

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

I. Bryskin
Huawei Technologies
X. Liu
Jabil
V. Beeram
Juniper Networks
T. Saad
Cisco Systems Inc
March 13, 2017

ONF/T-API Services vs. IETF/YANG Models and Interfaces
draft-bryskin-teas-yang-ietf-vs-onf-00

Abstract

This document compares IETF YANG TE (Traffic Engineering) data model and ONF/T-API 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 <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. Topology Service	3
2.1. Constrained Nodes	3
2.2. Intra-node Metrics	5
2.3. Topology Updates	6
2.4. Topology Telemetry Collection	9
2.5. Topology Name/Address Spaces	10
2.6. Topology Relationships	12
2.7. Topology Attributes	15
2.8. Topology Service Relationships with Other Services	16
2.9. Topology Negotiation and (Re-)configuration	16
2.10. Integration with IP/MPLS	18
3. Connectivity Service	18
3.1. Connectivity Service Protection	19
3.2. Hierarchical Connectivity Service	21
3.3. Connectivity Service Re-optimization	24
3.4. Connectivity Service Templates	24
3.5. Connectivity Service Attribute Change Update Notifications and Telemetry Streaming	24
3.6. Connectivity Scheduling	25
3.7. Potential Connectivity Service	25
4. Path Computation Service	26
5. Virtual Network Service	27
6. Data Modeling Language	28
7. Security Framework	29
8. IANA Considerations	30
9. Security Considerations	30
10. Acknowledgements	30
11. References	30
11.1. Normative References	30
11.2. Informative References	31
Authors' Addresses	31

1. Introduction

The success of T-SDN as an architecture depends to a large degree on the quality and widespread adoption of open standardized interfaces to/from T-SDN controllers, linking them flexibly into various hierarchies and confederations. Currently, the two most popular such interfaces are:

1. T-API developed by ONF;

2. RESTCONF/YANG [RFC7950] based on TE Topology and TE Tunnels models defined in [I-D.ietf-teas-yang-te-topo] and [I-D.ietf-teas-yang-te] documents respectively, the product of IETF TEAS WG.

The two interfaces have the close attention of network operators and vendors. There is a lot of confusion about their respective technical merits and "marketing" strengths, applications they can support, use cases they cover, and so on. Do they compete or could they somehow complement each other?

This memo is limited to a strictly technical comparison with the special focus on the models supporting the two interfaces, in particular, the semantics, relationships, informational flows and services they define. Our analysis suggests that the IETF models provide for implementation of powerful hierarchical T-SDN controller systems, supporting a broad range of client systems and use cases, and that in some identifiable respects, T-API appears to fall relatively shorter. This memo is largely organized around considering the identified "gaps".

2. Topology Service

2.1. Constrained Nodes

The T-API Topology service does not support the notion of blocking/constrained nodes. This means that if a T-API Topology service provider exposes to a client a topology with at least one node with constrained connectivity, e.g. the node can switch a potential TE path/connection, say, from interface (NodeEdge point) A to B, but not from A to C; there is no way for the provider to communicate the connection limitations to the client, thus making the provided TE topology unfit for the client's path computations. This is a serious issue because many transport physical switches and virtually all abstract composite nodes should be treated as blocking nodes.

Likewise, if a potential path source/destination node is constrained in such a way that the path may leave/enter the source/destination node over a link from a subset of (but not all) same-layer links connected to the node, the T-API Topology service provider has no way of communicating such a circumstance to the client.

The described issue is addressed in the IETF TE Topology model. A TE node's Connectivity Matrix attribute (Figure 1) fully describes the node's TE path/connection switching limitations, while a TE Tunnel Termination Point's (TTP's) Local Link Connectivity List attribute (Figure 2) describes the node's TE path/connection termination limitations with respect to each TTP hosted by the node in question.

Basic Connectivity Matrix:

```

LTP-6/label-x <=> LTP-1/label-y
LTP-5/label-x <=> LTP-2/label-y
LTP-5/label-x <=> LTP-4/label-y
LTP-4/label-x <=> LTP-1/label-y
LTP-3/label-x <=> LTP-2/label-y
...
```

Detailed Connectivity Matrix:

```

LTP-6/label-x <=> LTP-1/label-y
    (Cost c, Delay d, SRLB s, ...)
LTP-5/label-x <=> LTP-2/label-y
    (Cost c, Delay d, SRLB s, ...)
LTP-5/label-x <=> LTP-4/label-y
    (Cost c, Delay d, SRLB s, ...)
LTP-4/label-x <=> LTP-1/label-y
    (Cost c, Delay d, SRLB s, ...)
LTP-3/label-x <=> LTP-2/label-y
...
```

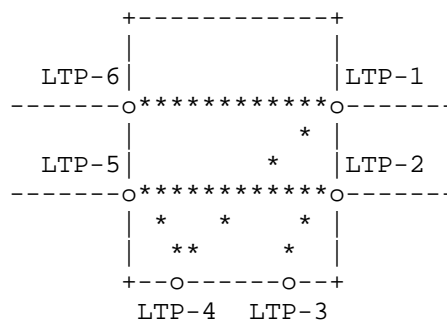


Figure 1: TE Node Connectivity Matrix

TTP-1 Basic LLCL:

```
TTP-1 <=> {LTP-5/label-x,
            LTP-2/label-y}
```

TTP-1 Detailed LLCL:

```
TTP-1 <=> {
    LTP-5/label-x,
    (Cost c, Delay d, SRLB s, ...),
    LTP-2/label-y,
    (Cost c, Delay d, SRLB s, ...)
}
```

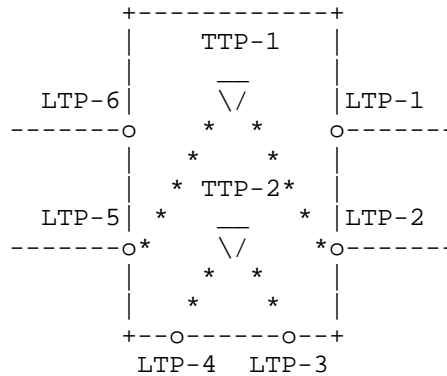


Figure 2: TTP Local Link Connectivity List

2.2. Intra-node Metrics

There is no good way for a T-API Topology service provider to articulate to the client what it would cost for a potential path (e.g., in terms of delay) to cross a node from interface (NodeEdge point) A to interface B. Because nodes (especially composite abstract nodes) may contribute to overall path costs much more than links connecting the nodes along the path, this fact makes the provided topology unfit for the client's path selection optimizations. [Note: To be fair, the T-API Topology service does allow a composite abstract node (representing a group of interconnected nodes) to refer to the topology describing the abstract node's internals (node's `encapTopology` attribute). Hence the client may in theory apply path computation algorithms on the abstract node's internal/encapsulated topology to figure out whether the abstract node can switch a path between a given pair of the abstract node's NodeEdge points, as well as the cost penalties the path will accrue by doing so. However, such a technique defeats the whole purpose of creating the abstract node in the first place, which is hiding multiple topological elements behind the abstract node, so that the top level topology becomes smaller and easier to use in path computations. In other words, if the client has to "dive" into the abstract node's internal topology every time the client needs to

understand whether and how a path can cross the abstract node, the client would be better off if the abstract node were not provided, and instead, the node's internals were presented directly in the top level topology.]

This issue does not exist in the IETF TE Topology model. A TE node's Detailed Connectivity Matrix attribute (Figure 1, upper right) associates with each (abstract or physical) node's connectivity matrix entry a vector of costs (in terms of generic TE cost, delay, intra-node SRLGs, etc.) that a potential TE path will have to add to its end-to-end costs should the path select the entry to cross the node. Likewise, a TE path's source/destination TTP's Detailed Local Link Connectivity List attribute (Figure 2, upper right) indicates what it would cost for the path to start/stop on a given first/last link. [Note: In the IETF TE topology model an abstract TE node also points to the encapsulated TE topology describing the node's internals. However, the client is expected to peruse the node's encapsulated TE topology only in exceptional situations (e.g. during trouble shooting), rather than under normal conditions, such as routine path computations.]

2.3. Topology Updates

Suppose that a T-API Topology service client has requested and received a topology from one of its providers (for example, the topology presented in Figure 3). It is imperative that as soon as this done the provider starts updating the client (continuously and in unsolicited way) with changes happening to the topological elements and their attributes that the client has expressed interest in - otherwise, the client would be forced to make decisions on stale information.

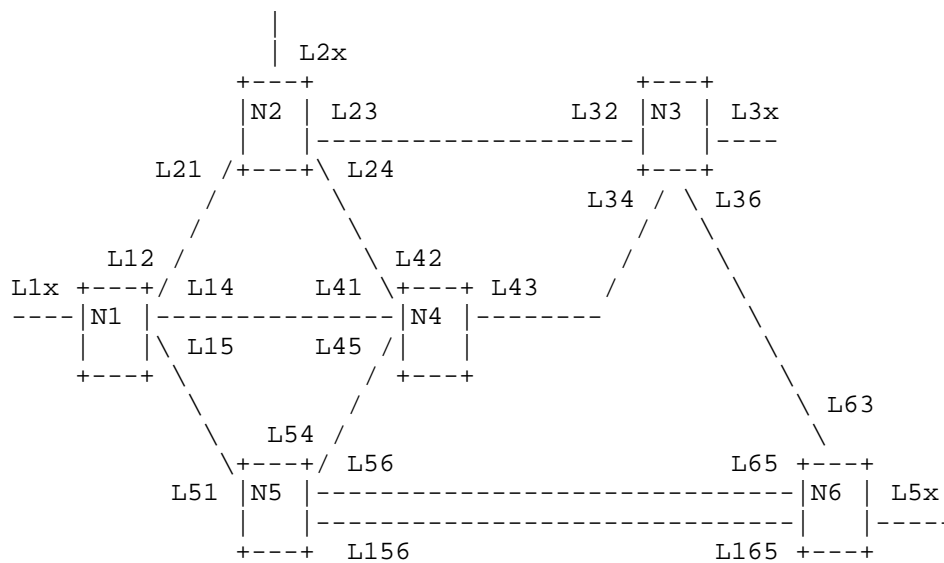


Figure 3: Topology presented to T-API Topology service client

The only way this could be done in T-API is via using T-API Notification service, specifically, the Attribute Value Change (AVC) Notification service, which in a nutshell works as follows:

- o Provider registers with the service the types of pre-defined AVC events it is willing and capable of providing notifications for, along with the set of pre-defined object types that may comprise the notification contents;
- o Client discovers the registered notifications it can subscribe to and subscribes to some of them, specifying filters to tailor the notifications to its needs.

There are two problems with this paradigm:

1. The client has a very limited way to express which notifications it is interested in, as well as the contents, triggers and frequency of such notifications. Note that even for the same topology element type (e.g., link) different clients may need to know different things, at different scopes and granularities, with respect to the attribute changes. For example, one client may want to hear about links that experienced changes in any attribute, while another client may be interested only in links with changes in specific attribute(s). One client may want to learn about link attribute modifications across all provided topologies, while another client may want to know only about such

links that belong to one or more specific (but not other) topologies. One client may want to receive in the notification the entire set of link attributes, while another client would want to learn only about incremental changes (i.e., changes that happened since the previous notification); some clients are interested not in just any attribute change, but rather, want to know when the attribute has reached a specific threshold, etc. As mentioned, a T-API client has only the option to discover what the provider is willing to offer (without the provider really knowing what their clients want to learn) and to subscribe to a subset of that;

2. In order for the client to understand/interpret the notifications registered by the provider, all notification event types, as well as the types of objects comprising the notification content, must be explicitly pre-defined. Considering the sheer number of, say, link attributes (especially, combinations of them) that different clients may be interested in, and the possible scopes, granularities and triggers of the notifications; explicit pre-definition of notifications is awkward, limited and impractical (if not infeasible).

In sharp contrast, the IETF TE topology model requires no explicit definition of notifications. When the client subscribes to a TE topology update notification it:

- a. defines the notification event type by specifying the YANG XPath from the TE topology data store root to the data store node(s) associated with link attribute(s) encompassing the client's points of interest;
- b. specifies another XPath pointing to the data store's sub-tree, node or group of nodes to identify the content of the notification and whether the entire new state or incremental changes must be provided;
- c. defines the trigger for the notification, which could be any change in the node(s) of interest or a specific increment in value or the value hitting a specific threshold;
- d. optionally defines the highest notification frequency at which the client wants to receive the notifications.

To illustrate this assume that the IETF TE Topology model client wants to be notified about all TE links whose available capacity has dropped below 10G, with the notification carrying the actual link's available capacity. In this case the client will:

- a. specify root->all TE topologies -> all TE links->linkAttributes->bandwidth XPath as the notification type;
- b. specify the same XPath to define the desired notification content;
- c. define the notification trigger by specifying the low and high thresholds (e.g. 10G and 15 G respectively);
- d. optionally specify the highest frequency of updates the client is capable/willing to consume.

Note that no explicit definitions for the notification were required. After the client registers with the provider the defined subscription, the latter knows exactly what the former wants to be notified about and how. Similar notifications are possible to register with the provider with respect to any TE topology element attribute or combination of thereof.

2.4. Topology Telemetry Collection

Topology service clients (which in the T-SDN context could be various controllers or applications, such as multi-domain coordinators, IP/transport integrators, orchestrators, big data collectors, analytics processors, network planners, etc.) are hungry for accurate real time network state information (a.k.a. network telemetry). This knowledge is instrumental for a client in keeping the network under its control healthy, stable and optimized under conditions of fiber cuts, hardware and software failures. In particular, network telemetry streams provided by the client's providers allow for the client to identify/predict failing network resources and route the provided transport/connectivity services away from them; to identify/predict points of congestion and eliminate/mitigate the congestion by deploying extra network capacity in a timely manner and so forth. Network telemetry is a valuable source of information useful for network planning, trouble shooting and many other things. Network telemetry is especially important for topology service clients because topologies represent - in an abstracted way - the physical network resources.

[Note: At the time of writing of this memo there were no known TAPI design/modeling activities related to telemetry streaming for any of the T-API services].

Topology telemetry collection is similar in nature to receiving updates on topology attribute changes. Per the description in section 1.3, T-API Notification service, State Change (SC) Notification service is the only mechanism theoretically (i.e. after

all the necessary modeling concepts and attributes, such as statistics counters, are in place) available for the client to subscribe and for the provider to stream the requested network telemetry. T-API SC Notification service has the same drawbacks as the AVC Notification service, specifically:

- a. limited capability for the client to articulate what telemetry (event type, content, granularity, etc.) it seeks to receive;
- b. necessity for explicit definition of the telemetry events and notification messages.

These issues do not exist in the network telemetry streaming machinery offered by the IETF Topology model. Let's consider, for example, that the client wants to identify "flipping" TE links (i.e. TE links frequently changing their UP/DOWN operational status) and obtain in the notification the entire state information for such TE links. In order to achieve this the client needs to:

- a. specify root->all TE topologies -> all TE links->linkStatistics->linkUPCounter XPath as the notification type;
- b. specify root->all TE topologies -> all TE links->linkState XPath to describe the desired notification content;
- c. define the notification trigger by specifying the number the model data state node of interest (the linkUPCounter) must increment by for the next notification to be issued;
- d. optionally specify the highest frequency of notifications of this type the client is capable/willing to consume.

2.5. Topology Name/Address Spaces

T-API topologies are required to have each node and link assigned a globally unique UUID. This means that all T-API Topology service clients and providers have to resolve potential UUID collisions via allocating the UUIDs from a universal name space governed by a centralized authority (in a similar way to how global IP addresses are assigned in IP networks).

The IETF TE Topology model allows for all TE topologies to have independent name spaces for the TE node, link and SRLG IDs, which not only eliminates the problem of ID collisions, but also greatly simplifies the design and implementation of network applications such as L0/L1 VPNs.

In Figure 4 a TE topology provider exposes its native (i.e. real, physical) TE topology as separate abstract TE topologies to two clients, each one customized separately on per client basis. According to the IETF TE Topology model each of the three depicted TE topologies may have an independent name space for their respective TE node, link and SRLG IDs.

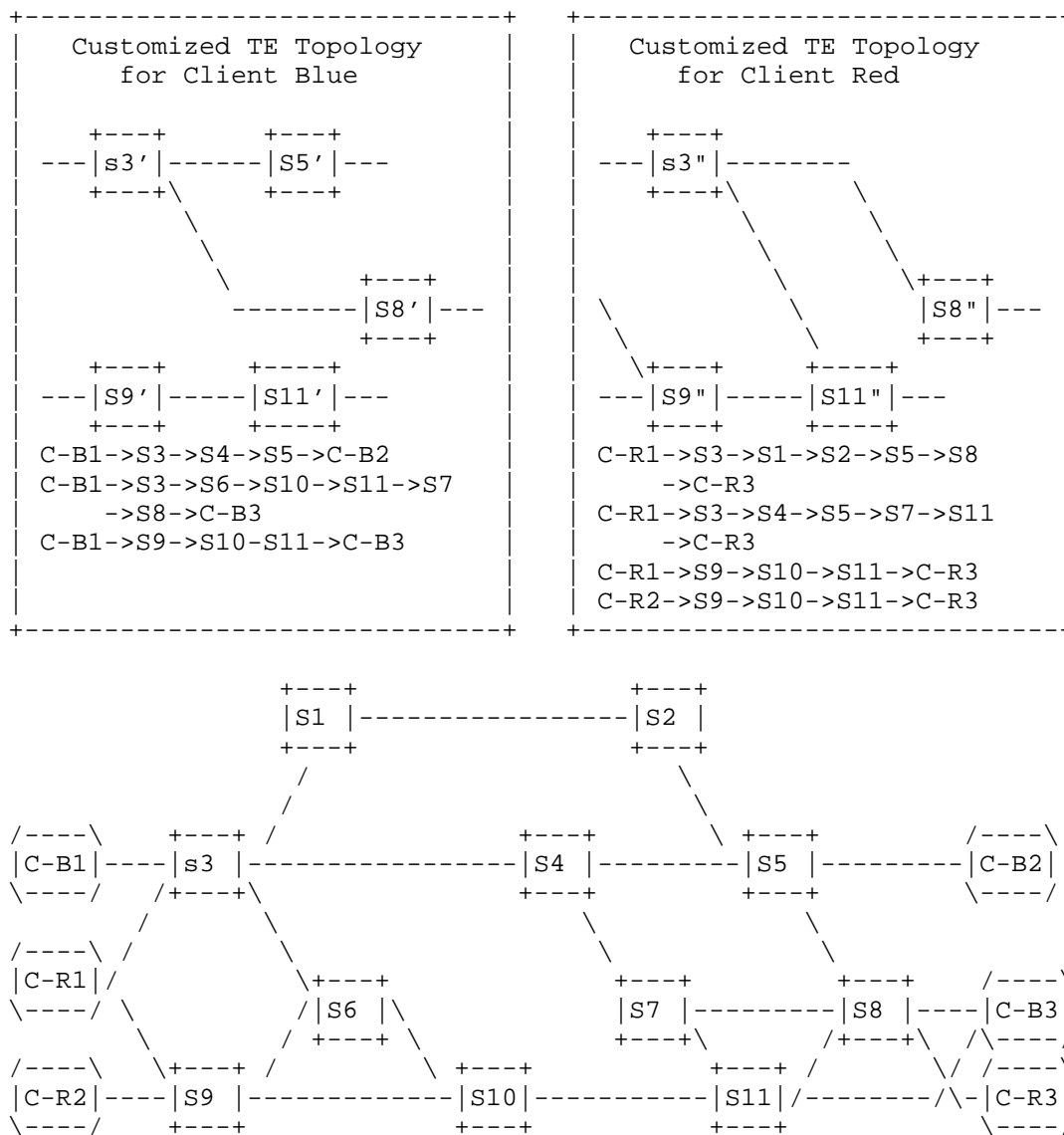


Figure 4: Abstract TE topologies customized for different clients

2.6. Topology Relationships

An IETF TE Topology model provider may expose to the same client multiple TE topologies, which:

- o could be native (as known to the provider, unmodified) or abstract (generated by the provider as overlays based on native or lower level abstract TE topologies);
- o could describe different layer networks in accordance with distinct layer-specific model augmentations;
- o abstract TE topologies could be of a different type (e.g. single node, link mesh, etc.) and of a different hierarchy level;
- o abstract TE topologies could be optimized based on different optimization criteria (e.g. smallest cost, shortest delay, best link protection, etc.)

The provider can convey to the client the TE topology optimization criteria, as well as the provider's preference as to the order in which the provided TE topologies are to be used via topology scope attributes specifically designed for this purpose. Furthermore, the TE Topology model defines various inter-topology relationships designed to describe abstract TE topology hierarchies, client-server layer network (vertical) relationships and domain neighboring (horizontal) relationships. The defined inter-topology relationships are as follows:

- o TE node underlay topology: A composite abstract TE node of a higher hierarchy level TE topology X, representing a group of inter-connected TE nodes that belong to a lower hierarchy level TE topology Y, has an attribute pointing to Y (i.e., ID of the abstract TE node's internal/encapsulated TE topology);
- o TE link underlay topology: A TE link of a TE topology X can point to TE Topology Y which was used by the provider to compute primary and backup TE paths that are (or are to be) used by the actual or potential TE tunnel (transport connectivity) supporting the TE link in question. The TE paths themselves could be provided in the same TE link attribute;
- o Supporting node/link topology: A given TE node or link may show up in multiple TE topologies catered by the provider to the client. In order for the provider not to provide/update (and for the client not to consume) multiple identical sets of attributes, the model allows for providing/updating only for one (original) TE node/link, and having the "twins" point to the original TE mode/

link, as well as to the TE topology where the original TE node/link could be found;

- o Source node/link topology: A given TE node or link catered by the provider as a part of a TE topology to the client may be provided to the provider by one of its own providers. In such case the TE node/link in question can point to the original TE node/link, as well as to the TE topology where the original is defined, thus allowing for multi-level multi-provider TE topology hierarchies (see Figure 5);
- o Inter-layer lock: This is the relationship/attribute that associates TE links of a higher layer network TE topology with TE Tunnel Termination Points (TTPs) of one or more lower layer network TE topology(ies) to articulate to the client inter-topology /inter-layer adaptation capabilities, to lock the TE topologies describing separate layer networks vertically, thus allowing for client multi-layer path computations and other multi-layer TE applications;
- o Inter-domain plug: This is a relationship modeled via an inter-domain TE link attribute that allows for a client managing interconnected multi-domain networks (with each domain served by a separate provider) to identify neighboring domains and to lock the TE topologies provided by all providers horizontally, thus producing TE topologies homogeneously describing the entire multi-domain network and allowing for end-to-end path computations across the network.

