 }
 description "Protocol type for the Termination Point.";
}

leaf adaptation-type {
 type adaptation-type;
 description
 "This attribute indicates the type of the supported
 adaptation function at the termination point.";
 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for the
 network element view.";
}

leaf sink-adapt-active {
 type boolean;
 description
 "This attribute allows for activation or deactivation of
 the sink adaptation function. The value of TRUE means active.";

 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for the
 network element view ";
}

leaf source-adapt-active {
 type boolean;
 description
 "This attribute allows for activation or deactivation of
 the sink adaptation function. The value of TRUE
 means active.";
 reference
 "G.874.1, January 2002: Optical transport network (OTN):
 Protocol-neutral management information model for
 the network element view ";
}

container tributary-slots {
 description
 "A list of tributary slots used by the ODU
 Termination Point.";
 leaf-list values {
 type uint8;
 description
 "Tributary slot value.";
 }
}
```

```
 reference
 "G.709/Y.1331, February 2012: Interfaces for the
 Optical Transport Network (OTN)";
 }
}

list supported-payload-types{
 key "index";

 description "supported payload types of a TP";

 leaf index {
 type uint16;
 description "payload type index";
 }

 leaf payload-type {
 type string;
 description "the payload type supported by this client
 tp";
 reference
 "http://www.iana.org/assignments/gmpls-sig-parameters
 /gmpls-sig-parameters.xhtml
 not: the payload type is defined as the generalized PIDs
 in GMPLS.";
 }
}

/*
* Data nodes
*/
augment "/nd:networks/nd:network/nd:network-types/tet:te-topology" {
 uses otn-topology-type;
 description "augment network types to include otn newtork";
}

augment "/nd:networks/nd:network" {
 when "nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 uses otn-topology-attributes;
 description "Augment network configuration";
}

augment "/nd:networks/nd:network/nd:node" {
 when "../nd:network-types/tet:te-topology/otn-topology" {
```

```
 description "Augment only for otn network";
 }
 description "Augment node configuration";
 uses otn-node-attributes;
}

augment "/nd:networks/nd:network/lnk:link/tet:te/tet:config" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network.";
 }
 description "Augment link configuration";

 uses otn-link-attributes;
}

augment "/nd:networks/nd:network/lnk:link/tet:te/tet:state" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network.";
 }
 description "Augment link state";

 uses otn-link-attributes;
}

augment "/nd:networks/nd:network/nd:node/"
 + "lnk:termination-point/tet:te/tet:config" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 description "OTN TP attributes config in a ODU topology.";
 uses otn-tp-attributes;
}

augment "/nd:networks/nd:network/nd:node/"
 + "lnk:termination-point/tet:te/tet:state" {
 when "../../../nd:network-types/tet:te-topology/otn-topology" {
 description "Augment only for otn network";
 }
 description "OTN TP attributes state in a ODU topology.";
 uses otn-tp-attributes;
}
}

<CODE ENDS>
```

#### 4. IANA Considerations

TBD.

#### 5. Manageability Considerations

TBD.

#### 6. Security Considerations

The data following the model defined in this draft is exchanged via, for example, the interface between an orchestrator and a transport network controller. The security concerns mentioned in [I-D.ietf-teas-yang-te-topo] for using ietf-te-topology.yang model also applies to this draft.

The YANG module defined in this document can be accessed via the RESTCONF protocol defined in [RFC8040], or maybe via the NETCONF protocol [RFC6241].

There are a number of data nodes defined in the YANG module which are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., POST) to these data nodes without proper protection can have a negative effect on network operations.

Editors note: to list specific subtrees and data nodes and their sensitivity/vulnerability.

#### 7. Acknowledgements

We would like to thank Igor Bryskin, Zhe Liu, Dieter Beller and Daniele Ceccarelli for their comments and discussions.

#### 8. Contributors

Baoquan Rao  
Huawei Technologies  
Email: raobaoquan@huawei.com

Xian Zhang  
Huawei Technologies  
Email: zhang.xian@huawei.com

Sergio Belotti  
Nokia  
Email: sergio.belotti@nokia.com



Huub van Helvoort  
Hai Gaoming BV  
the Netherlands  
Email: huubatwork@gmail.com

## 9. References

### 9.1. Normative References

- [I-D.ietf-teas-yang-te-topo]  
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-08 (work in progress), March 2017.
- [I-D.sharma-ccamp-otn-tunnel-model]  
Zhang, X., xiangkun, x., ansharma@infinera.com, a., and R. Rao, "OTN Tunnel YANG Model", draft-sharma-ccamp-otn-tunnel-model-01 (work in progress), March 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC7138] Ceccarelli, D., Ed., Zhang, F., Belotti, S., Rao, R., and J. Drake, "Traffic Engineering Extensions to OSPF for GMPLS Control of Evolving G.709 Optical Transport Networks", RFC 7138, DOI 10.17487/RFC7138, March 2014, <<http://www.rfc-editor.org/info/rfc7138>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<http://www.rfc-editor.org/info/rfc8040>>.

## 9.2. Informative References

- [I-D.ietf-ccamp-wson-yang]  
Lee, Y., Dhody, D., Zhang, X., Guo, A., Lopezalvarez, V., King, D., and B. Yoon, "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang-05 (work in progress), February 2017.
- [I-D.ietf-netmod-rfc6087bis]  
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-12 (work in progress), March 2017.
- [I-D.vergara-ccamp-flexigrid-yang]  
Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid Optical Networks", draft-vergara-ccamp-flexigrid-yang-04 (work in progress), March 2017.
- [RFC7062] Zhang, F., Ed., Li, D., Li, H., Belotti, S., and D. Ceccarelli, "Framework for GMPLS and PCE Control of G.709 Optical Transport Networks", RFC 7062, DOI 10.17487/RFC7062, November 2013, <<http://www.rfc-editor.org/info/rfc7062>>.

## Authors' Addresses

Haomian Zheng  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Zheyu Fan  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District  
Shenzhen, Guangdong 518129  
P.R.China

Email: [fanzheyu2@huawei.com](mailto:fanzheyu2@huawei.com)

Anurag Sharma  
Google  
1600 Amphitheatre Parkway  
Mountain View, CA 94043

Email: [ansha@google.com](mailto:ansha@google.com)

Xufeng Liu  
Jabil

Email: [Xufeng\\_Liu@jabil.com](mailto:Xufeng_Liu@jabil.com)