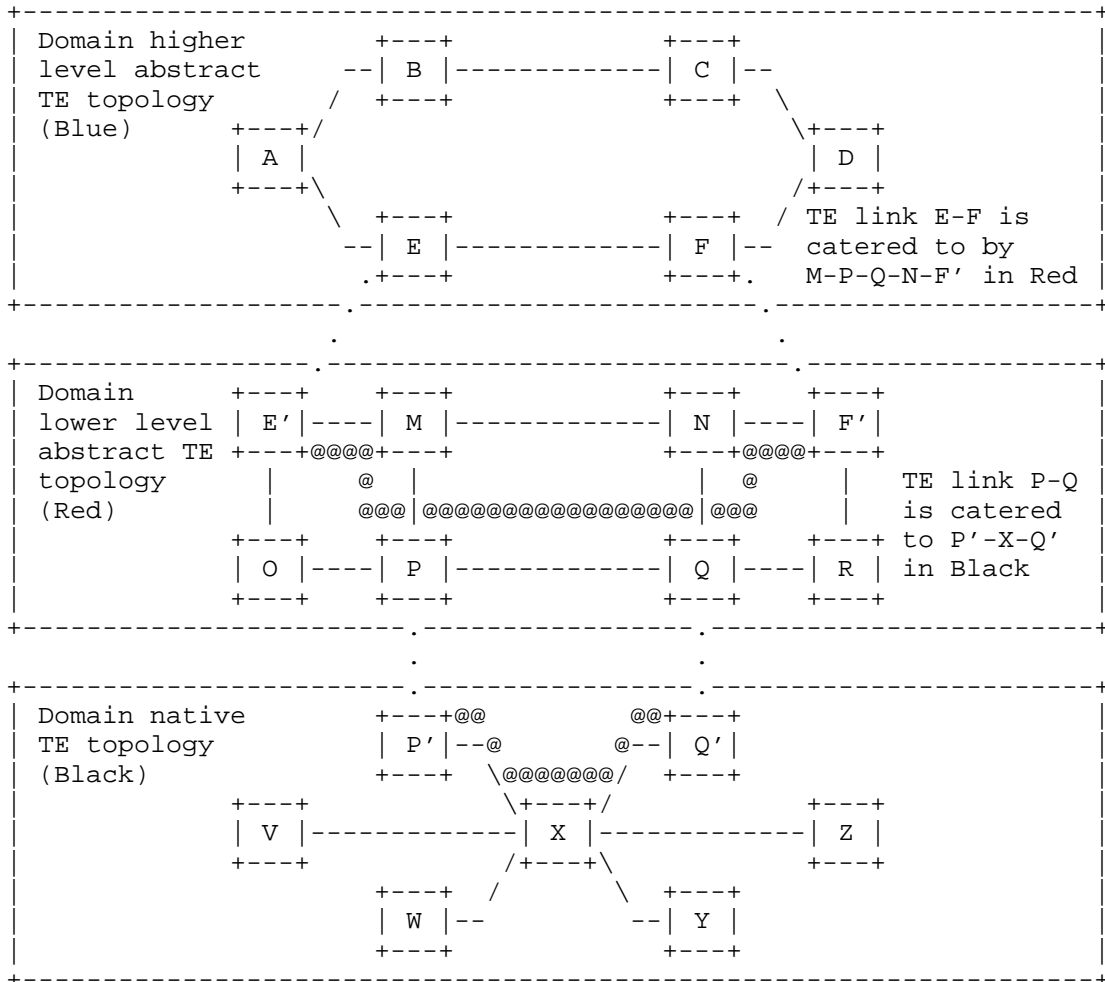


Figure 5: Hierarchical multi-provider abstract TE topologies

A T-API Topology service provider is also allowed to expose multiple topologies to the client. The only inter-topology relationship defined is the Node's `encapTopology` (which is effectively the same as the IETF's TE node underlay topology relationship described above). Otherwise, all the provided topologies are independent. It is not clear for the client what is the purpose of each of them, what is the provider's preference as to how and in which order they are supposed to be used, and why several same layer topologies, rather than one, were provided to the client in the first place.

2.7. Topology Attributes

Compared to the IETF TE Topology model, T-API Topology nodes and links are missing some important attributes. Specifically, T-API nodes, as mentioned in section 1.1, have no analogs to the Connectivity matrix attribute and the TE TTP container describing nodes switching and termination capabilities/limitations respectively. Furthermore, the T-API Topology service does not have a concept of TTP, which in the context of the IETF TE Topology model conveys to the client various important edge characteristics for a TE tunnel that could be provided by the network described by a given TE topology. Such characteristics include:

- o Potential TE tunnel protection capabilities (e.g., whether 1+1 protection could or could not be supported for the tunnel edge);
- o Adaptation capacities (i.e., which higher layer network payload types and from which higher layer link termination points can be adopted on the TE tunnel edge, the amount of adaptation bandwidth still available, etc.);
- o Technology-specific TTPs describe technology specific properties (e.g. TTP representing an OCh layer transponder can announce whether the transponder's receiver/transmitter is fixed or tunable, and in the latter case what is the range and resolution of the tunability; supported FECs and signal modulation modes, transmit/acceptable optical signal power levels and OSNRs, etc.)

The T-API Topology link is missing the following attributes:

- o Administrative groups (administrative colors) - an attribute describing the link's association with pre-defined groups of links; such groups could be used as constraints in the client's path selection/optimization algorithms to mandate/disallow or encourage/discourage the resulting paths to follow/avoid links related to the specified groups;
- o Link protection/restoration capability - an attribute that could be also used as a path computation constraint or path optimization criterion, for example, to force or encourage the resulting paths to follow sufficiently protected links;
- o Link properties defining whether the link is:
 - A. actual (with committed network resources) or potential;
 - B. static (with pre-established and always-in-place server layer connectivity supporting the link) or dynamic (for which the

connectivity is dynamically put in place if/when the link is used by at least one client connection and is dynamically released as soon as the link is used by none of the client's connections);

- o Link's underlay primary and backup paths and ID of the topology used for their computations.

2.8. Topology Service Relationships with Other Services

IETF TE topology and TE tunnel models are related. For example, a TE link can point via the Supporting Tunnel ID attribute to the lower layer network TE tunnel providing the transport connectivity for the TE link. Likewise, a TE tunnel has an attribute pointing to the TE link it supports, as well as the TE topology which the TE link is part of. These cross-references are instrumental for the client in terms of understanding which network resources a given TE link represents, especially useful at the times of trouble shooting. Additionally, IETF TE tunnel defines and supports the concept of Hierarchical TE links and tunnels. Hierarchical TE tunnels automatically insert dynamic hierarchical TE links into the specified TE topologies as soon as the tunnels are successfully set up (and remove the hierarchical TE links from the respective TE topologies when released). [Note: Hierarchical TE tunnels and links are instrumental in multi-layer traffic engineering].

Furthermore, both TE topology and TE tunnel models are tightly coupled with the IETF YANG based notification machinery, which allows the client to retrieve any telemetry or attribute change updates as long as those telemetry/attribute changes are defined as data state nodes or sub-trees in the respective models.

In contrast, all T-API services (i.e. Topology, Connectivity, Path computation, Virtual Network and Notification) are independent from each other.

2.9. Topology Negotiation and (Re-)configuration

When a client of the IETF TE Topology model/interface receives one or more abstract TE topologies from one of its providers, it may accept the topologies as-is and merge then into one or more of its own native TE topologies. Alternatively, the client may choose to request a re-configuration of one, some or all abstract TE topologies provided by the providers. Specifically, with respect to a given abstract TE topology, some of its TE nodes/links may be requested to be removed, while additional ones may be requested to be added. It is also possible that existing TE nodes/links may be asked to be re-configured (e.g., TE links may be requested to be SRLG disjoint).

Furthermore, the topology-wide optimization criteria may be requested to be changed. For example, underlay TE paths supporting the abstract TE links, currently optimized to be shortest (least-cost) paths, may be requested to be re-optimized based on the minimal delay criteria. Additionally, the client may request the providers to configure entirely new abstract TE topologies and/or to remove existing ones. Furthermore, future periodic or one-time additions, removals and/or re-configurations of abstract TE topologies, topological elements and/or their attributes could be (re-)scheduled by the client ahead of time.

It is the responsibility of the client to implement the logic behind the above-described abstract TE topology negotiation. It is expected that the logic is influenced by the client's local configuration/templates, policies conveyed by the client's clients, input from the network planning process, telemetry processor, analytics systems and/or direct human operator commands. Figure 6 exemplifies the abstract TE topology negotiation process. As shown in the Figure, the original abstract TE topology exposed by a provider was requested to be re-configured. Specifically, one of the abstract TE links was asked to be removed, while three new ones were asked to be added to the abstract TE topology.

The ONF T-API Topology service client has no say as to how the abstract topologies exposed to the client by its providers should look like. The only option for the client is to consume the provided topologies as offered. This is a serious disadvantage because it is the client (not providers) that knows which topologies suite best the client's needs.

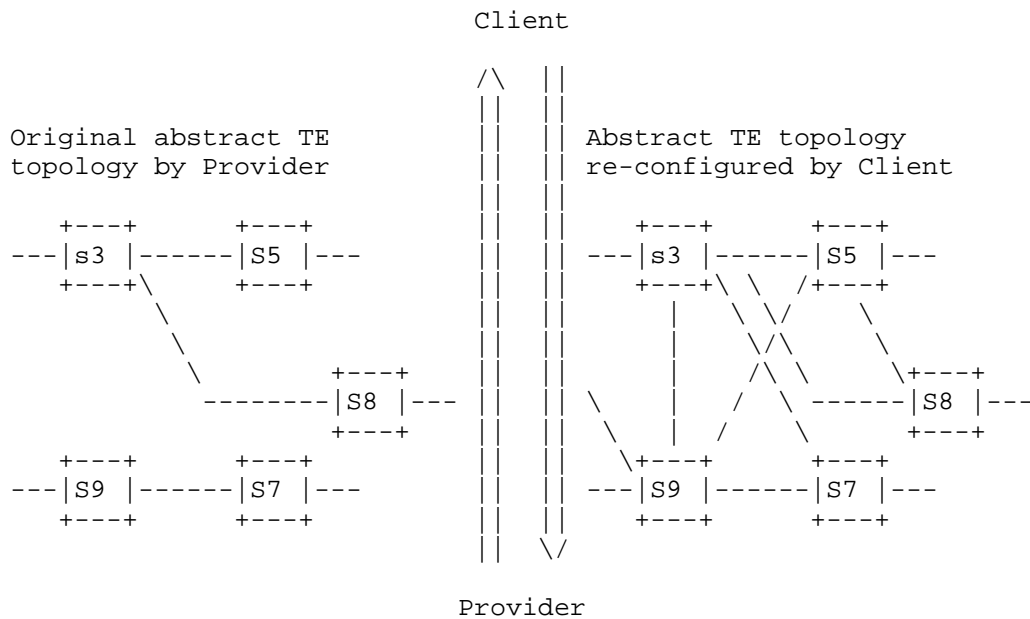


Figure 6: Provider-Client abstract TE topology negotiation

2.10. Integration with IP/MPLS

The IETF TE Topology model is naturally and intimately integrated with IP/MPLS layer models defined for IP/MPLS layer traffic engineering. For example, currently Segment Routing (SR) and Service Function Chaining (SFC) technologies heavily rely on and actively use the TE Topology model. Specifically, SR combines the TE topology model with layer 3 (IP reachability) topology model to facilitate path computations that account for either or both TE and IP reachability information. Likewise, SFC makes use of the TE topology model for computing service function chains optimized according to the combined criteria of real/virtual network function location and best available (possibly in different layers) TE paths to connect the network functions.

It is not clear how the ONF T-API Topology service can fit in and to what extent it can be integrated into the IP/MPLS layer traffic engineering.

3. Connectivity Service

3.1. Connectivity Service Protection

It is not possible for a T-API Connectivity service client to request from a provider a protected service like, for example, the one presented in Figure 7. In the Figure a connectivity service is supported by two disjoint connections - primary (solid blue) and backup (broken yellow), with the client traffic normally carried over the primary connection, but which could be quickly and dynamically switched onto the backup connection as soon as a network failure affecting the primary connection is detected.

The inability to request protected connectivity services from a provider leaves the T-API Connectivity service client with the problem of protecting its own traffic against the network's failures. Admittedly, the client can address this with the following sequence of operations:

1. The client requests a primary connectivity service connecting the desired pair of client device ports over the network managed by the T-API Connectivity service provider;
2. The client requests a secondary connectivity service connecting the same pair of client device ports, which is sufficiently diverse from the primary service (incidentally, this could be problematic due to the independent nature of the path computations carried out by the provider. Specifically, the path selected for the primary service may block disjoint paths for the secondary service. This is a known issue related to sequential/independent path computations, which could be solved via concurrent path computation for both services);
3. The client binds at both ends the two connectivity services in accordance with the desired protection scheme;
4. From then on the client is constantly monitoring the performance and health of both services;
5. In case the primary service is affected by a network failure (while the secondary service remaining healthy), the client coordinates the protection switchover;
6. In case it is detected that the previously broken primary connectivity service is repaired, the client coordinates the protection reversion (i.e. reversion to the normal forwarding of the client traffic).

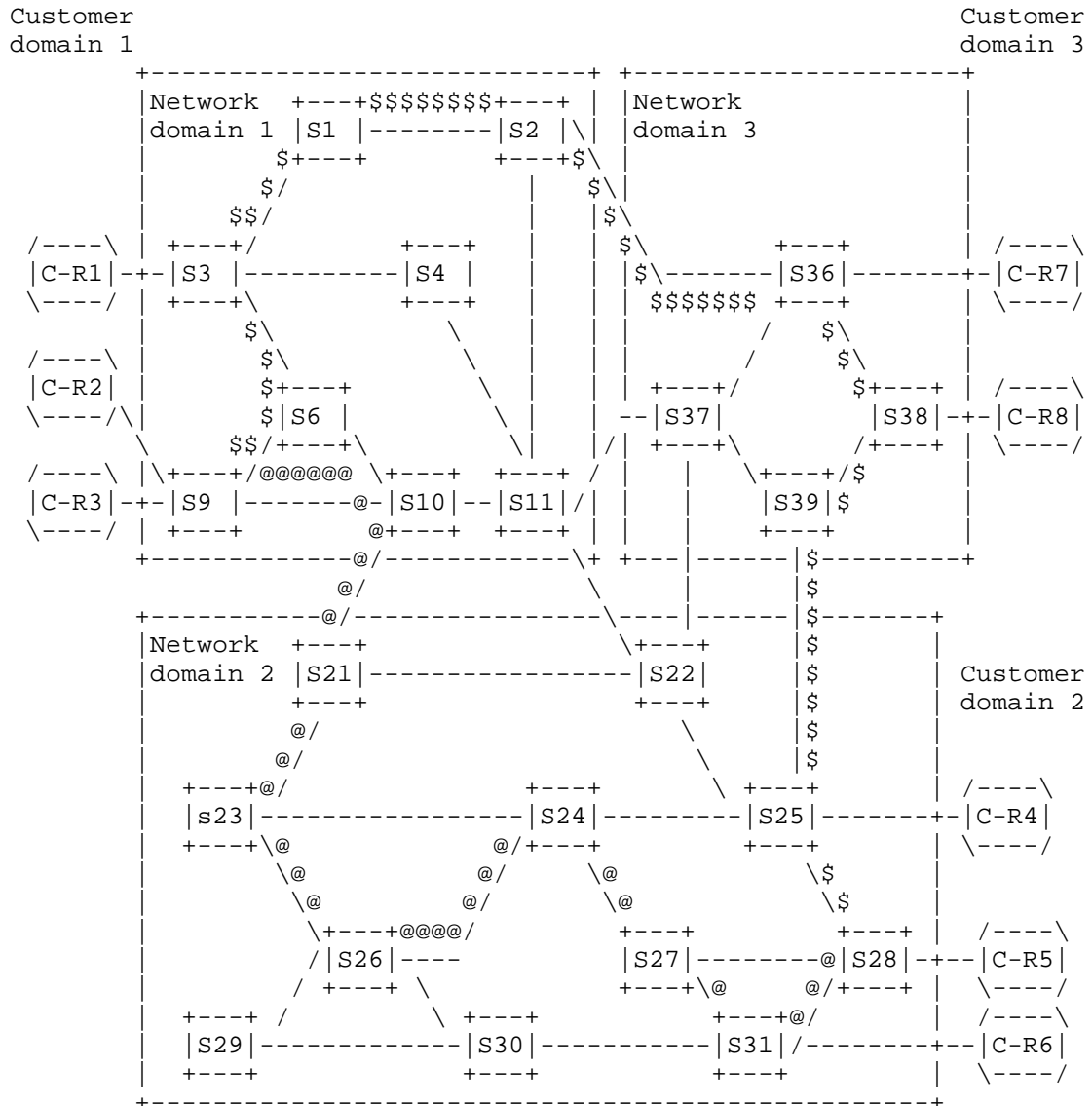


Figure 7: Protected connectivity service

In contrast, an IETF TE tunnel model client normally delegates all the described above operations to the provider by simply configuring the requested transport service (i.e. TE tunnel or a single-domain segment of a multi-domain TE tunnel) to be protected. In doing so the client specifies the required protection type, as well as the level of primary/backup connections disjointedness. Additionally,

the client may specify a set of constraints common for both connections, as well as constraints (e.g. inclusions, exclusions, etc) specific to each connection. Furthermore, the client may even specify, for a given transport service, multiple sets of such constraints in descending preference order for the provider to try before notifying the client about the setup failure. For example, the client may request in this way for a TE tunnel that the primary and backup connections must be SRLG disjoint, and, if this proves to be not possible, to relax the disjointedness criterion to link-disjoint.

3.2. Hierarchical Connectivity Service

A transport network provider may control a multi-layer (e.g. Ethernet/ODUk/OCh) network. On such a network the provider has flexibility to dynamically set up connectivity/transport services in one or more lower layer networks to augment a higher layer topology that is otherwise insufficient for provisioning of a connectivity service requested by the client.

In the top-to-bottom approach the client simply requests a connectivity service in the desired layer network. While processing the request, the provider:

- o performs its internal multi-layer path computation,
- o identifies one or more lower layer connectivity services required for the successful provisioning of the requested service;
- o dynamically (and unknowingly to the client) sets up the so-identified lower layer connections;
- o sets up the connection(s) supporting the connectivity service requested by the client.

Both T-API Connectivity service and IETF TE Topology model/interface support the described top-to-bottom multi-layer connectivity services. The approach is simple for the client; however it does not work in many multi-domain use cases. Consider, for example, a multi-domain transport network presented in Figure 8. Consider further that a Multi-Domain Service Coordinator is requested to set up Ethernet layer connectivity service (marked in blue) across three domains, each of which is controlled by a separate provider. Assume also that in order to satisfy the request an underlay ODUk layer TE tunnel (marked as red) also spanning multiple domains needs to be provisioned. This could be achieved via a bottom-to-top multi-layer connectivity service provisioning approach, which includes the following:

- o the client (i.e. the Multi-Domain Coordinator) performs its own multi-layer path computation on a network wide TE topology (a product of merging the TE topologies exposed by all providers);
- o the client identifies one or more lower layer TE tunnels required for the successful provisioning of the requested service;
- o the client coordinates the multi-domain setup of each of the identified lower layer TE tunnels;
- o the client instructs each lower layer TE tunnel's first and last domain provider to add a dynamic TE link in their respective higher layer TE topologies;
- o the client triggers and coordinates the setup of the connection(s) supporting the requested connectivity service, constraining the connection path(s) to follow the dynamic TE links supported by the lower layer TE tunnels;
- o the client adds into its own (network-wide) TE topology, dynamic TE links supported by the lower layer TE tunnels to make the remaining capacity on the tunnels available for path computations for other higher layer connectivity services.

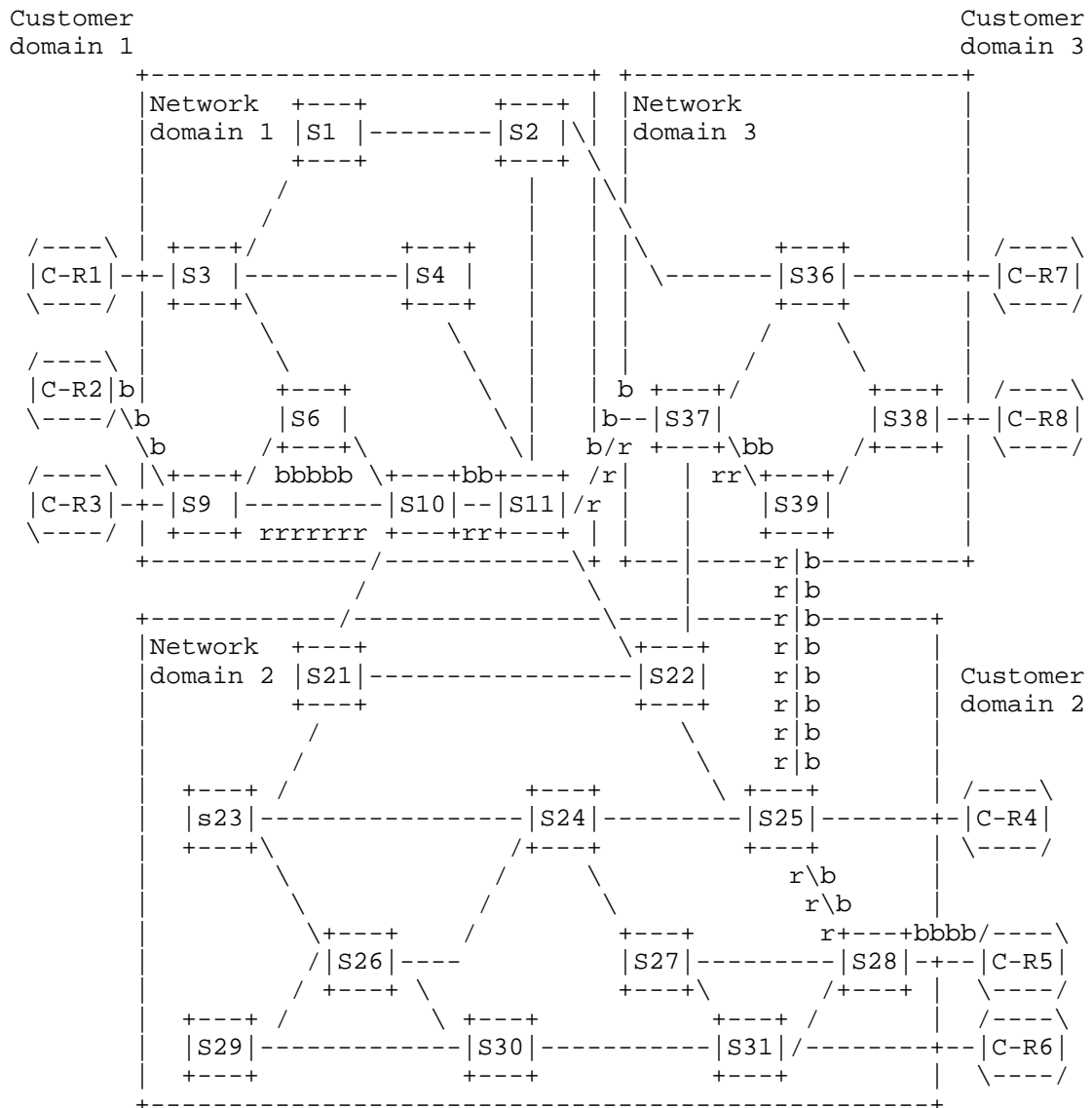


Figure 8: Hierarchical connectivity service

The IETF TE topology model supports the described bottom-to-top multi-layer connectivity service provisioning paradigm via Hierarchy TE tunnels. A hierarchy TE tunnel, once successfully set up, automatically adds into the specified TE topology a TE link it supports and withdraws the TE link from the TE topology if/when released.

T-API Connectivity and Topology services do not support the concept of hierarchical connectivity/dynamic links.

3.3. Connectivity Service Re-optimization

An IETF TE tunnel model/interface client, when requesting a transport service from a provider, can control - via a designed for this purpose knob (lockDown attribute) - whether the connection(s) supporting the service must be "pinned" to their respective original paths (the paths selected at the setup stage), or whether the provider may occasionally perform a service re-optimization, resulting in service connection replacement toward more optimal paths. This knob is especially useful in conjunction with a connectivity scheduling service (see section 2.6), allowing for the client to specify time intervals at which the re-optimization of a given transport service (and subsequent potential traffic hits) is acceptable for the client. For example, the client may configure a transport service to get "unpinned" every Saturday at 1 am for service re-optimization procedures and to get "re-pinned" after that for another week.

T-API Connectivity service clients have no way of controlling of connectivity service re-optimization operations.

3.4. Connectivity Service Templates

The IETF TE tunnel model defines containers of named transport service configuration sets that could be shared by multiple services. This not only simplifies for the client the process of transport service configuration, but also allows manipulation of multiple services by a single configuration change. For example, a client may define a set of constraints named Foo that forces a transport service primary path to go through a node X. If, later, the client modifies Foo by substituting node X with node Y, all transport services configured with the constraint set Foo will (be attempted to) be replaced onto path(s) going through node Y.

The T-API Connectivity service model does not have a similar concept.

3.5. Connectivity Service Attribute Change Update Notifications and Telemetry Streaming

Both T-API and IETF modeling rely on respective notification tools universal across all interfaces. Therefore, connectivity service attribute change notifications and telemetry streaming is no different from the topology notifications and telemetry streaming discussed in sections 2.3 and 2.4

3.6. Connectivity Scheduling

T-API Connectivity service has the `_schedule` attribute that includes just two parameters: `startTime` and `endTime`. This allows for a client to schedule at a specified time and for a specified period of time a one-time kickoff of a service configured initially (presumably) as disabled. It is not possible to schedule multi-time (periodic) kickoffs. Furthermore, the scheduling granularity is connectivity service as a whole. In particular, it is not possible to schedule re-configurations of one or several service parameters (e.g. bandwidth requirement, inclusion/exclusion path, etc.).

There is an ongoing effort in IETF to produce a generic scheduling tool that could be applied to any of YANG models. Similar to the notification subscription tool - allowing for the client to subscribe on notifications with respect to any data state (`CONFIG=FALSE`) node defined in any supported by the provider data store - the scheduling tool will allow for the client to schedule periodic and/or one-time modification of any configuration (`CONFIG=TRUE`) leaf of any supported data store. For example, if it is required to schedule a re-configuration of the bandwidth requirement for one or more selected services, the client will specify an XPath pointing to the configured bandwidth attribute of the services of interest and convey the new bandwidth requirement and the timetable for the service bandwidth re-configuration. [Note: At time intervals outside of the scheduled range, the service configured bandwidth will remain/be restored to the value provided during initial service configuration.]

3.7. Potential Connectivity Service

The IETF TE topology model defines a number of "unconventional" configuration modes to be specified by a client and supported by a provider of transport services. One of those modes is the `COMPUTE_ONLY` mode. When a provider processes a request for a transport service configured in the `COMPUTE_ONLY` mode, it performs the normal path computation for the service, but does not trigger setup of the connection(s) supporting the service. Instead, the computed paths are returned to the client as a part of normal service attribute change notification. Furthermore, when the provider detects a change in the managed network potentially affecting the returned paths, it may re-evaluate the paths and notify the client if they have become infeasible or more optimal paths are available.

The concept of `COMPUTE_ONLY` transport services makes a good foundation for Path computation service/interface between the Client and the Provider (see more in section 4).

4. Path Computation Service

A client of a transport network can discover the network resources available for the client in one of the two ways:

- o by requesting from the network provider , via a topology interface, one or more topologies describing the network with respect to its availability to the client;

or

- o by requesting, via a path computation interface, that the provider identify potential paths that could connect various client device ports across the network.

To support the latter option, ONF T-API has introduced a Path computation service dedicated to the purpose. A T-API Path computation service client can issue a path computation request specifying the identities of the required path source and destination end points, the layer network in which the paths are to be determined, the required mutual diversity of the resulting paths, various path computation constraints (e.g., bandwidth requirements, inclusions, exclusions, etc.) and path selection optimization criteria (e.g., smallest cost, shortest delay, etc.). A T-API Path computation service provider is expected to satisfy the request by running a path computation algorithm and responding to the client with zero, one or more resulting paths.

In contrast, IETF modeling does not offer a dedicated mechanism/model to support the Client<=>Provider path computation interface. Instead, it is suggested to use the YANG TE tunnel model and request and manipulate path computations in the form of COMPUTE_ONLY TE tunnels as described in section 2.7. This approach has some important advantages as compared to the T-API Path computation service:

Simplicity: provided that both the client and the provider know how to request, manipulate and support transport services, there is no additional interface/model for the client to learn how to use and functionality for the provider to support;

Accuracy: T-API Path computation and Connectivity services are not related. It cannot be guaranteed that the set of path computation constraints conveyed by a T-API Path computation service client will match the set of path computation constraints internally generated by a T-API Connectivity service provider even when the configuration parameters - source/destination, layer network,

bandwidth and others - match. There are many reasons for that, including:

- A. additional constraints could be imposed by the provider based on some internal and possibly proprietary knowledge about the network (unknown to the client);
- B. various internal policies could relax, harden or overwrite other constraints;
- C. various internal policies could modify or overwrite the requested optimization criteria;
- D. etc.

Furthermore, the provider may even use different path computation engines to provide the Path computation and connectivity services. All this may result in the paths returned to the Path computation service client being different from the paths taken by the corresponding (same source/destination and other constraints) connectivity services. The difference may be in path costs, delay and fate sharing characteristics, etc. In extreme cases the Path computation service client may even receive unprovisionable and hence useless paths.

IETF COMPUTE_ONLY TE tunnels, on the other hand, do not have such problems. It is inherently guaranteed that the client will be notified/updated with paths which are exactly the same as the ones that would be taken by connections of "conventional" TE tunnels for the same configuration inputs;

Path staleness: paths returned to the T-API Path computation service client may become unfeasible at some later time because of changes in the network's state. There is no way for the Path computation service provider to convey this fact to the client. In contrast, IETF COMPUTE_ONLY TE tunnel provider can use the intrinsic attribute change notifications to let the client know that previously provided paths have changed, have become unfeasible or that better, more optimal paths have become available.

5. Virtual Network Service

A client of a transport network may want to limit the transport network connectivity of a particular type and quality to defined subsets of its device ports interconnected across the network. Furthermore, a given transport network may serve more than one client. In this case some or all clients may want to ensure the availability of transport network resources in case dynamic

(re-)connection of their device ports across the network is envisioned. In all such cases a client may want to set up one or more Virtual Networks over the provided transport network.

ONF T-API has introduced a dedicated service for this purpose - the Virtual Network service (VNS). A VNS client can request creation of a VNS specifying the layer network of the VNS and the Traffic Matrix requirement. The client has no control over the requested VN beyond that. In particular, it is up to the provider to decide which network resources will support the VN in question. The client has no say as to how the underlying network topology should look, how the topology needs to be optimized for the VN (e.g. shortest delay rather than smallest cost), what is the required level of the topology link protection and mutual diversity, and so forth.

As in case of the path computation interface, IETF modeling does not offer a separate model to support VNS. Instead, it encourages using the TE topology model - leveraging the IETF abstract TE topology's ability to be configured by the client. In a nutshell, the client configures and manipulates a VN as a customized abstract TE topology based on the TE topologies already exposed by the provider. In the simplest case the client requests a single node ("black box") abstract TE topology with desired attributes. In more complex cases the client may opt to construct, for the VN, a separate multi-node/link arbitrary abstract TE topology. In doing so, the client may "borrow" into the VN's topology TE nodes and links from other topologies. Additionally the client may add new composite abstract TE nodes specifying the IDs of TE topologies the nodes will encapsulate, connected by abstract TE links pointing to the respective underlay TE topologies to be used for computation and provisioning of the TE tunnels supporting them. The client/provider negotiation of a "so-cooked" TE topology is described in 1.9. In short, the client is able to manipulate the VN's topology at the granularity of individual topological elements (such as TE nodes and links).

6. Data Modeling Language

Today YANG is a very popular data modeling language. It is a product of IETF NETMOD WG. It is not the only data modeling language produced by IETF (for example, FORCES WG has developed one of its own, arguably - in some aspects - superior to YANG). YANG is neither stable nor perfect. It is constantly evolving with the sole objective to make IETF models more scalable, efficient, inclusive, information-rich: better in all aspects. Supporting non-IETF (e.g. ONF) data models is not a priority. Therefore It is not clear why ONF, while investing a lot of effort in designing Core Information Models, is devoting no effort to designing a data modeling language

of its own that would closely suit support of its CIM. Nor it is clear what would happen if the IETF NETMOD WG decides, for whatever reason to obsolete some of the YANG features/properties/capabilities that ONF models rely upon.

Furthermore, writing CIMS in UML and having them mechanically translated into YANG has its own issues, which includes the following:

- o Many useful YANG features that do not have analogs in UML are not used. For example, T-API YANG models use only non-extendible enumeration type, rather than extendible identity type. This prevents T-API YANG models from being easily extendible via augmentation;
- o T-API YANG models heavily overuse and often misuse YANG RPCs for operations that could be handled simpler and more efficiently by NETCONF/RESTCONF protocol via native edit-config and get operations;
- o T-API YANG models unnecessarily define their own notification subscription/streaming and scheduling mechanisms, instead of leveraging the NETCONF/RESTCONF machinery easily applicable to all YANG models;
- o T-API YANG models make no use of YANG templates and defaults designed to simplify for the client the provider's data store (re-)configuration;
- o T-API YANG models follow the conventions inherited from UML and previously defined REST APIs. As a consequence. the models sometimes are not compatible with the current best practices recommended for YANG model writers and do not always follow YANG model guidelines defined in [I-D.ietf-netmod-rfc6087bis]

7. Security Framework

ONF T-API does not have a security framework of its own. It simply assumes that the proper security could be inherently provided by the underlying protocols. IETF TEAS interfaces, on the other hand, take the security considerations very seriously. They rely on the generic framework ([RFC6241], [RFC8040], [RFC6536], and [I-D.ietf-netconf-rfc6536bis]) allowing for the provider to configure in a universal way various strength AAA protection for any YANG modeled data store accessible via NETCONF or RESTCONF protocol. In particular, said framework allows for the client authentication, identification of the client's privileges with respect to the

information access, required filtering and scoping of the provided information, as well as secure client-provider communication.

8. IANA Considerations

This document has no actions for IANA.

9. Security Considerations

This document does not define networking protocols and data, hence are not directly responsible for security risks.

This document compares two interface technologies of T-SDN controllers. For each specific technology discussed in the document, security framework has been described and compared in the corresponding section.

10. Acknowledgements

The authors would like to thank Christopher Jenz, Diego Caviglia, Aihua Guo, Fatai Zhang, and Italo Busi for their helpful comments and valuable contributions.

11. References

11.1. Normative References

- [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>>.
- [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>>.
- [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>>.
- [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>>.

[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.ietf-teas-yang-te]

Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-05 (work in progress), October 2016.

[I-D.ietf-netconf-rfc6536bis]

Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", draft-ietf-netconf-rfc6536bis-01 (work in progress), March 2017.

11.2. Informative 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.

Authors' Addresses

Igor Bryskin
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Xufeng Liu
Jabil

EMail: Xufeng_Liu@jabil.com

Vishnu Pavan Beeram
Juniper Networks

EMail: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc

EMail: tsaad@cisco.com

TEAS Working Group
Internet-Draft
Intended Status: Standards Track
Expires: September 11, 2017

R. Gandhi, Ed.
Cisco Systems, Inc.
H. Shah
Ciena
Jeremy Whittaker
Verizon
March 10, 2017

Fast Reroute Procedures For Associated Bidirectional Label
Switched Paths (LSPs)
draft-gandhishah-teas-assoc-corouted-bidir-04

Abstract

Resource Reservation Protocol (RSVP) association signaling can be used to bind two unidirectional LSPs into an associated bidirectional LSP. When an associated bidirectional LSP is co-routed, the reverse LSP follows the same path as its forward LSP. This document describes Fast Reroute (FRR) procedures for both single-sided and double-sided provisioned associated bidirectional LSPs. The FRR procedures are applicable to co-routed and non co-routed LSPs. For co-routed LSPs, the FRR procedures can ensure that traffic flows on co-routed paths in the forward and reverse directions after a failure event.

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."

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	3
2.1. Key Word Definitions	3
2.2. Terminology	3
2.2.1. Reverse Co-routed Unidirectional LSPs	4
3. Overview	4
3.1. Fast Reroute Bypass Tunnel Assignment	4
3.2. Bidirectional LSP Association At Mid-Points	5
4. Signaling Procedure	6
4.1. Bidirectional LSP Fast Reroute	6
4.2. Bidirectional LSP Association At Mid-points	7
5. Message and Object Definitions	7
5.1. Extended ASSOCIATION Object	7
6. Compatibility	8
7. Security Considerations	9
8. IANA Considerations	9
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Authors' Addresses	11

1. Introduction

The Resource Reservation Protocol (RSVP) (Extended) ASSOCIATION Object is specified in [RFC6780] which can be used generically to associate (G)Multi-Protocol Label Switching (MPLS) Label Switched Paths (LSPs). [RFC7551] defines mechanisms for binding two point-to-point unidirectional LSPs [RFC3209] into an associated bidirectional LSP. There are two models described in [RFC7551] for provisioning an associated bidirectional LSP, single-sided and double-sided. In both models, the reverse LSP of the bidirectional LSP may or may not be co-routed and follow the same path as its forward LSP.

[GMPLS-FRR] defines Fast Reroute (FRR) procedure for GMPLS signaled LSPs to co-ordinate bypass tunnel assignments in the forward and reverse directions. The mechanisms defined in [GMPLS-FRR] are applicable to FRR of associated bidirectional LSPs.

In packet transport networks, there are requirements where the reverse LSP of a bidirectional LSP needs to follow the same path as its forward LSP [RFC6373]. The MPLS Transport Profile (TP) [RFC6370] architecture facilitates the co-routed bidirectional LSP by using the GMPLS extensions [RFC3473] to achieve congruent paths. However, the RSVP association signaling allows to enable co-routed bidirectional LSPs without having to deploy GMPLS extensions in the existing networks. The association signaling also allows to take advantage of the existing Traffic Engineering (TE) and FRR mechanisms in the network.

This document describes FRR procedures for both single-sided and double-sided provisioned associated bidirectional LSPs. The FRR procedures are applicable to co-routed and non co-routed LSPs. For co-routed LSPs, the FRR procedures can ensure that traffic flows on co-routed paths in the forward and reverse directions after a failure event.

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 RFC 2119 [RFC2119].

2.2. Terminology

The reader is assumed to be familiar with the terminology in [RFC2205], [RFC3209], [RFC4090] and [RFC7551].

2.2.1. Reverse Co-routed Unidirectional LSPs

Two reverse unidirectional point-to-point (P2P) LSPs are setup in the opposite directions between a pair of source and destination nodes to form an associated bidirectional LSP. A reverse unidirectional LSP originates on the same node where the forward unidirectional LSP terminates, and it terminates on the same node where the forward unidirectional LSP originates. A reverse co-routed unidirectional LSP traverses along the same path of the forward direction unidirectional LSP in the opposite direction.

3. Overview

As specified in [RFC7551], in the single-sided provisioning case, the RSVP TE tunnel is configured only on one endpoint node of the bidirectional LSP. An LSP for this tunnel is initiated by the originating endpoint with (Extended) ASSOCIATION Object containing Association Type set to "single-sided associated bidirectional LSP" and REVERSE_LSP Object inserted in the Path message. The remote endpoint then creates the corresponding reverse TE tunnel and signals the reverse LSP in response using the information from the REVERSE_LSP Object and other objects present in the received Path message. As specified in [RFC7551], in the double-sided provisioning case, the RSVP TE tunnel is configured on both endpoint nodes of the bidirectional LSP. Both forward and reverse LSPs are initiated independently by the two endpoints with (Extended) ASSOCIATION Object containing Association Type set to "double-sided associated bidirectional LSP". In both single-sided and double-sided provisioned bidirectional LSPs, the reverse LSP may or may not be congruent (i.e. co-routed) and follow the same path as its forward LSP.

In the case of single-sided provisioned LSP, the originating LSP with REVERSE_LSP Object is identified as a forward LSP. In the case of double-sided provisioned LSP, the LSP originating from the higher node address (as source) and terminating on the lower node address (as destination) is identified as a forward LSP. The reverse LSP of the bidirectional LSP traverses in the opposite direction of the forward LSP.

Both single-sided and double-sided associated bidirectional LSPs require solutions to the following issues for fast reroute.

3.1. Fast Reroute Bypass Tunnel Assignment

In order to ensure that the traffic flows on a co-routed path after a link or node failure on the protected LSP path, the mid-point Point

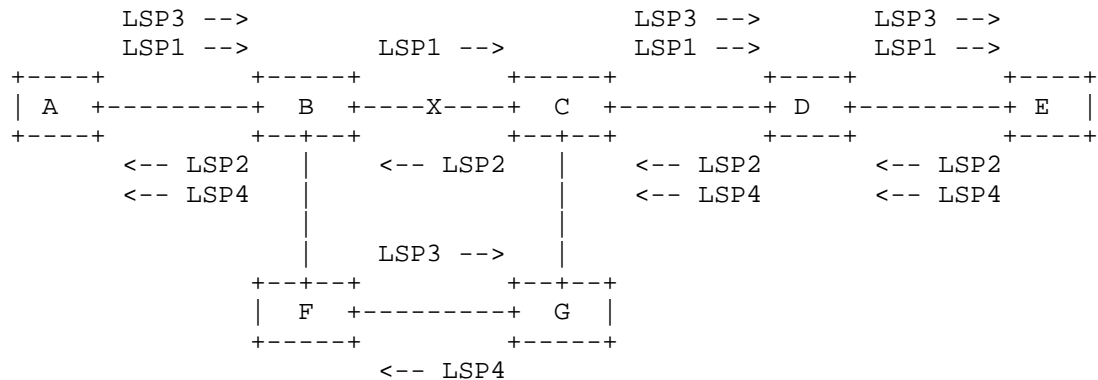


Figure 2: Restoration LSP Set-up After Link Failure

As shown in Figure 2, protected LSPs LSP1 and LSP2 are an associated LSP pair, similarly restoration LSPs LSP3 and LSP4 are an associated LSP pair, both pairs belong to the same associated bidirectional LSP and carry identical (Extended) ASSOCIATION Objects. In this example, mid-point node D may mistakenly associate LSP1 with reverse LSP4 instead of reverse LSP3 due to the matching (Extended) ASSOCIATION Objects. This may cause the bidirectional LSP to become non co-routed. Since a reverse LSP reflects the bypass tunnel assignment received in the forward LSP, this can also lead to undesired bypass tunnel assignments.

4. Signaling Procedure

4.1. Bidirectional LSP Fast Reroute

The mechanisms defined in [GMPLS-FRR] are used for fast reroute of both single-sided and double-sided associated bidirectional LSPs as following.

- o As described in [GMPLS-FRR], BYPASS_ASSIGNMENT subobject is signaled in the RRO of the Path message to co-ordinate bypass tunnel assignment between the forward and reverse direction PLR nodes. A BYPASS_ASSIGNMENT subobject MUST be added by the forward direction PLR node in the Path message of the forward LSP to indicate the bypass tunnel assigned.
- o The forward direction PLR node always initiates the bypass tunnel assignment for the forward LSP. The reverse direction PLR (forward direction LSP Merge Point (MP)) node simply reflects the bypass tunnel assignment for the reverse direction LSP.

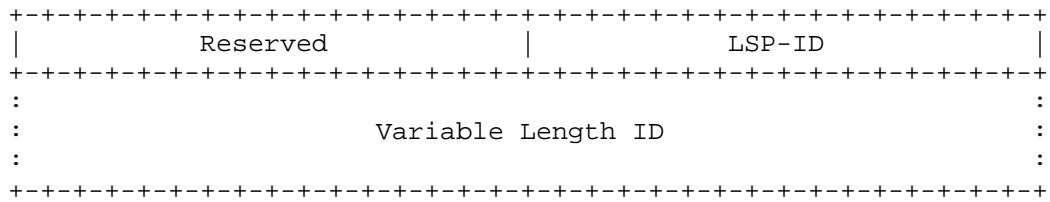


Figure 3: IPv4 Extended Association ID

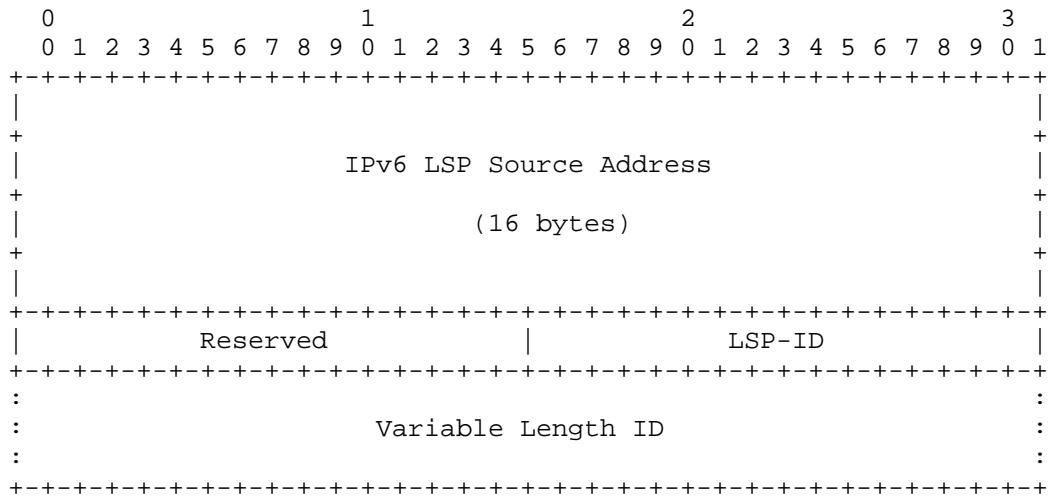


Figure 4: IPv6 Extended Association ID

LSP Source Address

IPv4/IPv6 source address of the forward LSP.

LSP-ID

16-bits LSP-ID of the forward LSP.

Variable Length ID

Variable length ID inserted by the endpoint node of the associated bidirectional LSP [RFC6780].

6. Compatibility

This document describes the procedures for fast reroute for associated bidirectional LSPs. Operators wishing to use this function SHOULD ensure that it is supported on the nodes on the LSP path.

7. Security Considerations

This document uses signaling mechanisms defined in [RFC7551] and [GMPLS-FRR] and does not introduce any additional security considerations other than already covered in [RFC7551], [GMPLS-FRR] and the MPLS/GMPLS security framework [RFC5920].

8. IANA Considerations

This document does not make any request for IANA action.

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.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP Association Object Extensions", RFC 6780, October 2012.
- [RFC7551] Zhang, F., Ed., Jing, R., and Gandhi, R., Ed., "RSVP-TE Extensions for Associated Bidirectional LSPs", RFC 7551, May 2015.
- [GMPLS-FRR] Taillon, M., Saad, T., Ed., Gandhi, R., Ed., Ali, Z., Bhatia, M., "Extensions to Resource Reservation Protocol For Fast Reroute of Traffic Engineering GMPLS LSPs", draft-ietf-teas-gmpls-lsp-fastreroute, work in progress.

9.2. Informative References

- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.
- [RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, September 2011.
- [RFC6373] Andersson, L., Berger, L., Fang, L., Bitar, N., and E. Gray, "MPLS Transport Profile (MPLS-TP) Control Plane Framework", RFC 6373, September 2011.
- [GMPLS-REST] Zhang, X., Zheng, H., Ed., Gandhi, R., Ed., Ali, Z.,

Brzozowski, P., "RSVP-TE Signaling Procedure for End-to-End GMPLS Restoration and Resource Sharing", draft-ietf-teas-gmpls-resource-sharing-proc, work in progress.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.

EMail: rgandhi@cisco.com

Himanshu Shah
Ciena

EMail: hshah@ciena.com

Jeremy Whittaker
Verizon

EMail: jeremy.whittaker@verizon.com

TEAS Working Group
Internet Draft
Intended status: Informational
Expires: November 28, 2018

Daniele Ceccarelli (Ed)
Ericsson
Young Lee (Ed)
Huawei

May 28, 2018

Framework for Abstraction and Control of Traffic Engineered Networks
draft-ietf-teas-actn-framework-15

Abstract

Traffic Engineered networks have a variety of mechanisms to facilitate the separation of the data plane and control plane. They also have a range of management and provisioning protocols to configure and activate network resources. These mechanisms represent key technologies for enabling flexible and dynamic networking. The term "Traffic Engineered network" refers to a network that uses any connection-oriented technology under the control of a distributed or centralized control plane to support dynamic provisioning of end-to-end connectivity.

Abstraction of network resources is a technique that can be applied to a single network domain or across multiple domains to create a single virtualized network that is under the control of a network operator or the customer of the operator that actually owns the network resources.

This document provides a framework for Abstraction and Control of Traffic Engineered Networks (ACTN) to support virtual network services and connectivity services.

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 November 3, 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 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. Overview.....	4
2.1. Terminology.....	5
2.2. VNS Model of ACTN.....	7
2.2.1. Customers.....	9
2.2.2. Service Providers.....	10
2.2.3. Network Operators.....	10
3. ACTN Base Architecture.....	10
3.1. Customer Network Controller.....	12
3.2. Multi-Domain Service Coordinator.....	13
3.3. Provisioning Network Controller.....	13
3.4. ACTN Interfaces.....	14
4. Advanced ACTN Architectures.....	15
4.1. MDSC Hierarchy.....	15
4.2. Functional Split of MDSC Functions in Orchestrators.....	16
5. Topology Abstraction Methods.....	17
5.1. Abstraction Factors.....	17
5.2. Abstraction Types.....	18
5.2.1. Native/White Topology.....	18
5.2.2. Black Topology.....	19
5.2.3. Grey Topology.....	20
5.3. Methods of Building Grey Topologies.....	21

5.3.1. Automatic Generation of Abstract Topology by Configuration.....	21
5.3.2. On-demand Generation of Supplementary Topology via Path Compute Request/Reply.....	21
5.4. Hierarchical Topology Abstraction Example.....	22
5.5. VN Recursion with Network Layers.....	24
6. Access Points and Virtual Network Access Points.....	25
6.1. Dual-Homing Scenario.....	27
7. Advanced ACTN Application: Multi-Destination Service.....	28
7.1. Pre-Planned End Point Migration.....	29
7.2. On the Fly End-Point Migration.....	30
8. Manageability Considerations.....	30
8.1. Policy.....	31
8.2. Policy Applied to the Customer Network Controller.....	32
8.3. Policy Applied to the Multi-Domain Service Coordinator...	32
8.4. Policy Applied to the Provisioning Network Controller....	32
9. Security Considerations.....	33
9.1. CNC-MDSC Interface (CMI).....	34
9.2. MDSC-PNC Interface (MPI).....	34
10. IANA Considerations.....	34
11. References.....	35
11.1. Informative References.....	35
12. Contributors.....	36
Authors' Addresses.....	37
APPENDIX A - Example of MDSC and PNC Functions Integrated in A Service/Network Orchestrator.....	37

1. Introduction

The term "Traffic Engineered network" refers to a network that uses any connection-oriented technology under the control of a distributed or centralized control plane to support dynamic provisioning of end-to-end connectivity. Traffic Engineered (TE) networks have a variety of mechanisms to facilitate the separation of data plane and control plane including distributed signaling for path setup and protection, centralized path computation for planning and traffic engineering, and a range of management and provisioning protocols to configure and activate network resources. These mechanisms represent key technologies for enabling flexible and dynamic networking. Some examples of networks that are in scope of this definition are optical networks, Multiprotocol Label Switching (MPLS) Transport Profile (MPLS-TP) networks [RFC5654], and MPLS-TE networks [RFC2702].

One of the main drivers for Software Defined Networking (SDN) [RFC7149] is a decoupling of the network control plane from the data

plane. This separation has been achieved for TE networks with the development of MPLS/GMPLS [RFC3945] and the Path Computation Element (PCE) [RFC4655]. One of the advantages of SDN is its logically centralized control regime that allows a global view of the underlying networks. Centralized control in SDN helps improve network resource utilization compared with distributed network control. For TE-based networks, a PCE may serve as a logically centralized path computation function.

This document describes a set of management and control functions used to operate one or more TE networks to construct virtual networks that can be presented to customers and that are built from abstractions of the underlying TE networks. For example, a link in the customer's network is constructed from a path or collection of paths in the underlying networks. We call this set of functions "Abstraction and Control of Traffic Engineered Networks" (ACTN).

2. Overview

Three key aspects that need to be solved by SDN are:

- . Separation of service requests from service delivery so that the configuration and operation of a network is transparent from the point of view of the customer, but remains responsive to the customer's services and business needs.
- . Network abstraction: As described in [RFC7926], abstraction is the process of applying policy to a set of information about a TE network to produce selective information that represents the potential ability to connect across the network. The process of abstraction presents the connectivity graph in a way that is independent of the underlying network technologies, capabilities, and topology so that the graph can be used to plan and deliver network services in a uniform way
- . Coordination of resources across multiple independent networks and multiple technology layers to provide end-to-end services regardless of whether the networks use SDN or not.

As networks evolve, the need to provide support for distinct services, separated service orchestration, and resource abstraction have emerged as key requirements for operators. In order to support multiple customers each with its own view of and control of the server network, a network operator needs to partition (or "slice") or manage sharing of the network resources. Network slices can be assigned to each customer for guaranteed usage which is a step further than shared use of common network resources.

Furthermore, each network represented to a customer can be built from virtualization of the underlying networks so that, for example, a link in the customer's network is constructed from a path or collection of paths in the underlying network.

ACTN can facilitate virtual network operation via the creation of a single virtualized network or a seamless service. This supports operators in viewing and controlling different domains (at any dimension: applied technology, administrative zones, or vendor-specific technology islands) and presenting virtualized networks to their customers.

The ACTN framework described in this document facilitates:

- . Abstraction of the underlying network resources to higher-layer applications and customers [RFC7926].
- . Virtualization of particular underlying resources, whose selection criterion is the allocation of those resources to a particular customer, application, or service [ONF-ARCH].
- . TE Network slicing of infrastructure to meet specific customers' service requirements.
- . Creation of an abstract environment allowing operators to view and control multi-domain networks as a single abstract network.
- . The presentation to customers of networks as a virtual network via open and programmable interfaces.

2.1. Terminology

The following terms are used in this document. Some of them are newly defined, some others reference existing definitions:

- . Domain: A domain [RFC4655] is any collection of network elements within a common sphere of address management or path computation responsibility. Specifically within this document we mean a part of an operator's network that is under common management (i.e., under shared operational management using the same instances of a tool and the same policies). Network elements will often be grouped into domains based on technology types, vendor profiles, and geographic proximity.
- . Abstraction: This process is defined in [RFC7926].

- . TE Network Slicing: In the context of ACTN, a TE network slice is a collection of resources that is used to establish a logically dedicated virtual network over one or more TE networks. TE network slicing allows a network operator to provide dedicated virtual networks for applications/customers over a common network infrastructure. The logically dedicated resources are a part of the larger common network infrastructures that are shared among various TE network slice instances which are the end-to-end realization of TE network slicing, consisting of the combination of physically or logically dedicated resources.
- . Node: A node is a vertex on the graph representation of a TE topology. In a physical network topology, a node corresponds to a physical network element (NE) such as a router. In an abstract network topology, a node (sometimes called an abstract node) is a representation as a single vertex of one or more physical NEs and their connecting physical connections. The concept of a node represents the ability to connect from any access to the node (a link end) to any other access to that node, although "limited cross-connect capabilities" may also be defined to restrict this functionality. Network abstraction may be applied recursively, so a node in one topology may be created by applying abstraction to the nodes in the underlying topology.
- . Link: A link is an edge on the graph representation of a TE topology. Two nodes connected by a link are said to be "adjacent" in the TE topology. In a physical network topology, a link corresponds to a physical connection. In an abstract network topology, a link (sometimes called an abstract link) is a representation of the potential to connect a pair of points with certain TE parameters (see [RFC7926] for details). Network abstraction may be applied recursively, so a link in one topology may be created by applying abstraction to the links in the underlying topology.
- . Abstract Topology: The topology of abstract nodes and abstract links presented through the process of abstraction by a lower layer network for use by a higher layer network.
- . A Virtual Network (VN) is a network provided by a service provider to a customer for the customer to use in any way it wants as though it was a physical network. There are two views of a VN as follows:

- a) The VN can be abstracted as a set of edge-to-edge links (a Type 1 VN). Each link is referred as a VN member and is formed as an end-to-end tunnel across the underlying networks. Such tunnels may be constructed by recursive slicing or abstraction of paths in the underlying networks and can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.
- b) The VN can also be abstracted as a topology of virtual nodes and virtual links (a Type 2 VN). The operator needs to map the VN to actual resource assignment, which is known as virtual network embedding. The nodes in this case include physical end points, border nodes, and internal nodes as well as abstracted nodes. Similarly the links include physical access links, inter-domain links, and intra-domain links as well as abstract links.

Clearly a Type 1 VN is a special case of a Type 2 VN.

- . Access link: A link between a customer node and a operator node.
- . Inter-domain link: A link between domains under distinct management administration.
- . Access Point (AP): An AP is a logical identifier shared between the customer and the operator used to identify an access link. The AP is used by the customer when requesting a VNS. Note that the term "TE Link Termination Point" (LTP) defined in [TE-Topo] describes the end points of links, while an AP is a common identifier for the link itself.
- . VN Access Point (VNAP): A VNAP is the binding between an AP and a given VN.
- . Server Network: As defined in [RFC7926], a server network is a network that provides connectivity for another network (the Client Network) in a client-server relationship.

2.2. VNS Model of ACTN

A Virtual Network Service (VNS) is the service agreement between a customer and operator to provide a VN. When a VN is a simple connectivity between two points, the difference between VNS and connectivity service becomes blurred. There are three types of VNS defined in this document.

- o Type 1 VNS refers to a VNS in which the customer is allowed to create and operate a Type 1 VN.
- o Type 2a and 2b VNS refer to VNSs in which the customer is allowed to create and operates a Type 2 VN. With a Type 2a VNS, the VN is statically created at service configuration time and the customer is not allowed to change the topology (e.g., by adding or deleting abstract nodes and links). A Type 2b VNS is the same as a Type 2a VNS except that the customer is allowed to make dynamic changes to the initial topology created at service configuration time.

VN Operations are functions that a customer can exercise on a VN depending on the agreement between the customer and the operator.

- o VN Creation allows a customer to request the instantiation of a VN. This could be through off-line pre-configuration or through dynamic requests specifying attributes to a Service Level Agreement (SLA) to satisfy the customer's objectives.
- o Dynamic Operations allow a customer to modify or delete the VN. The customer can further act upon the virtual network to create/modify/delete virtual links and nodes. These changes will result in subsequent tunnel management in the operator's networks.

There are three key entities in the ACTN VNS model:

- Customers
- Service Providers
- Network Operators

These entities are related in a three tier model as shown in Figure 1.

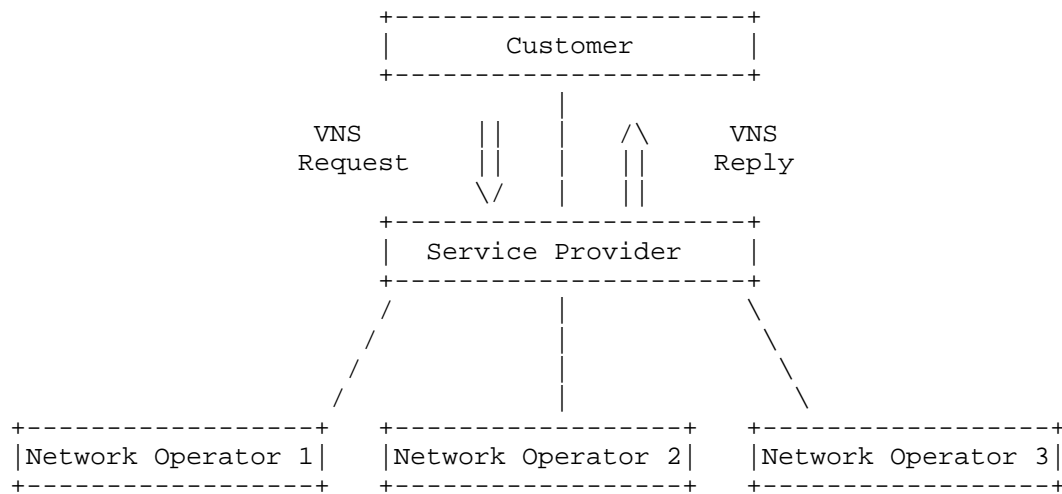


Figure 1: The Three Tier Model.

The commercial roles of these entities are described in the following sections.

2.2.1. Customers

Basic customers include fixed residential users, mobile users, and small enterprises. Each requires a small amount of resources and is characterized by steady requests (relatively time invariant). Basic customers do not modify their services themselves: if a service change is needed, it is performed by the provider as a proxy.

Advanced customers include enterprises and governments. Such customers ask for both point-to point and multipoint connectivity with high resource demands varying significantly in time. This is one of the reasons why a bundled service offering is not enough and it is desirable to provide each advanced customer with a customized virtual network service. Advanced customers may also have the ability to modify their service parameters within the scope of their virtualized environments. The primary focus of ACTN is Advanced Customers.

As customers are geographically spread over multiple network operator domains, they have to interface to multiple operators and may have to support multiple virtual network services with different underlying objectives set by the network operators. To enable these

customers to support flexible and dynamic applications they need to control their allocated virtual network resources in a dynamic fashion, and that means that they need a view of the topology that spans all of the network operators. Customers of a given service provider can in turn offer a service to other customers in a recursive way.

2.2.2. Service Providers

In the scope of ACTN, service providers deliver VNSs to their customers. Service providers may or may not own physical network resources (i.e., may or may not be network operators as described in Section 2.2.3). When a service provider is the same as the network operator, this is similar to existing VPN models applied to a single operator although it may be hard to use this approach when the customer spans multiple independent network operator domains.

When network operators supply only infrastructure, while distinct service providers interface to the customers, the service providers are themselves customers of the network infrastructure operators. One service provider may need to keep multiple independent network operators because its end-users span geographically across multiple network operator domains. In some cases, service provider is also a network operator when it owns network infrastructure on which service is provided.

2.2.3. Network Operators

Network operators are the infrastructure operators that provision the network resources and provide network resources to their customers. The layered model described in this architecture separates the concerns of network operators and customers, with service providers acting as aggregators of customer requests.

3. ACTN Base Architecture

This section provides a high-level model of ACTN showing the interfaces and the flow of control between components.

The ACTN architecture is based on a 3-tier reference model and allows for hierarchy and recursion. The main functionalities within an ACTN system are:

- . Multi-domain coordination: This function oversees the specific aspects of different domains and builds a single abstracted end-to-end network topology in order to coordinate end-to-end path computation and path/service provisioning. Domain

sequence path calculation/determination is also a part of this function.

- . Abstraction: This function provides an abstracted view of the underlying network resources for use by the customer - a customer may be the client or a higher level controller entity. This function includes network path computation based on customer service connectivity request constraints, path computation based on the global network-wide abstracted topology, and the creation of an abstracted view of network resources allocated to each customer. These operations depend on customer-specific network objective functions and customer traffic profiles.
- . Customer mapping/translation: This function is to map customer requests/commands into network provisioning requests that can be sent from the Multi-Domain Service Coordinator (MDSC) to the Provisioning Network Controller (PNC) according to business policies provisioned statically or dynamically at the Operations Support System (OSS)/ Network Management System (NMS). Specifically, it provides mapping and translation of a customer's service request into a set of parameters that are specific to a network type and technology such that network configuration process is made possible.
- . Virtual service coordination: This function translates customer service-related information into virtual network service operations in order to seamlessly operate virtual networks while meeting a customer's service requirements. In the context of ACTN, service/virtual service coordination includes a number of service orchestration functions such as multi-destination load balancing, guarantees of service quality, bandwidth and throughput. It also includes notifications for service fault and performance degradation and so forth.

The base ACTN architecture defines three controller types and the corresponding interfaces between these controllers. The following types of controller are shown in Figure 2:

- . CNC - Customer Network Controller
- . MDSC - Multi-Domain Service Coordinator
- . PNC - Provisioning Network Controller

Figure 2 also shows the following interfaces:

- . CMI - CNC-MDSC Interface
- . MPI - MDSC-PNC Interface

. SBI - Southbound Interface

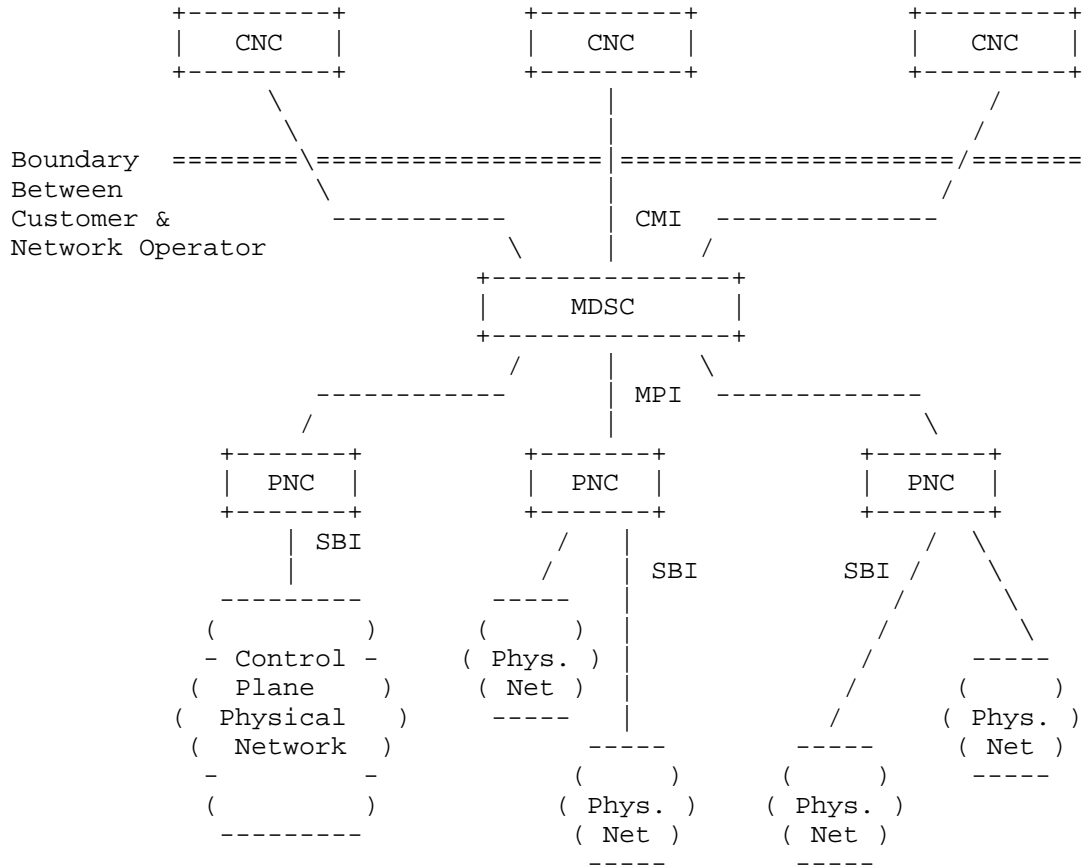


Figure 2: ACTN Base Architecture

Note that this is a functional architecture: an implementation and deployment might collocate one or more of the functional components. Figure 2 shows a case where service provider is also a network operator.

3.1. Customer Network Controller

A Customer Network Controller (CNC) is responsible for communicating a customer's VNS requirements to the network operator over the CNC-MDSC Interface (CMI). It has knowledge of the end-points associated

with the VNS (expressed as APs), the service policy, and other QoS information related to the service.

As the Customer Network Controller directly interfaces to the applications, it understands multiple application requirements and their service needs. The capability of a CNC beyond its CMI role is outside the scope of ACTN and may be implemented in different ways. For example, the CNC may in fact be a controller or part of a controller in the customer's domain, or the CNC functionality could also be implemented as part of a service provider's portal.

3.2. Multi-Domain Service Coordinator

A Multi-Domain Service Coordinator (MDSC) is a functional block that implements all of the ACTN functions listed in Section 3 and described further in Section 4.2. Two functions of the MDSC, namely, multi-domain coordination and virtualization/abstraction are referred to as network-related functions while the other two functions, namely, customer mapping/translation and virtual service coordination are referred to as service-related functions. The MDSC sits at the center of the ACTN model between the CNC that issues connectivity requests and the Provisioning Network Controllers (PNCs) that manage the network resources.

The key point of the MDSC (and of the whole ACTN framework) is detaching the network and service control from underlying technology to help the customer express the network as desired by business needs. The MDSC envelopes the instantiation of the right technology and network control to meet business criteria. In essence it controls and manages the primitives to achieve functionalities as desired by the CNC.

In order to allow for multi-domain coordination a 1:N relationship must be allowed between MDSCs and PNCs.

In addition to that, it could also be possible to have an M:1 relationship between MDSCs and PNC to allow for network resource partitioning/sharing among different customers not necessarily connected to the same MDSC (e.g., different service providers) but all using the resources of a common network infrastructure operator.

3.3. Provisioning Network Controller

The Provisioning Network Controller (PNC) oversees configuring the network elements, monitoring the topology (physical or virtual) of

the network, and collecting information about the topology (either raw or abstracted).

The PNC functions can be implemented as part of an SDN domain controller, a Network Management System (NMS), an Element Management System (EMS), an active PCE-based controller [Centralized] or any other means to dynamically control a set of nodes and implementing a north bound interface from the standpoint of the nodes (which is out of the scope of this document). A PNC domain includes all the resources under the control of a single PNC. It can be composed of different routing domains and administrative domains, and the resources may come from different layers. The interconnection between PNC domains is illustrated in Figure 3.

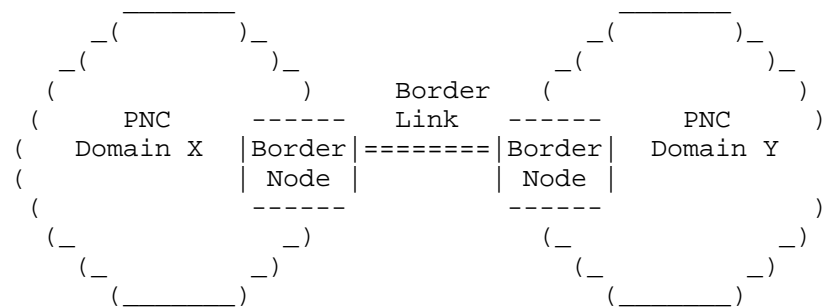


Figure 3: PNC Domain Borders

3.4. ACTN Interfaces

Direct customer control of transport network elements and virtualized services is not a viable proposition for network operators due to security and policy concerns. Therefore, the network has to provide open, programmable interfaces, through which customer applications can create, replace and modify virtual network resources and services in an interactive, flexible and dynamic fashion.

Three interfaces exist in the ACTN architecture as shown in Figure 2.

- . CMI: The CNC-MDSC Interface (CMI) is an interface between a CNC and an MDSC. The CMI is a business boundary between customer and network operator. It is used to request a VNS for an application. All service-related information is conveyed over this interface (such as the VNS type, topology, bandwidth, and

service constraints). Most of the information over this interface is agnostic of the technology used by network operators, but there are some cases (e.g., access link configuration) where it is necessary to specify technology-specific details.

- . MPI: The MDSC-PNC Interface (MPI) is an interface between an MDSC and a PNC. It communicates requests for new connectivity or for bandwidth changes in the physical network. In multi-domain environments, the MDSC needs to communicate with multiple PNCs each responsible for control of a domain. The MPI presents an abstracted topology to the MDSC hiding technology specific aspects of the network and hiding topology according to policy.
- . SBI: The Southbound Interface (SBI) is out of scope of ACTN. Many different SBIs have been defined for different environments, technologies, standards organizations, and vendors. It is shown in Figure 3 for reference reason only.

4. Advanced ACTN Architectures

This section describes advanced configurations of the ACTN architecture.

4.1. MDSC Hierarchy

A hierarchy of MDSCs can be foreseen for many reasons, among which are scalability, administrative choices, or putting together different layers and technologies in the network. In the case where there is a hierarchy of MDSCs, we introduce the terms higher-level MDSC (MDSC-H) and lower-level MDSC (MDSC-L). The interface between them is a recursion of the MPI. An implementation of an MDSC-H makes provisioning requests as normal using the MPI, but an MDSC-L must be able to receive requests as normal at the CMI and also at the MPI. The hierarchy of MDSCs can be seen in Figure 4.

Another implementation choice could foresee the usage of an MDSC-L for all the PNCs related to a given technology (e.g., Internet Protocol (IP)/Multiprotocol Label Switching (MPLS)) and a different MDSC-L for the PNCs related to another technology (e.g., Optical Transport Network (OTN)/Wavelength Division Multiplexing (WDM)) and an MDSC-H to coordinate them.

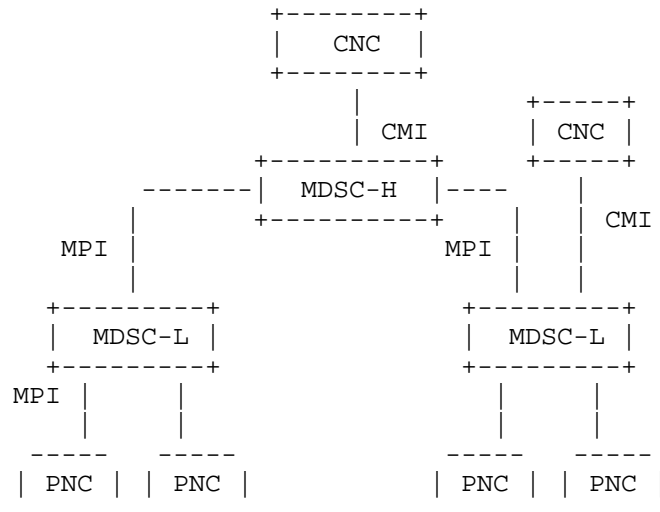
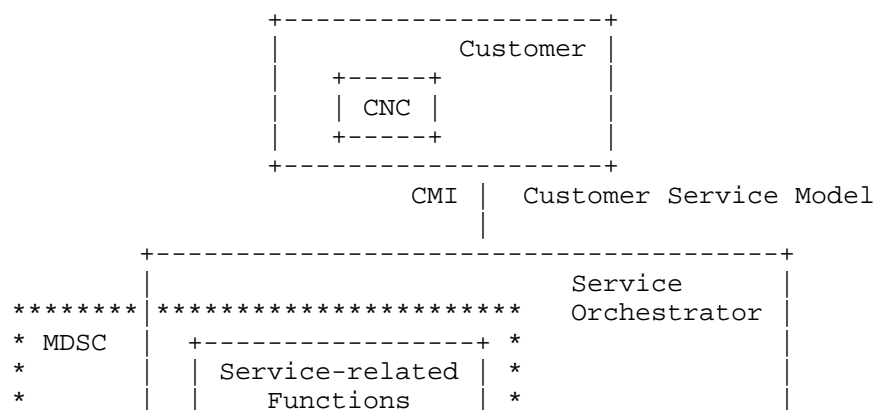


Figure 4: MDSC Hierarchy

The hierarchy of MDSC can be recursive, where an MDSC-H is in turn an MDSC-L to a higher level MDSC-H.

4.2. Functional Split of MDSC Functions in Orchestrators

An implementation choice could separate the MDSC functions into two groups, one group for service-related functions and the other for network-related functions. This enables the implementation of a service orchestrator that provides the service-related functions of the MDSC and a network orchestrator that provides the network-related functions of the MDSC. This split is consistent with the Yet Another Next Generation (YANG) service model architecture described in [Service-YANG]. Figure 5 depicts this and shows how the ACTN interfaces may map to YANG models.



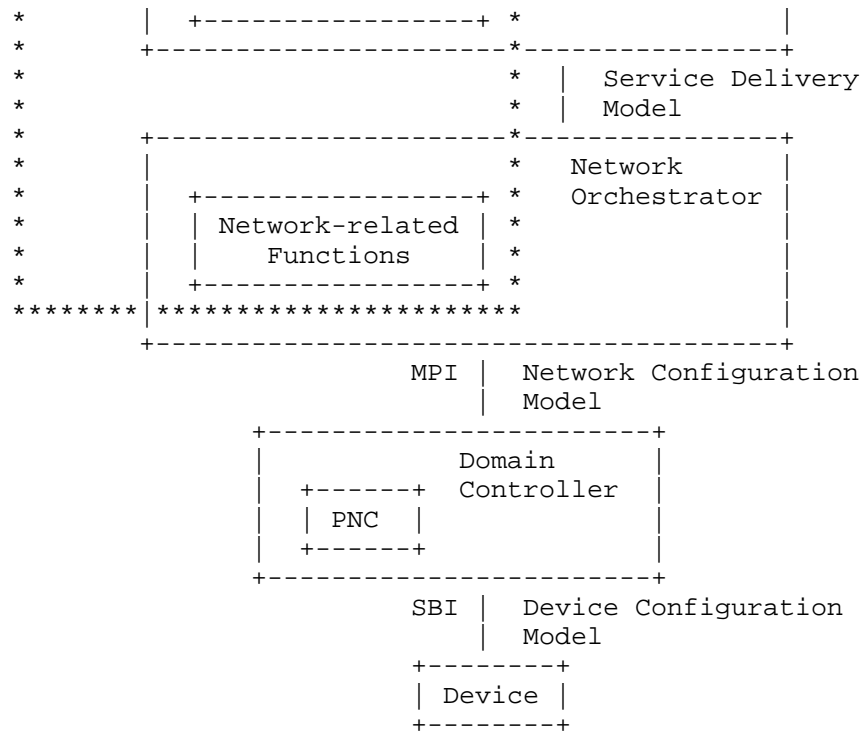


Figure 5: ACTN Architecture in the Context of the YANG Service Models

5. Topology Abstraction Methods

Topology abstraction is described in [RFC7926]. This section discusses topology abstraction factors, types, and their context in the ACTN architecture.

Abstraction in ACTN is performed by the PNC when presenting available topology to the MDSC, or by an MDSC-L when presenting topology to an MDSC-H. This function is different to the creation of a VN (and particularly a Type 2 VN) which is not abstraction but construction of virtual resources.

5.1. Abstraction Factors

As discussed in [RFC7926], abstraction is tied with policy of the networks. For instance, per an operational policy, the PNC would not provide any technology specific details (e.g., optical parameters for Wavelength Switched Optical Network (WSO) in the abstract topology it provides to the MDSC. Similarly, policy of the

networks may determine the abstraction type as described in Section 5.2.

There are many factors that may impact the choice of abstraction:

- Abstraction depends on the nature of the underlying domain networks. For instance, packet networks may be abstracted with fine granularity while abstraction of optical networks depends on the switching units (such as wavelengths) and the end-to-end continuity and cross-connect limitations within the network.
- Abstraction also depends on the capability of the PNCs. As abstraction requires hiding details of the underlying network resources, the PNC's capability to run algorithms impacts the feasibility of abstraction. Some PNC may not have the ability to abstract native topology while other PNCs may have the ability to use sophisticated algorithms.
- Abstraction is a tool that can improve scalability. Where the native network resource information is of large size there is a specific scaling benefit to abstraction.
- The proper abstraction level may depend on the frequency of topology updates and vice versa.
- The nature of the MDSC's support for technology-specific parameters impacts the degree/level of abstraction. If the MDSC is not capable of handling such parameters then a higher level of abstraction is needed.
- In some cases, the PNC is required to hide key internal topological data from the MDSC. Such confidentiality can be achieved through abstraction.

5.2. Abstraction Types

This section defines the following three types of topology abstraction:

- . Native/White Topology (Section 5.2.1)
- . Black Topology (Section 5.2.2)
- . Grey Topology (Section 5.2.3)

5.2.1. Native/White Topology

This is a case where the PNC provides the actual network topology to the MDSC without any hiding or filtering of information, i.e., no

abstraction is performed. In this case, the MDSC has the full knowledge of the underlying network topology and can operate on it directly.

5.2.2. Black Topology

A black topology replaces a full network with a minimal representation of the edge-to-edge topology without disclosing any node internal connectivity information. The entire domain network may be abstracted as a single abstract node with the network's access/egress links appearing as the ports to the abstract node and the implication that any port can be 'cross-connected' to any other. Figure 6 depicts a native topology with the corresponding black topology with one virtual node and inter-domain links. In this case, the MDSC has to make a provisioning request to the PNCs to establish the port-to-port connection. If there is a large number of inter-connected domains, this abstraction method may impose a heavy coordination load at the MDSC level in order to find an optimal end-to-end path since the abstraction hides so much information that it is not possible to determine whether an end-to-end path is feasible without asking each PNC to set up each path fragment. For this reason, the MPI might need to be enhanced to allow the PNCs to be queried for the practicality and characteristics of paths across the abstract node.

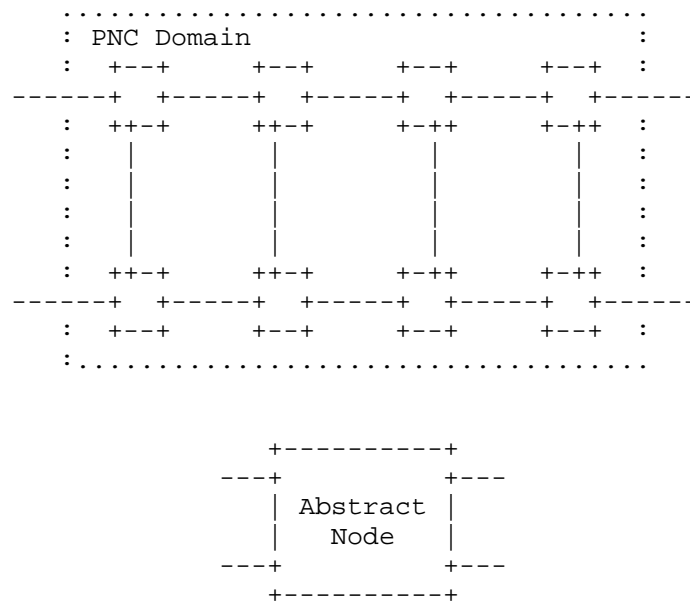


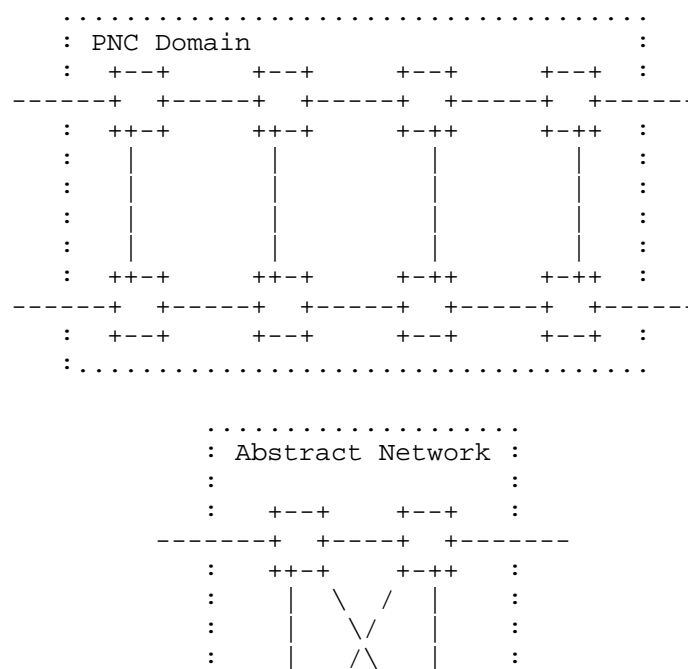
Figure 6: Native Topology with Corresponding Black Topology Expressed as an Abstract Node

5.2.3. Grey Topology

A grey topology represents a compromise between black and white topologies from a granularity point of view. In this case, the PNC exposes an abstract topology containing all PNC domains border nodes and an abstraction of the connectivity between those border nodes. This abstraction may contain either physical or abstract nodes/links.

Two types of grey topology are identified:

- . In a type A grey topology, border nodes are connected by a full mesh of TE links (see Figure 7).
- . In a type B grey topology, border nodes are connected over a more detailed network comprising internal abstract nodes and abstracted links. This mode of abstraction supplies the MDSC with more information about the internals of the PNC domain and allows it to make more informed choices about how to route connectivity over the underlying network.



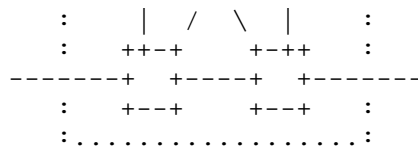


Figure 7: Native Topology with Corresponding Grey Topology

5.3. Methods of Building Grey Topologies

This section discusses two different methods of building a grey topology:

- . Automatic generation of abstract topology by configuration (Section 5.3.1)
- . On-demand generation of supplementary topology via path computation request/reply (Section 5.3.2)

5.3.1. Automatic Generation of Abstract Topology by Configuration

Automatic generation is based on the abstraction/summarization of the whole domain by the PNC and its advertisement on the MPI. The level of abstraction can be decided based on PNC configuration parameters (e.g., "provide the potential connectivity between any PE and any ASBR in an MPLS-TE network").

Note that the configuration parameters for this abstract topology can include available bandwidth, latency, or any combination of defined parameters. How to generate such information is beyond the scope of this document.

This abstract topology may need to be periodically or incrementally updated when there is a change in the underlying network or the use of the network resources that make connectivity more or less available.

5.3.2. On-demand Generation of Supplementary Topology via Path Compute Request/Reply

While abstract topology is generated and updated automatically by configuration as explained in Section 5.3.1, additional supplementary topology may be obtained by the MDSC via a path compute request/reply mechanism.

The abstract topology advertisements from PNCs give the MDSC the border node/link information for each domain. Under this scenario,

when the MDSC needs to create a new VN, the MDSC can issue path computation requests to PNCs with constraints matching the VN request as described in [ACTN-YANG]. An example is provided in Figure 8, where the MDSC is creating a P2P VN between AP1 and AP2. The MDSC could use two different inter-domain links to get from domain X to domain Y, but in order to choose the best end-to-end path it needs to know what domain X and Y can offer in terms of connectivity and constraints between the PE nodes and the border nodes.

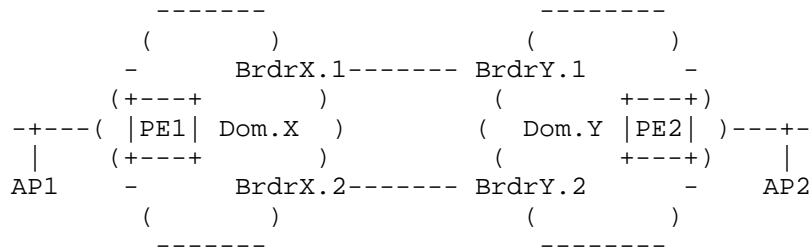
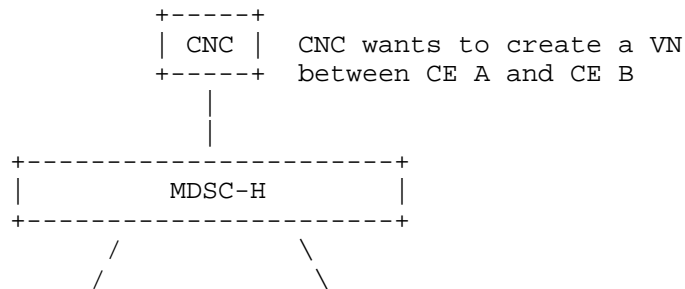


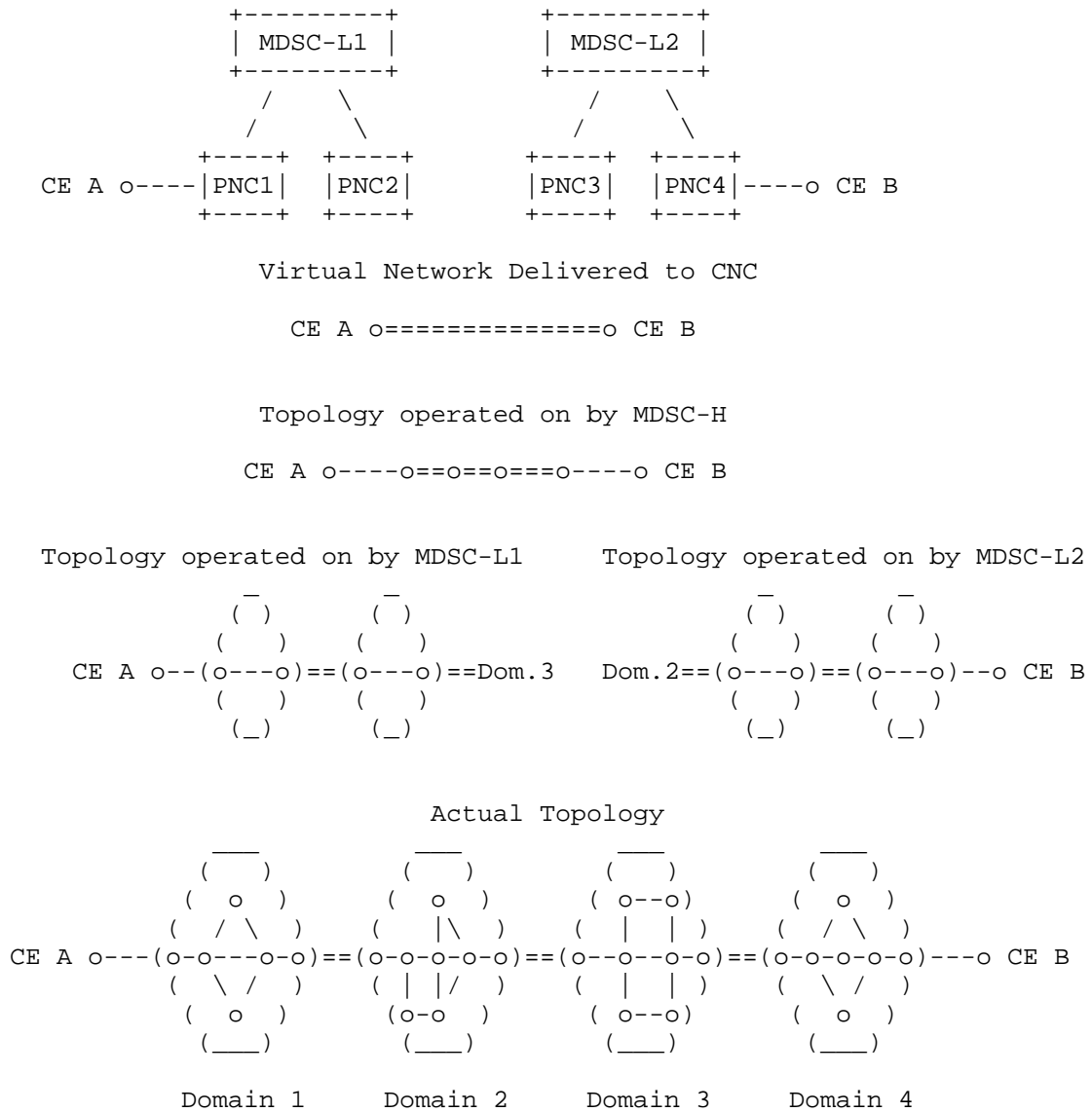
Figure 8: A Multi-Domain Example

The MDSC issues a path computation request to PNC.X asking for potential connectivity between PE1 and border node BrdrX.1 and between PE1 and BrdrX.2 with related objective functions and TE metric constraints. A similar request for connectivity from the border nodes in domain Y to PE2 will be issued to PNC.Y. The MDSC merges the results to compute the optimal end-to-end path including the inter domain links. The MDSC can use the result of this computation to request the PNCs to provision the underlying networks, and the MDSC can then use the end-to-end path as a virtual link in the VN it delivers to the customer.

5.4. Hierarchical Topology Abstraction Example

This section illustrates how topology abstraction operates in different levels of a hierarchy of MDSCs as shown in Figure 9.





Where

o is a node
 --- is a link
 === border link

Figure 9: Illustration of Hierarchical Topology Abstraction

In the example depicted in Figure 9, there are four domains under control of PNCs PNC1, PNC2, PNC3, and PNC4. MDSC-L1 controls PNC1

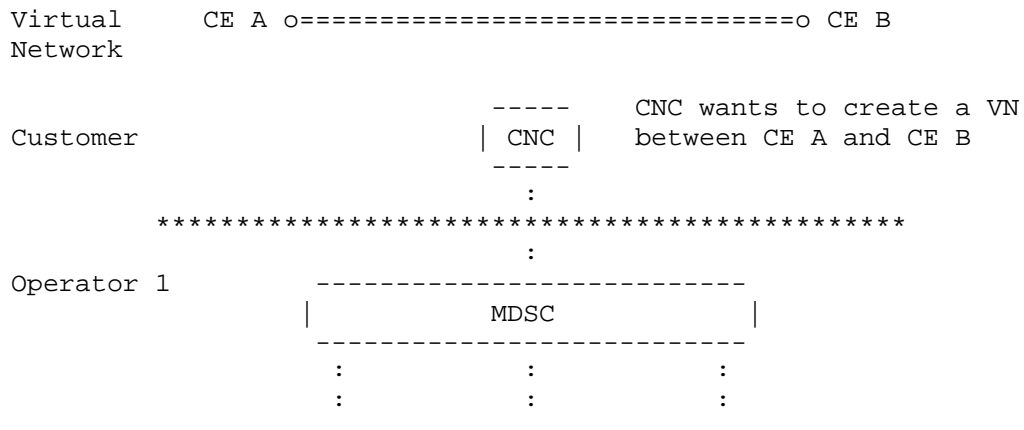
and PNC2 while MDSC-L2 controls PNC3 and PNC4. Each of the PNCs provides a grey topology abstraction that presents only border nodes and links across and outside the domain. The abstract topology MDSC-L1 that operates is a combination of the two topologies from PNC1 and PNC2. Likewise, the abstract topology that MDSC-L2 operates is shown in Figure 9. Both MDSC-L1 and MDSC-L2 provide a black topology abstraction to MDSC-H in which each PNC domain is presented as a single virtual node. MDSC-H combines these two topologies to create the abstraction topology on which it operates. MDSC-H sees the whole four domain networks as four virtual nodes connected via virtual links.

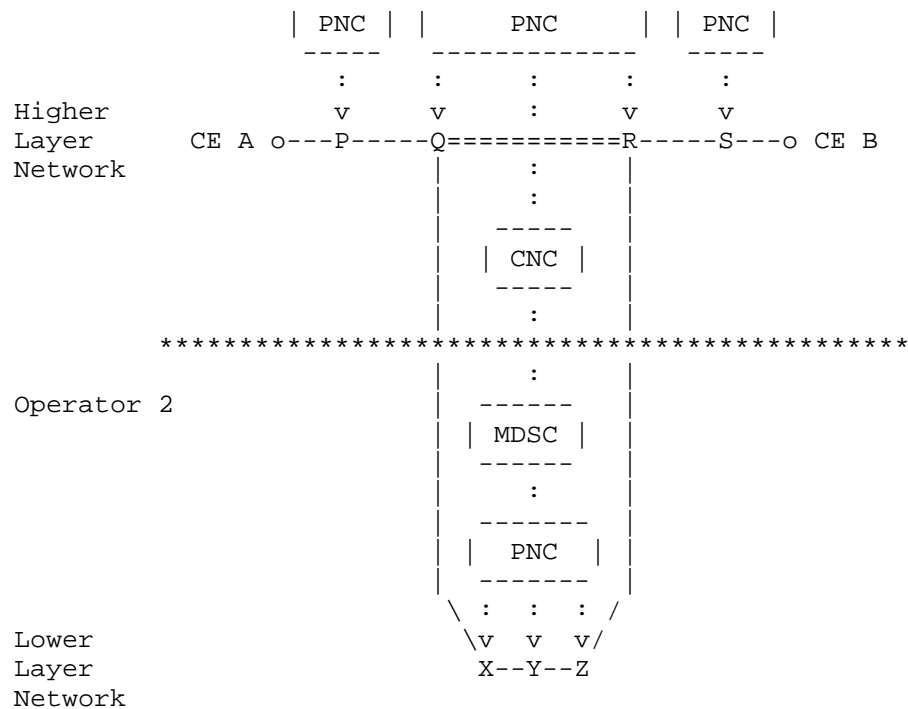
5.5. VN Recursion with Network Layers

In some cases the VN supplied to a customer may be built using resources from different technology layers operated by different operators. For example, one operator may run a packet TE network and use optical connectivity provided by another operator.

As shown in Figure 10, a customer asks for end-to-end connectivity between CE A and CE B, a virtual network. The customer's CNC makes a request to Operator 1's MDSC. The MDSC works out which network resources need to be configured and sends instructions to the appropriate PNCs. However, the link between Q and R is a virtual link supplied by Operator 2: Operator 1 is a customer of Operator 2.

To support this, Operator 1 has a CNC that communicates to Operator 2's MDSC. Note that Operator 1's CNC in Figure 10 is a functional component that does not dictate implementation: it may be embedded in a PNC.





Where

--- is a link

=== is a virtual link

Figure 10: VN recursion with Network Layers

6. Access Points and Virtual Network Access Points

In order to map identification of connections between the customer's sites and the TE networks and to scope the connectivity requested in the VNS, the CNC and the MDSC refer to the connections using the Access Point (AP) construct as shown in Figure 11.

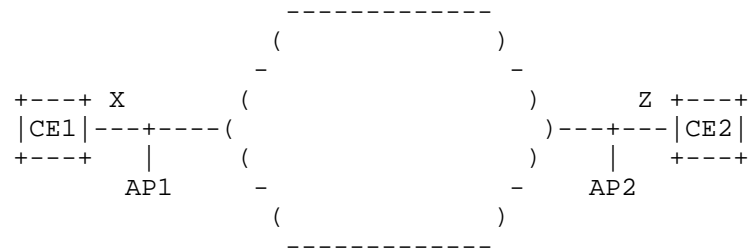


Figure 11: Customer View of APs

Let's take as an example a scenario shown in Figure 11. CE1 is connected to the network via a 10 Gbps link and CE2 via a 40 Gbps link. Before the creation of any VN between AP1 and AP2 the customer view can be summarized as shown in Table 1.

+-----+-----+			
End Point		Access Link Bandwidth	
+-----+-----+			
AP id	CE,port	MaxResBw	AvailableBw
+-----+-----+			
AP1	CE1,portX	10 Gbps	10 Gbps
+-----+-----+			
AP2	CE2,portZ	40 Gbps	40 Gbps
+-----+-----+			

Table 1: AP - Customer View

On the other hand, what the operator sees is shown in Figure 12.

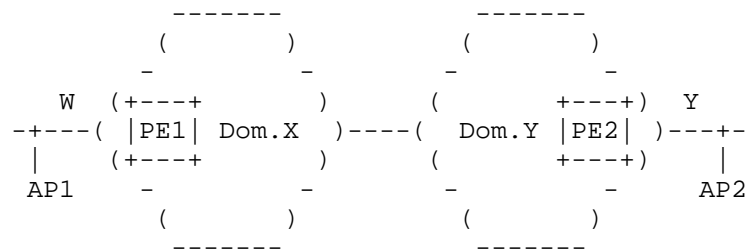


Figure 12: Operator view of the AP

Which results in a summarization as shown in Table 2.

End Point		Access Link Bandwidth	
AP id	PE,port	MaxResBw	AvailableBw
AP1	PE1,portW	10 Gbps	10 Gbps
AP2	PE2,portY	40 Gbps	40 Gbps

Table 2: AP - Operator View

A Virtual Network Access Point (VNAP) needs to be defined as binding between an AP and a VN. It is used to allow for different VNs to start from the same AP. It also allows for traffic engineering on the access and/or inter-domain links (e.g., keeping track of bandwidth allocation). A different VNAP is created on an AP for each VN.

In this simple scenario we suppose we want to create two virtual networks. The first with VN identifier 9 between AP1 and AP2 with bandwidth of 1 Gbps, while the second with VN identifier 5, again between AP1 and AP2 and with bandwidth 2 Gbps.

The operator view would evolve as shown in Table 3.

End Point		Access Link/VNAP Bw	
AP/VNAPid	PE,port	MaxResBw	AvailableBw
AP1	PE1,portW	10 Gbps	7 Gbps
-VNAP1.9		1 Gbps	N.A.
-VNAP1.5		2 Gbps	N.A.
AP2	PE2,portY	40 Gbps	37 Gbps
-VNAP2.9		1 Gbps	N.A.
-VNAP2.5		2 Gbps	N.A.

Table 3: AP and VNAP - Operator View after VNS Creation

6.1. Dual-Homing Scenario

Often there is a dual homing relationship between a CE and a pair of PEs. This case needs to be supported by the definition of VN, APs, and VNAPs. Suppose CE1 connected to two different PEs in the

operator domain via AP1 and AP2 and that the customer needs 5 Gbps of bandwidth between CE1 and CE2. This is shown in Figure 12.

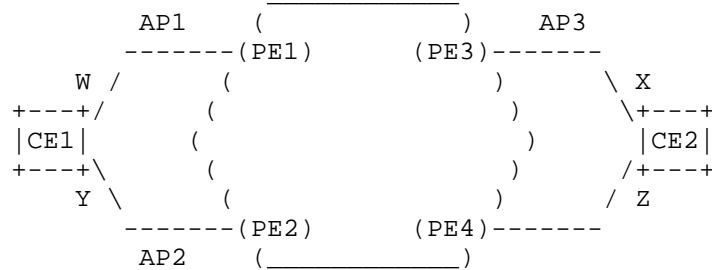


Figure 12: Dual-Homing Scenario

In this case, the customer will request for a VN between AP1, AP2, and AP3 specifying a dual homing relationship between AP1 and AP2. As a consequence no traffic will flow between AP1 and AP2. The dual homing relationship would then be mapped against the VNAPs (since other independent VNs might have AP1 and AP2 as end points).

The customer view would be shown in Table 4.

	End Point	Access Link/VNAP Bw		
AP/VNAPid	CE,port	MaxResBw	AvailableBw	Dual Homing
AP1 -VNAP1.9	CE1,portW	10 Gbps 5 Gbps	5 Gbps N.A.	VNAP2.9
AP2 -VNAP2.9	CE1,portY	40 Gbps 5 Gbps	35 Gbps N.A.	VNAP1.9
AP3 -VNAP3.9	CE2,portX	50 Gbps 5 Gbps	45 Gbps N.A.	NONE

Table 4: Dual-Homing - Customer View after VN Creation

7. Advanced ACTN Application: Multi-Destination Service

A further advanced application of ACTN is in the case of Data Center selection, where the customer requires the Data Center selection to be based on the network status; this is referred to as Multi-Destination in [ACTN-REQ]. In terms of ACTN, a CNC could request a

VNS between a set of source APs and destination APs and leave it up to the network (MDSC) to decide which source and destination access points to be used to set up the VNS. The candidate list of source and destination APs is decided by a CNC (or an entity outside of ACTN) based on certain factors which are outside the scope of ACTN.

Based on the AP selection as determined and returned by the network (MDSC), the CNC (or an entity outside of ACTN) should further take care of any subsequent actions such as orchestration or service setup requirements. These further actions are outside the scope of ACTN.

Consider a case as shown in Figure 14, where three data centers are available, but the customer requires the data center selection to be based on the network status and the connectivity service setup between the AP1 (CE1) and one of the destination APs (AP2 (DC-A), AP3 (DC-B), and AP4 (DC-C)). The MDSC (in coordination with PNCs) would select the best destination AP based on the constraints, optimization criteria, policies, etc., and setup the connectivity service (virtual network).

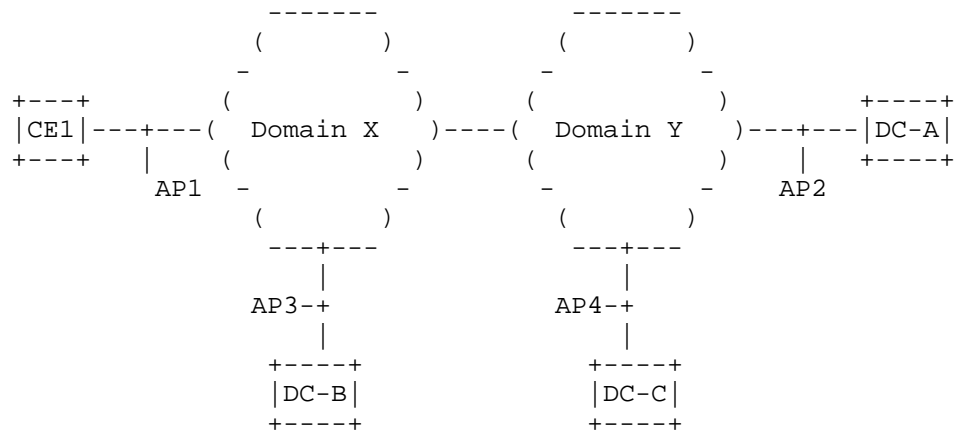


Figure 14: End-Point Selection Based on Network Status

7.1. Pre-Planned End Point Migration

Furthermore, in case of Data Center selection, customer could request for a backup DC to be selected, such that in case of failure, another DC site could provide hot stand-by protection. As shown in Figure 15 DC-C is selected as a backup for DC-A. Thus, the VN should be setup by the MDSC to include primary connectivity

between AP1 (CE1) and AP2 (DC-A) as well as protection connectivity between AP1 (CE1) and AP4 (DC-C).

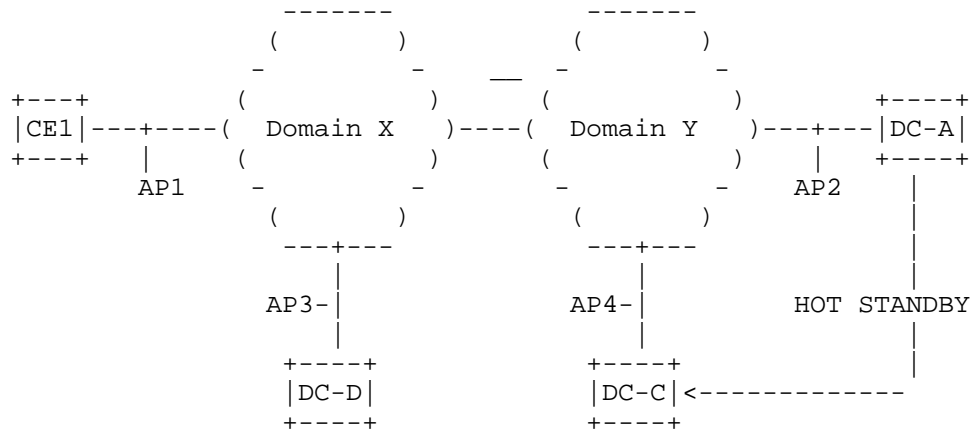


Figure 15: Pre-planned End-Point Migration

7.2. On the Fly End-Point Migration

Compared to pre-planned end point migration, on the fly end point selection is dynamic in that the migration is not pre-planned but decided based on network condition. Under this scenario, the MDSC would monitor the network (based on the VN Service-level Agreement (SLA) and notify the CNC in case where some other destination AP would be a better choice based on the network parameters. The CNC should instruct the MDSC when it is suitable to update the VN with the new AP if it is required.

8. Manageability Considerations

The objective of ACTN is to manage traffic engineered resources, and provide a set of mechanisms to allow customers to request virtual connectivity across server network resources. ACTN supports multiple customers each with its own view of and control of a virtual network built on the server network, the network operator will need to partition (or "slice") their network resources, and manage the resources accordingly.

The ACTN platform will, itself, need to support the request, response, and reservations of client and network layer connectivity. It will also need to provide performance monitoring and control of traffic engineered resources. The management requirements may be categorized as follows:

- . Management of external ACTN protocols
- . Management of internal ACTN interfaces/protocols
- . Management and monitoring of ACTN components
- . Configuration of policy to be applied across the ACTN system

The ACTN framework and interfaces are defined to enable traffic engineering for virtual network services and connectivity services. Network operators may have other Operations, Administration, and Maintenance (OAM) tasks for service fulfillment, optimization, and assurance beyond traffic engineering. The realization of OAM beyond abstraction and control of traffic engineered networks is not considered in this document.

8.1. Policy

Policy is an important aspect of ACTN control and management. Policies are used via the components and interfaces, during deployment of the service, to ensure that the service is compliant with agreed policy factors and variations (often described in SLAs), these include, but are not limited to: connectivity, bandwidth, geographical transit, technology selection, security, resilience, and economic cost.

Depending on the deployment of the ACTN architecture, some policies may have local or global significance. That is, certain policies may be ACTN component specific in scope, while others may have broader scope and interact with multiple ACTN components. Two examples are provided below:

- o A local policy might limit the number, type, size, and scheduling of virtual network services a customer may request via its CNC. This type of policy would be implemented locally on the MDSC.
- o A global policy might constrain certain customer types (or specific customer applications) to only use certain MDSCs, and be restricted to physical network types managed by the PNCs. A global policy agent would govern these types of policies.

The objective of this section is to discuss the applicability of ACTN policy: requirements, components, interfaces, and examples. This section provides an analysis and does not mandate a specific method for enforcing policy, or the type of policy agent that would

be responsible for propagating policies across the ACTN components. It does highlight examples of how policy may be applied in the context of ACTN, but it is expected further discussion in an applicability or solution specific document, will be required.

8.2. Policy Applied to the Customer Network Controller

A virtual network service for a customer application will be requested by the CNC. The request will reflect the application requirements and specific service needs, including bandwidth, traffic type and survivability. Furthermore, application access and type of virtual network service requested by the CNC, will be need adhere to specific access control policies.

8.3. Policy Applied to the Multi-Domain Service Coordinator

A key objective of the MDSC is to support the customer's expression of the application connectivity request via its CNC as a set of desired business needs, therefore policy will play an important role.

Once authorized, the virtual network service will be instantiated via the CNC-MDSC Interface (CMI); it will reflect the customer application and connectivity requirements, and specific service transport needs. The CNC and the MDSC components will have agreed connectivity end-points; use of these end-points should be defined as a policy expression when setting up or augmenting virtual network services. Ensuring that permissible end-points are defined for CNCs and applications will require the MDSC to maintain a registry of permissible connection points for CNCs and application types.

Conflicts may occur when virtual network service optimization criteria are in competition. For example, to meet objectives for service reachability a request may require an interconnection point between multiple physical networks; however, this might break a confidentially policy requirement of specific type of end-to-end service. Thus an MDSC may have to balance a number of the constraints on a service request and between different requested services. It may also have to balance requested services with operational norms for the underlying physical networks. This balancing may be resolved using configured policy and using hard and soft policy constraints.

8.4. Policy Applied to the Provisioning Network Controller

The PNC is responsible for configuring the network elements, monitoring physical network resources, and exposing connectivity

(direct or abstracted) to the MDSC. It is therefore expected that policy will dictate what connectivity information will be exported between the PNC, via the MDSC-PNC Interface (MPI), and MDSC.

Policy interactions may arise when a PNC determines that it cannot compute a requested path from the MDSC, or notices that (per a locally configured policy) the network is low on resources (for example, the capacity on key links become exhausted). In either case, the PNC will be required to notify the MDSC, which may (again per policy) act to construct a virtual network service across another physical network topology.

Furthermore, additional forms of policy-based resource management will be required to provide virtual network service performance, security and resilience guarantees. This will likely be implemented via a local policy agent and additional protocol methods.

9. Security Considerations

The ACTN framework described in this document defines key components and interfaces for managed traffic engineered networks. Securing the request and control of resources, confidentiality of the information, and availability of function, should all be critical security considerations when deploying and operating ACTN platforms.

Several distributed ACTN functional components are required, and implementations should consider encrypting data that flows between components, especially when they are implemented at remote nodes, regardless these data flows are on external or internal network interfaces.

The ACTN security discussion is further split into two specific categories described in the following sub-sections:

- o Interface between the Customer Network Controller and Multi-Domain Service Coordinator (MDSC), CNC-MDSC Interface (CMI)
- o Interface between the Multi-Domain Service Coordinator and Provisioning Network Controller (PNC), MDSC-PNC Interface (MPI)

From a security and reliability perspective, ACTN may encounter many risks such as malicious attack and rogue elements attempting to connect to various ACTN components. Furthermore, some ACTN components represent a single point of failure and threat vector, and must also manage policy conflicts, and eavesdropping of communication between different ACTN components.

The conclusion is that all protocols used to realize the ACTN framework should have rich security features, and customer, application and network data should be stored in encrypted data stores. Additional security risks may still exist. Therefore, discussion and applicability of specific security functions and protocols will be better described in documents that are use case and environment specific.

9.1. CNC-MDSC Interface (CMI)

Data stored by the MDSC will reveal details of the virtual network services, and which CNC and customer/application is consuming the resource. The data stored must therefore be considered as a candidate for encryption.

CNC Access rights to an MDSC must be managed. The MDSC must allocate resources properly, and methods to prevent policy conflicts, resource wastage, and denial of service attacks on the MDSC by rogue CNCs, should also be considered.

The CMI will likely be an external protocol interface. Suitable authentication and authorization of each CNC connecting to the MDSC will be required, especially, as these are likely to be implemented by different organizations and on separate functional nodes. Use of the AAA-based mechanisms would also provide role-based authorization methods, so that only authorized CNC's may access the different functions of the MDSC.

9.2. MDSC-PNC Interface (MPI)

Where the MDSC must interact with multiple (distributed) PNCs, a PKI-based mechanism is suggested, such as building a TLS or HTTPS connection between the MDSC and PNCs, to ensure trust between the physical network layer control components and the MDSC. Trust anchors for the PKI can be configured to use a smaller (and potentially non-intersecting) set of trusted Certificate Authorities (CAs) than in the Web PKI.

Which MDSC the PNC exports topology information to, and the level of detail (full or abstracted), should also be authenticated, and specific access restrictions and topology views should be configurable and/or policy-based.

10. IANA Considerations

This document has no actions for IANA.

11. References

11.1. Informative References

- [RFC2702] Awduche, D., et. al., "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", IETF RFC 4655, August 2006.
- [RFC5654] Niven-Jenkins, B. (Ed.), D. Brungard (Ed.), and M. Betts (Ed.), "Requirements of an MPLS Transport Profile", RFC 5654, September 2009.
- [RFC7149] Boucadair, M. and Jacquenet, C., "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, March 2014.
- [RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.
- [RFC3945] Manning, E., et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture2", RFC 3945, October 2004.
- [ONF-ARCH] Open Networking Foundation, "SDN architecture", Issue 1.1, ONF TR-521, June 2016.
- [Centralized] Farrel, A., et al., "An Architecture for Use of PCE and PCEP in a Network with Central Control", draft-ietf-teas-pce-central-control, work in progress.
- [Service-YANG] Lee, Y., Dhody, D., and Ceccarelli, C., "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.
- [ACTN-YANG] Lee, Y., et al., "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.
- [ACTN-REQ] Lee, Y., et al., "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.
- [TE-Topo] X. Liu et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.

12. Contributors

Adrian Farrel
Old Dog Consulting
Email: adrian@olddog.co.uk

Italo Busi
Huawei
Email: Italo.Busi@huawei.com

Khuzema Pithewan
Infinera
Email: kpithewan@infinera.com

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

Luyuan Fang
eBay
Email: luyuanf@gmail.com

Diego Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
28006 Madrid, Spain
Email: diego@tid.es

Sergio Belotti
Alcatel Lucent
Via Trento, 30
Vimercate, Italy
Email: sergio.belotti@nokia.com

Daniel King
Lancaster University
Email: d.king@lancaster.ac.uk

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India
Email: dhruv.ietf@gmail.com

Gert Grammel
 Juniper Networks
 Email: ggrammel@juniper.net

Authors' Addresses

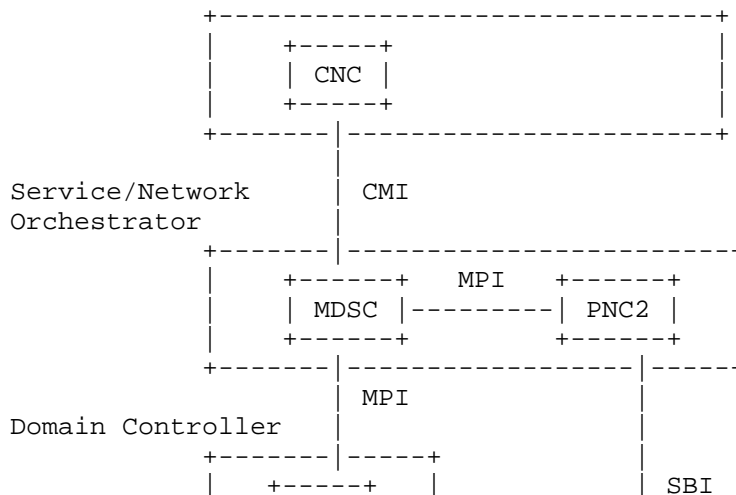
Daniele Ceccarelli
 Ericsson
 Torshamnsgatan, 48
 Stockholm, Sweden
 Email: daniele.ceccarelli@ericsson.com

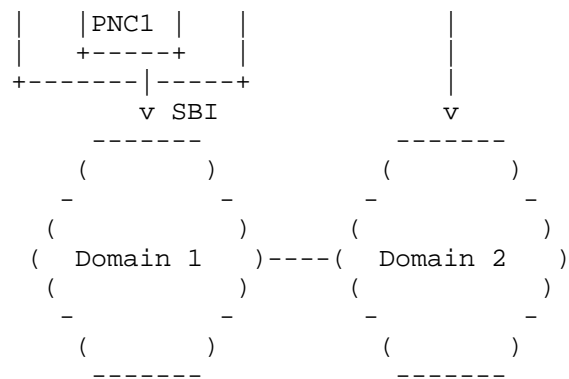
Young Lee
 Huawei Technologies
 5340 Legacy Drive
 Plano, TX 75023, USA
 Phone: (469)277-5838
 Email: leeyoung@huawei.com

APPENDIX A - Example of MDSC and PNC Functions Integrated in A Service/Network Orchestrator

This section provides an example of a possible deployment scenario, in which Service/Network Orchestrator can include a number of functionalities, among which, in the example below, PNC functionalities for domain 2 and MDSC functionalities to coordinate the PNC1 functionalities (hosted in a separate domain controller) and PNC2 functionalities (co-hosted in the network orchestrator).

Customer





TEAS Working Group
Internet Draft
Intended status: Informational

Expires September 1, 2018.

Young Lee (Editor)
Huawei

Daniele Ceccarelli
Ericsson

Takuya Miyasaka
KDDI

Jong Yoon Shin
SKT

Kwang-koog Lee
KT

March 1, 2018

Requirements for Abstraction and Control of TE Networks

draft-ietf-teas-actn-requirements-09

Abstract

This document provides a set of functional requirements for abstraction and control of Traffic Engineering networks to facilitate virtual network operation via the creation of a single virtualized network or a seamless service. This supports operators in viewing and controlling different domains (at any dimension: applied technology, administrative zones, or vendor-specific technology islands) as a single virtualized network.

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 September 1, 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 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. Requirements Language.....	4
2. High-level ACTN requirements.....	4
2.1. Service-Specific Requirements.....	5
2.2. Network-Related Requirements.....	7
3. Security Considerations.....	9
4. IANA Considerations.....	9
5. References.....	10
5.1. Normative References.....	10
5.2. Informative References.....	10
6. Contributors.....	11
Authors' Addresses.....	12

1. Introduction

This document provides a set of functional requirements for Abstraction and Control of Traffic Engineering (TE) Networks (ACTN) identified in various use-cases specified by the operators. [ACTN-Frame] defines the base reference architecture and terminology.

ACTN refers to the set of virtual network service operations needed to coordinate, control and manage large-scale multi-domain TE networks so as to facilitate network programmability, automation, efficient resource sharing, and end-to-end virtual service aware connectivity.

These operations are summarized as follows:

- Abstraction and coordination of underlying network resources independent of how these resources are managed or controlled, so that higher-layer entities can dynamically control virtual networks based on those resources. Control includes creating, modifying, monitoring, and deleting virtual networks.
- Collation of the identifiers and other attributes of the resources from multiple TE networks (multiple technologies, equipment from multiple vendors, under the control of multiple administrations) through a process of recursive abstraction to present a customer with a single virtual network. This is achieved by presenting the network domain as an abstracted topology to the customer via open and programmable interfaces. Recursive abstraction allows for the recursion of abstracted data in a hierarchy of controllers.. It is expected that the recursion levels should be at least three levels: customer level, multi-domain network level, and domain network level.
- Coordination of end-to-end virtual network services and applications via allocation of network resources to meet specific service, application and customer requirements. Refer to [ACTN-Frame] for the definition of coordination.
- Adaptation of customer requests (to control virtual resources) to the physical network resources performing the necessary mapping, translation, isolation and, policy that allows conveying, managing and enforcing customer policies with respect to the services and the network of the customer.

- Provision via a data model and virtual control capability to customers who request virtual network services. Note that these customers could, themselves, be service providers.

ACTN solutions will build on, and extend, existing TE constructs and TE mechanisms wherever possible and appropriate. Support for controller-based approaches is specifically included in the possible solution set.

1.1. 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.

2. High-level ACTN requirements

This section provides a summary of use-cases in terms of two categories: (i) service-specific requirements; (ii) network-related requirements. All these requirements are specified by operators that are interested in implementing ACTN.

Service-specific requirements listed below are uniquely applied to the work scope of ACTN. Service-specific requirements are related to the virtual service coordination function. These requirements are related to customer's Virtual Networks (VN) in terms of service policy associated with VNs such as service performance objectives, VN endpoint location information for certain required service specific functions (e.g., security and others), VN survivability requirement, or dynamic service control policy, etc.

Network-related requirements are related to and necessary for coherent/seamless for the virtual network operation function. These requirements are related to multi-domain and multi-layer signaling, routing, protection/restoration and re-optimization/re-grooming, etc.

Each requirement specified in Sections 2.1 and 2.2 is derived from ACTN use-cases: [CHENG], [DHODY], [FANG], [KLEE], [KUMAKI], [LOPEZ], [SHIN], [XU], [XU2], and [SUZUKI].

2.1. Service-Specific Requirements

1. Requirement 1: Virtual Network Service (VNS) creation

Customer MUST be able to request/instantiate the VNS to the network within the confines of mutual agreement between customer and network operator and network operator's capability. A VNS is the service agreement between a customer and provider to provide a VN [ACTN-Frame]. There are different types of VNS in terms of the VN types the customer is allowed to operate (e.g., a VN type can be simply a set of edge-to-edge links, or it can comprise of virtual nodes and virtual links, etc.). The customer MUST be able to express VNS preference that captures Service Level Agreements (SLA) associated with virtual network service (e.g., Endpoint selection preference, routing preference, time-related preference, etc.)

Reference: [KLEE], [LOPEZ], [SHIN], [DHODY], [FANG].

2. Requirement 2: Virtual Network Service Query

Customer SHOULD be able to request VNS Query ("Can you give me these VN(s)?") that include the following parameters:

- VN type: various VN types defined by the customer (e.g., path, graph, etc.)
- VN end-points (Customer Edge interface information)
- VN Topology Service-specific Objective Functions (e.g., a set of objective functions as defined in [RFC5541] to be supported on the paths, but not limited to).
- VN constraints requirement (e.g., Maximum Latency threshold, Minimum Bandwidth, etc.)

Reference: [KUMAKI], [FANG], [CHENG].

3. Requirement 3: VNS Instantiation ("Please create a VNS for me")

Customer MUST be able to instantiate VNS that includes various VNS related parameters:

- VN type: various VN types defined by the customer (e.g., Type 1, Type 2, etc. See [ACTN-Frame] for the definition of VN Type 1 and Type 2).
- VN end-points (Customer Edge interface information)
- VN Topology Service-specific Objective Functions (e.g., a set of objective functions as defined in [RFC5541] to be supported on the paths, but not limited to).
- VN constraints requirement (e.g., Maximum Latency threshold, Minimum Bandwidth, etc.)
- VN Topology diversity when there are multiple instances of VNS (e.g., VN1 and VN2 must be disjoint; Node/link disjoint from other VNs)

Note that Requirement 3 provides specific details of Requirement 1.

Reference: [KUMAKI], [FANG], [CHENG].

4. Requirement 4: VNS Lifecycle Management & Operation (M&O)

Customer MUST be able to perform the following VNS operations:

- VNS Delete: Customer MUST be able to delete VNS.
- VNS Modify: Customer MUST be able to modify VNS related parameters during the lifecycle of the instantiated VNS.

Reference: [FANG], [KUMAKI], [LOPEZ], [DHODY], [FANG], [KLEE].

5. Requirement 5: VNS Isolation

Customer's VN should be able to use arbitrary network topology, routing, or forwarding functions as well as customized control mechanisms independent of the underlying physical network and of other coexisting virtual networks. Other customers' VNS operation MUST NOT impact a particular customer's VNS network operation.

Reference: [KUMAKI], [FANG], [LOPEZ]

6. Requirement 6: Multi-Destination Coordination

Customer MUST be able to define and convey service/preference requirements for multi-destination applications (e.g., set of candidate sources/destinations, thresholds for load balancing, disaster recovery preference, etc.)

Reference: [FANG], [LOPEZ], [SHIN].

7. Requirement 7: VNS Performance Monitoring

The customer MUST be able to define performance monitoring parameters and its associated preference such as frequency of report, abstraction/aggregation level of performance data (e.g., VN level, tunnel level, etc.) with dynamic feedback loop from the network.

Reference: [XU], [XU2], [DHODY], [CHENG]

8. Requirement 8: VNS Confidentiality and Security Requirements

The following confidentiality/security requirements MUST be supported in all interfaces:

- Securing the request and control of resources, confidentiality of the information, and availability of function.
- Trust domain verification between a customer entity and a network entity. It verifies if a trust relationship has been established between these entities.
- Encrypting data that flow between components, especially when they are implemented at remote nodes, regardless if these are external or internal network interfaces.

Reference: [KUMAKI], [FANG], [LOPEZ]

2.2. Network-Related Requirements

1. Requirement 1: Virtual Network Service Coordination

Network MUST be able to support the following VNS operations:

- VNS Create: Upon customer's VNS creation request, network MUST be able to create VNS within the confines of network resource availability.
- VNS Delete: Upon customer's VNS deletion request, network MUST be able to delete VNS.
- VNS Modify: Upon customer's VNS modification request, network MUST be able to modify VNS related parameters during the lifecycle of the instantiated VNS.
- VNS Monitor: Upon customer's VNS performance monitoring setup, the network MUST be able to support VNS level Operations, Administration and Management (OAM) Monitoring under service agreement.

Reference: [FANG], [KUMAKI], [LOPEZ], [DHODY], [FANG], [KLEE].

2. Requirement 2: Topology Abstraction Capability

The network MUST be capable of managing its networks based on the principle of topology abstraction to be able to scale multi-layer, multi-domain networks.

Reference: [KLEE], [LOPEZ], [DHODY], [CHENG].

3. Requirement 3: Multi-Domain & Multi-layer Coordination

Network coordination for multi-domain and multi-layer path computation and path setup operation MUST be provided:

- End-to-end path computation across multi-domain networks (based on abstract topology from each domain)
- Domain sequence determination
- Request for path signaling to each domain controller
- Alternative TE path computation if any of the domain controllers cannot find its domain path

Reference: [CHENG], [DHODY], [KLEE], [LOPEZ], [SHIN], [SUZUKI].

4. Requirement 4: End-to-End Path Protection

End-to-end Path Protection Operations MUST be provided with seamless coordination between domain-level protection schemes and cross-domain protection schemes.

Reference: [CHENG], [KLEE], [DHODY], [LOPEZ], [SHIN].

5. Requirement 5: Dynamicity of virtual network control operations

Dynamic virtual network control operations MUST be supported. This includes, but is not limited to, the following:

- Real-time VNS control (e.g., fast recovery/reroute upon network failure).
- Fast convergence of abstracted topologies upon changes due to failure or reconfiguration across the network domain view, the multi-domain network view and the customer view.
- Large-scale VNS operation (e.g., the ability to process tens of thousands of connectivity requests) for time-sensitive applications.

Reference: [SHIN], [XU], [XU2], [KLEE], [KUMAKI], [SUZUKI].

3. Security Considerations

The ACTN requirements described in this document do not directly bear specific security concerns. When these requirements are implemented in specific interfaces, securing the request and control of resources, confidentiality of the information, and availability of function, should all be critical security considerations.

4. IANA Considerations

This document has no actions for IANA.

5. References

5.1. Normative References

- [ACTN-Frame] D. Ceccarelli, et al., "Framework for Abstraction and Control of Transport Networks", draft-ietf-teas-actn-framework, work in progress.

5.2. Informative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.
- [RFC5541] JL. Le Roux, JP. Vasseur, and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.
- [RFC8174] B. Leiba, "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.
- [CHENG] W. Cheng, et. al., "ACTN Use-cases for Packet Transport Networks in Mobile Backhaul Networks", draft-cheng-actn-ptn-requirements-00, July 21, 2014.
- [DHODY] D. Dhody, et. al., "Packet Optical Integration (POI) Use Cases for Abstraction and Control of Transport Networks (ACTN)", draft-dhody-actn-poi-use-case-07, October 28, 2016.
- [FANG] L. Fang, "ACTN Use Case for Multi-domain Data Center Interconnect", draft-fang-actn-multidomain-dci-01, September 29, 2014.
- [KLEE] K. Lee, H. Lee, R. Vilata, V. Lopez, "ACTN Use-case for E2E Network Services in Multiple Vendor Domain Transport Networks", draft-klee-teas-actn-connectivity-multi-domain-03, July 31, 2017.
- [KUMAKI] K. Kumaki, T. Miyasaka, "ACTN : Use case for Multi Tenant VNO", draft-kumaki-teas-actn-multitenant-vno-00, May 29, 2014.
- [LOPEZ] D. Lopez (Ed), "ACTN Use-case for Virtual Network Operation for Multiple Domains in a Single Operator Network", draft-lopez-actn-vno-multidomains-01, October 27, 2014.

- [SHIN] J. Shin, R. Hwang, J. Lee, "ACTN Use-case for Mobile Virtual Network Operation for Multiple Domains in a Single Operator Network", draft-shin-actn-mvno-multi-domain-00, June 30, 2014.
- [XU] Y. Xu, et. al., "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control-03, April 23, 2015.
- [XU2] Y. Xu, et. al., "Requirements of Abstract Alarm Report in ACTN architecture", draft-xu-teas-actn-abstract-alarm-report-00, July 6, 2015.
- [SUZUKI] T. Suzuki, et. al., "Use-case and Requirements for Multi-domain Operation Plane Change", draft-suzuki-teas-actn-multidomain-opc-00, July 6, 2015.

6. Contributors

Dhruv Dhody
Huawei Technologies
Email: dhruv.ietf@gmail.com

Sergio Belotti
Nokia
Via Trento, 30
Vimercate, Italy
Email: sergio.belotti@nokia.com

Khuzema Pithewan
Peloton Technology
Email: khuzemap@gmail.com

Yunbin Xu
CATR
Email: xuyunbin@ritt.cn

Toshiaki Suzuki
Hitachi

Email: toshiaki.suzuki.cs@hitachi.com

Haomian Zheng
Huawei
Email: zhenghaomian@huawei.com

Authors' Addresses

Young Lee (Editor)
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838
Email: leeyoung@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan, 48
Stockholm, Sweden
Email: daniele.ceccarelli@ericsson.com

Takuya Miyasaka
KDDI
Email: ta-miyasaka@kddi.com

Jong Yoon Shin
SKT
Email: jongyoon.shin@sk.com

Kwang-koog Lee
KT
Email: kwangkoog.lee@kt.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.

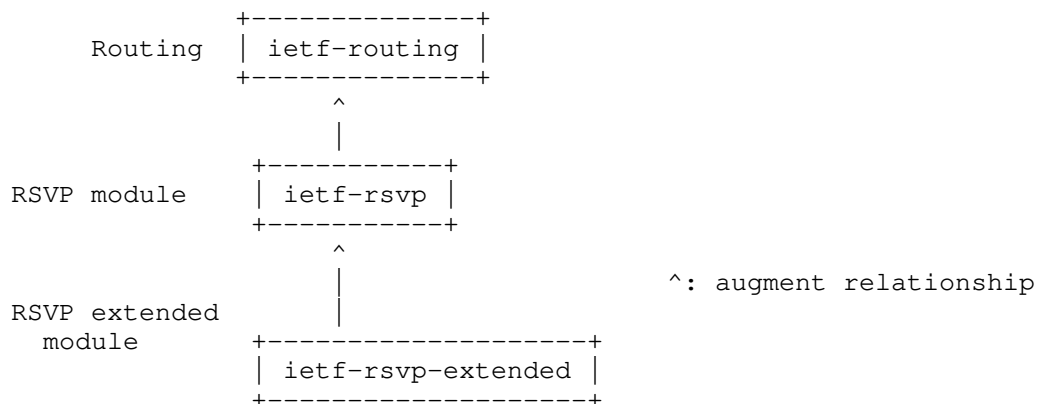


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

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)

```

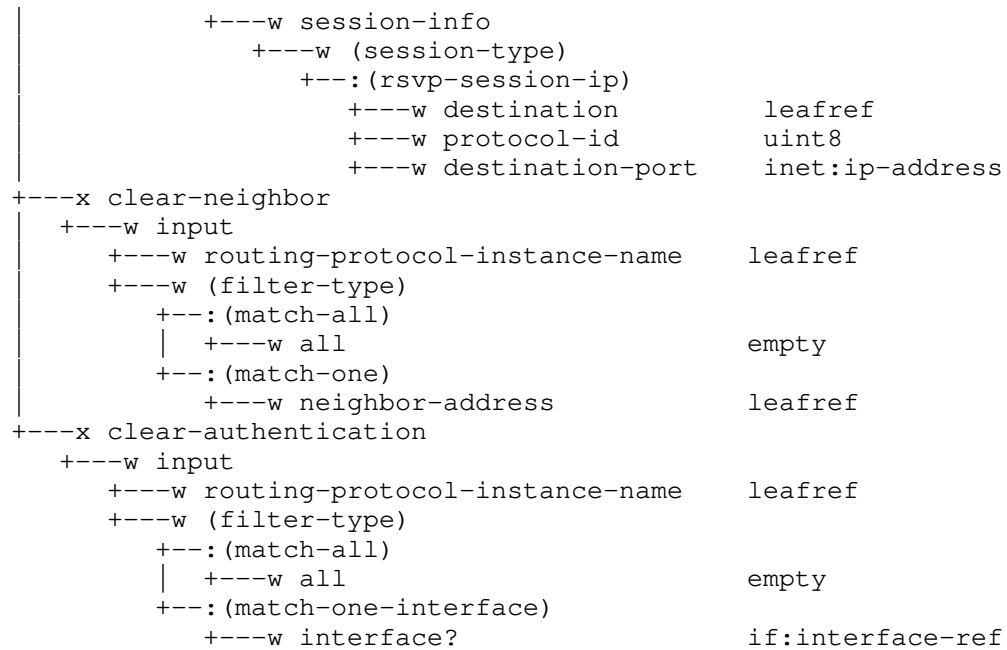


Figure 3: RSVP model tree diagram

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

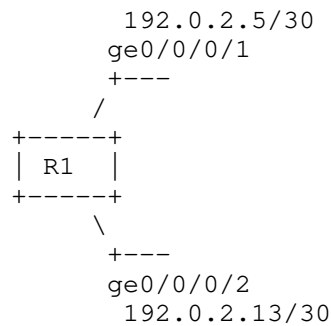


Figure 6: Example of network configuration.

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: August 26, 2021

V. Beeram
T. Saad
Juniper Networks
R. Gandhi
Cisco Systems, Inc.
X. Liu
Volta Networks
I. Bryskin
Huawei Technologies
H. Shah
Ciena
February 22, 2021

A YANG Data Model for RSVP-TE Protocol
draft-ietf-teas-yang-rsvp-te-09

Abstract

This document defines a YANG data model for the configuration and management of RSVP (Resource Reservation Protocol) to establish Traffic-Engineered (TE) Label-Switched Paths (LSPs) for MPLS (Multi-Protocol Label Switching) and other technologies.

The model defines a generic RSVP-TE module for signaling LSPs that are technology agnostic. The generic RSVP-TE module is to be augmented by technology specific RSVP-TE modules that define technology specific data. This document also defines the augmentation for RSVP-TE MPLS LSPs model.

This model covers data for the 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 August 26, 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	2
1.1. Terminology	3
1.2. Prefixes in Data Node Names	3
2. Model Overview	4
2.1. Module Relationship	4
2.2. Model Tree Diagrams	5
2.2.1. RSVP-TE Model Tree Diagram	5
2.2.2. RSVP-TE MPLS Model Tree Diagram	9
2.3. YANG Modules	11
2.3.1. RSVP-TE YANG Module	11
2.3.2. RSVP-TE MPLS YANG Module	24
3. IANA Considerations	36
4. Security Considerations	37
5. Acknowledgement	38
6. Contributors	38
7. References	38
7.1. Normative References	38
7.2. Informative References	40
Authors' Addresses	41

1. Introduction

YANG [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 generic YANG data model for configuring and managing RSVP-TE LSP(s) [RFC3209]. The RSVP-TE generic model augments the RSVP base and extended models defined in [I-D.ietf-teas-yang-rsvp], and adds TE extensions to the RSVP protocol [RFC2205] model configuration and state data. The technology specific RSVP-TE models augment the generic RSVP-TE model with additional technology specific parameters. For example, this document also defines the MPLS RSVP-TE model for configuring and managing MPLS RSVP TE LSP(s).

In addition to augmenting the RSVP YANG module, the modules defined in this document augment the TE Interfaces, Tunnels and LSP(s) YANG module defined in [I-D.ietf-teas-yang-te] to define additional parameters to enable signaling for RSVP-TE.

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 terminology for describing YANG data models is found in [RFC7950].

1.2. 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]
te	ietf-te	[I-D.ietf-teas-yang-te]
rsvp	ietf-rsvp	[I-D.ietf-teas-yang-rsvp]
te-dev	ietf-te-device	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te-types]
te-mpls-types	ietf-te-mpls-types	[I-D.ietf-teas-yang-te-types]
rsvp-te	ietf-rsvp-te	this document
rsvp-te-mpls	ietf-rsvp-te-mpls	this document

Table 1: Prefixes and corresponding YANG modules

2. Model Overview

The RSVP-TE generic model augments the RSVP base and extended YANG models defined in [I-D.ietf-teas-yang-rsvp]. It also augments the TE tunnels and interfaces module defined in [I-D.ietf-teas-yang-te] to cover parameters specific to the configuration and management of RSVP-TE interfaces, tunnels and LSP(s).

The RSVP-TE MPLS YANG model augments the RSVP-TE generic model with parameters to configure and manage signaling of MPLS RSVP-TE LSPs. RSVP-TE model augmentation for other dataplane technologies (e.g. OTN or WDM) are outside the scope of this document.

There are three types of configuration and state data nodes in module(s) defined in this document:

- o those augmenting or extending the base RSVP module that is defined in [I-D.ietf-teas-yang-rsvp]
- o those augmenting or extending the base TE module defined in [I-D.ietf-teas-yang-te]
- o those that are specific to the RSVP-TE and RSVP-TE MPLS modules defined in this document.

2.1. Module Relationship

The data pertaining to RSVP-TE in this document is divided into two modules: a technology agnostic RSVP-TE module that holds generic parameters for RSVP-TE applicable to all technologies, and a MPLS technology specific RSVP-TE module that holds parameters specific to MPLS technology.

The relationship between the different modules is shown in Figure 1.

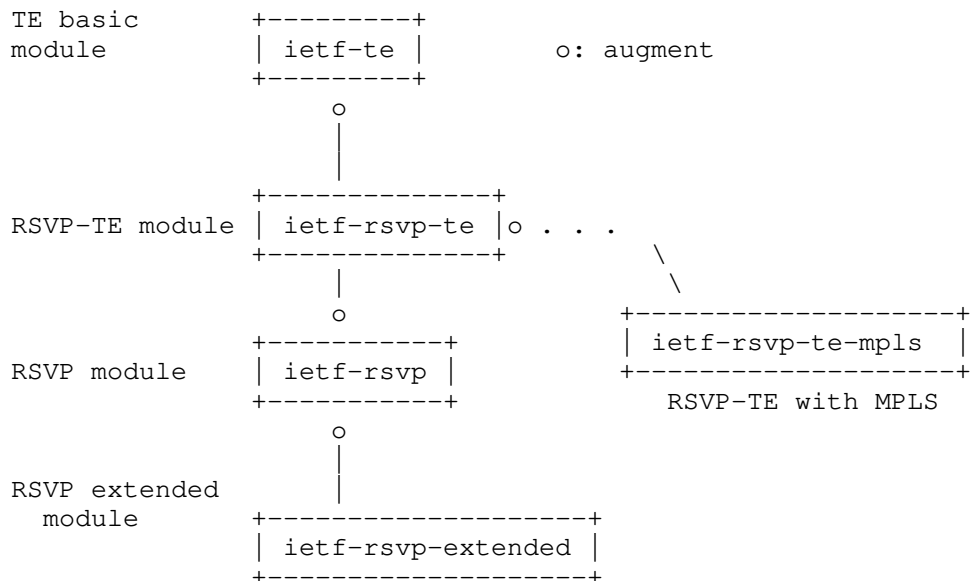


Figure 1: Relationship of RSVP and RSVP-TE modules with other protocol modules

2.2. Model Tree Diagrams

A full tree diagram of the module(s) defined in this document as per the syntax defined in [RFC8340] are given in subsequent sections.

2.2.1. RSVP-TE Model Tree Diagram

Figure 2 shows the YANG tree diagram of the RSVP-TE generic YANG model defined in module `ietf-rsvp-te.yang`.

```

module: ietf-rsvp-te
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp:
      +---rw global-soft-preemption!
      +---rw soft-preemption-timeout?  uint16
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
      +---rw rsvp-te-interface-attributes
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
      /rsvp:interface:
        +---rw rsvp-te-interface-attributes
  
```

```

augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rsvp:rsvp/rsvp:sessions:
    +---ro session-te* [tunnel-endpoint tunnel-id extended-tunnel-id]
      +---ro tunnel-endpoint      inet:ip-address
      +---ro tunnel-id            uint16
      +---ro extended-tunnel-id   inet:ip-address
      +---ro destination-port?    inet:port-number
      +---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 tspec-average-rate?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro tspec-size?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro tspec-peak-rate?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro min-policed-unit?  uint32
        |   +---ro max-packet-size?   uint32
      +---ro rsbs
        | +---ro rsb* []
        |   +---ro source-port?      inet:port-number
        |   +---ro reservation-style  identityref
        |   +---ro expires-in?       uint32
        |   +---ro fspec-average-rate?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro fspec-size?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro fspec-peak-rate?
        |   |   rt-types:bandwidth-ieee-float32
        |   +---ro min-policed-unit?  uint32
        |   +---ro max-packet-size?   uint32
augment /rt:routing/rt:control-plane-protocols
  /rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors:
augment /te:te/te:tunnels/te:tunnel:
  +---rw lsp-sigaled-name?  string
  +---rw session-attribute*  identityref
  +---rw lsp-attribute*      identityref
  +---rw retry-timer?        uint16
augment /te:te/te:lsps/te:lsp:
  +---ro associated-rsvp-session?  leafref
  +---ro lsp-sigaled-name?         string

```

```

+--ro session-attribute*                identityref
+--ro lsp-attribute*                    identityref
+--ro rsvp-message-type?                identityref
+--ro rsvp-error-code?                  uint8
+--ro rsvp-error-subcode?               uint16
+--ro explicit-route-objects
|
|   +--ro incoming-explicit-route-hop* [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 outgoing-explicit-route-hop* [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 incoming-record-route-subobjects
+---ro incoming-record-route-subobject* [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)
|   +---ro label-hop
|       +---ro te-label
|           +---ro (technology)?
|               +---:(generic)
|                   +---ro generic?
|                       rt-types:generalized-label
|       +---ro direction?     te-label-direction
|       +---ro flags*      path-attribute-flags
+---ro outgoing-record-route-subobjects
+---ro outgoing-record-route-subobject* [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)
|   +---ro label-hop
|       +---ro te-label
|           +---ro (technology)?
|               +---:(generic)
|                   +---ro generic?
|                       rt-types:generalized-label
|       +---ro direction?   te-label-direction
|       +---ro flags*       path-attribute-flags
augment /te:te/te-dev:interfaces/te-dev:interface:

```

Figure 2: RSVP-TE model Tree diagram

2.2.2. RSVP-TE MPLS Model Tree Diagram

Figure 5 shows the YANG tree diagram of the RSVP-TE MPLS YANG model defined in module `ietf-rsvp-te-mpls.yang` and that augments RSVP-TE module as well as RSVP and TE YANG modules.

```

module: ietf-rsvp-te-mpls
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp:
    +---rw rsvp-frr-local-revert-delay?  uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
    /rsvp:interface:
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:sessions:
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors:
augment /te:te/te:tunnels/te:tunnel:
    +---rw session-attribute*  identityref
augment /te:te/te:lsps/te:lsp:

```



```

    +---ro session-attribute*   identityref
    +---ro backup-info
      +---ro backup-tunnel-name?   string
      +---ro backup-frr-on?        uint8
      +---ro backup-protected-lsp-num?  uint32
    augment /te:te/te:tunnels/te:tunnel/te:primary-paths
      /te:primary-path/te:lsps/te:lsp:
      +---ro session-attribute*   identityref
      +---ro backup-info
        +---ro backup-tunnel-name?   string
        +---ro backup-frr-on?        uint8
        +---ro backup-protected-lsp-num?  uint32
    augment /te:te/te:tunnels/te:tunnel/te:secondary-paths
      /te:secondary-path/te:lsps/te:lsp:
      +---ro session-attribute*   identityref
      +---ro backup-info
        +---ro backup-tunnel-name?   string
        +---ro backup-frr-on?        uint8
        +---ro backup-protected-lsp-num?  uint32
    augment /te:te/te-dev:interfaces/te-dev:interface:
      +---rw bandwidth-mpls-reservable
      +---rw (bandwidth-value)?
      |   +---:(absolute)
      |   |   +---rw absolute-value?   te-packet-types:bandwidth-kbps
      |   +---:(percentage)
      |   |   +---rw percent-value?    uint32
      +---rw (bc-model-type)?
      |   +---:(bc-model-rdm)
      |   |   +---rw bc-model-rdm
      |   |   |   +---rw bandwidth-mpls-constraints
      |   |   |   |   +---rw maximum-reservable?
      |   |   |   |   |   te-packet-types:bandwidth-kbps
      |   |   |   +---rw bc-value*      uint32
      |   +---:(bc-model-mam)
      |   |   +---rw bc-model-mam
      |   |   |   +---rw bandwidth-mpls-constraints
      |   |   |   |   +---rw maximum-reservable?
      |   |   |   |   |   te-packet-types:bandwidth-kbps
      |   |   |   +---rw bc-value*      uint32
      |   +---:(bc-model-mar)
      |   |   +---rw bc-model-mar
      |   |   |   +---rw bandwidth-mpls-constraints
      |   |   |   |   +---rw maximum-reservable?
      |   |   |   |   |   te-packet-types:bandwidth-kbps
      |   |   |   +---rw bc-value*      uint32
    augment /te:te/te-dev:interfaces/te-dev:interface:
      +---rw rsvp-te-frr-backups
      +---rw (type)?

```

```

+---:(static-tunnel)
|   +---rw static-backups
|       +---rw static-backup* [backup-tunnel-name]
|       +---rw backup-tunnel-name
|           -> /te:te/tunnels/tunnel/name
+---:(auto-tunnel)
    +---rw auto-tunnel-backups
    +---rw auto-backup-protection?          identityref
    +---rw auto-backup-path-computation?    identityref

```

Figure 3: RSVP-TE MPLS Tree diagram

2.3. YANG Modules

2.3.1. RSVP-TE YANG Module

The RSVP-TE generic YANG module "ietf-rsvp-te" imports the following modules:

- o ietf-rsvp defined in [I-D.ietf-teas-yang-rsvp]
- o ietf-routing-types defined in [RFC8294]
- o ietf-te-types defined in [I-D.ietf-teas-yang-te-types]
- o ietf-te and ietf-te-dev defined in [I-D.ietf-teas-yang-te]

This module references the following documents:

[I-D.ietf-teas-yang-rsvp], [RFC8349], [I-D.ietf-teas-yang-te], [I-D.ietf-teas-yang-te-types], [RFC2210], [RFC4920], [RFC5420], [RFC7570], [RFC4859].

```

<CODE BEGINS> file "ietf-rsvp-te@2021-02-21.yang"
module ietf-rsvp-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te";
  prefix rsvp-te;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "draft-ietf-teas-yang-rsvp: A YANG Data Model for
      Resource Reservation Protocol (RSVP)";
  }
  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-te {
  prefix te;
  reference
    "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
    Engineering Tunnels and Interfaces";
}
import ietf-te-device {
  prefix te-dev;
  reference
    "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
    Engineering Tunnels and Interfaces";
}

/* 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";
}

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.net@gmail.com>

  Editor:    Rakesh Gandhi
             <mailto:rgandhi@cisco.com>
```

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Igor Bryskin
<mailto:Igor.Bryskin@huawei.com>

Editor: Himanshu Shah
<mailto:hshah@ciena.com>;

description

"This module contains the RSVP-TE YANG generic data model.
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 2021-02-21 {  
  description  
    "A YANG Data Model for RSVP-TE";  
  reference  
    "RFCXXXX: A YANG Data Model for RSVP-TE Protocol";  
}
```

```
identity rsvp-message-type {  
  description  
    "RSVP message types";  
}
```

```
identity rsvp-message-path {  
  base rsvp-message-type;  
  description  
    "RSVP Path message";  
  reference  
    "RFC2205";  
}
```

```
}

identity rsvp-message-resv {
  base rsvp-message-type;
  description
    "RSVP Resv message";
  reference
    "RFC2205";
}

identity rsvp-message-path-err {
  base rsvp-message-type;
  description
    "RSVP Path-Err message";
  reference
    "RFC2205";
}

identity rsvp-message-resv-err {
  base rsvp-message-type;
  description
    "RSVP Resv-Err message";
  reference
    "RFC2205";
}

identity rsvp-message-path-tear {
  base rsvp-message-type;
  description
    "RSVP Path Tear message";
  reference
    "RFC2205";
}

identity rsvp-message-resv-conf {
  base rsvp-message-type;
  description
    "RSVP Resv Confirm message";
  reference
    "RFC2205";
}

identity rsvp-message-srefresh {
  base rsvp-message-type;
  description
    "RSVP SRefresh message";
  reference
    "RFC2961";
}
```

```
}

identity rsvp-message-hello {
  base rsvp-message-type;
  description
    "RSVP Hello message";
  reference
    "RFC3209";
}

identity rsvp-message-bundle {
  base rsvp-message-type;
  description
    "RSVP Bundle message";
  reference
    "RFC2961";
}

identity rsvp-message-notify {
  base rsvp-message-type;
  description
    "RSVP Notify message";
  reference
    "RFC3473";
}

/**
 * RSVP-TE LSPs groupings.
 */

grouping lsp-record-route-information-state {
  description
    "recorded route information grouping";
  container incoming-record-route-subobjects {
    description
      "RSVP recorded route object incoming information";
    list incoming-record-route-subobject {
      when "../te:origin-type != 'ingress'" {
        description
          "Applicable on non-ingress LSPs only";
      }
      key "index";
      ordered-by user;
      description
        "List of RSVP Path record-route objects";
      uses te-types:record-route-state;
    }
  }
}
```

```
    container outgoing-record-route-subobjects {
      description
        "RSVP recorded route object outgoing information";
      list outgoing-record-route-subobject {
        when "../te:origin-type != 'egress'" {
          description
            "Applicable on non-egress LSPs only";
        }
        key "index";
        ordered-by user;
        description
          "List of RSVP Resv record-route objects";
        uses te-types:record-route-state;
      }
    }
  }

  grouping lsp-explicit-route-information-state {
    description
      "RSVP-TE LSP explicit-route information";
    container explicit-route-objects {
      description
        "Explicit route object information";
      list incoming-explicit-route-hop {
        when "../te:origin-type != 'ingress'" {
          description
            "Applicable on non-ingress LSPs only";
        }
        key "index";
        ordered-by user;
        description
          "List of incoming RSVP Path explicit-route objects";
        leaf index {
          type uint32;
          description
            "Explicit route hop index. The index is used to
            identify an entry in the list. The order of entries
            is defined by the user without relying on key values";
        }
        uses te-types:explicit-route-hop;
      }
      list outgoing-explicit-route-hop {
        when "../te:origin-type != 'egress'" {
          description
            "Applicable on non-egress LSPs only";
        }
        key "index";
        ordered-by user;
      }
    }
  }
```

```
    description
      "List of outgoing RSVP Path explicit-route objects";
    leaf index {
      type uint32;
      description
        "Explicit route hop index. The index is used to
         identify an entry in the list. The order of entries
         is defined by the user without relying on key values";
    }
    uses te-types:explicit-route-hop;
  }
}

grouping lsp-attributes-flags {
  description
    "Configuration parameters relating to RSVP-TE LSP
     attribute flags";
  leaf-list lsp-attribute {
    type identityref {
      base te-types:lsp-attributes-flags;
    }
    description
      "RSVP per LSP attributes flags";
    reference
      "RFC4920, RFC5420, RFC7570";
  }
}

grouping lsp-session-attributes-obj-flags {
  description
    "Configuration parameters relating to RSVP-TE LSP
     session attribute flags";
  reference
    "RFC4859: Registry for RSVP-TE Session Flags";
  leaf-list session-attribute {
    when "../session-attribute !=
      'te-types:bandwidth-protection-desired' or
      ../session-attribute !=
      'te-types:soft-preemption-desired'" {
      description
        "Session attributes applicable to generic technologies
         only.";
    }
    type identityref {
      base te-types:session-attributes-flags;
    }
    description

```



```
        "RSVP session attributes flags";
      reference
        "RFC4859: Registry for RSVP-TE Session Flags";
    }
  }

  grouping lsp-properties {
    description
      "Configuration parameters relating to RSVP-TE LSP
       session attribute flags";
    leaf lsp-signaled-name {
      type string;
      description
        "Sets the session name to use in the session
         attribute object.";
    }
    uses lsp-session-attributes-obj-flags;
    uses lsp-attributes-flags;
  }

  grouping tunnel-properties {
    description
      "RSVP-TE Tunnel properties grouping";
    leaf retry-timer {
      type uint16 {
        range "1..600";
      }
      units "seconds";
      description
        "sets the time between attempts to establish the
         LSP";
    }
  }
}

/**** End of RSVP-TE LSP groupings ****/
/**
 * RSVP-TE generic global properties.
 */

grouping global-soft-preemption {
  description
    "Configuration for global RSVP-TE soft preemption";
  container global-soft-preemption {
    presence "Enables soft preemption on a node.";
    description
      "Top level container for RSVP-TE soft-preemption";
    leaf soft-preemption-timeout {
      type uint16 {
```

```
        range "0..300";
    }
    units "seconds";
    default "0";
    description
        "Timeout value for soft preemption to revert
         to hard preemption";
    }
}

/** End of RSVP-TE generic global properties. */
/**
 * RSVP-TE interface generic groupings.
 */

grouping rsvp-te-interface-attributes {
    description
        "Top level grouping for RSVP-TE interface properties.";
    container rsvp-te-interface-attributes {
        description
            "Top level container for RSVP-TE interface
             properties";
    }
}

/** End of RSVP-TE generic groupings */
/* RSVP-TE global properties */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp" {
    description
        "RSVP-TE augmentation to RSVP globals";
    uses global-soft-preemption;
}

/* Linkage to the base RSVP all links */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
    description
        "RSVP-TE generic data augmentation pertaining to interfaces";
    uses rsvp-te-interface-attributes;
}

/* Linkage to per RSVP interface */

augment "/rt:routing/rt:control-plane-protocols/"
```

```
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:interface" {
description
  "RSVP-TE generic data augmentation pertaining to specific
  interface";
uses rsvp-te-interface-attributes;
}

/* add augmentation for sessions and neighbors */

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/"
  + "rsvp:sessions" {
description
  "RSVP-TE generic data augmentation pertaining to session";
list session-te {
  key "tunnel-endpoint tunnel-id extended-tunnel-id";
  config false;
  description
    "List of RSVP sessions";
  leaf tunnel-endpoint {
    type inet:ip-address;
    description
      "XX";
  }
  leaf tunnel-id {
    type uint16;
    description
      "XX";
  }
  leaf extended-tunnel-id {
    type inet:ip-address;
    description
      "XX";
  }
  uses rsvp:session-attributes;
}
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/"
  + "rsvp:sessions/session-te/psbs/psb" {
description
  "RSVP-TE generic data augmentation pertaining to session";
/* To be added */
leaf tspec-average-rate {
  type rt-types:bandwidth-ieee-float32;
  units "Bytes per second";
}
```

```
    description
      "Tspec Token Bucket Average Rate";
    reference
      "RFC2210: RSVP with INTSERV";
  }
  leaf tspec-size {
    type rt-types:bandwidth-ieee-float32;
    units "Bytes per second";
    description
      "Tspec Token Bucket Burst Rate";
    reference
      "RFC2210";
  }
  leaf tspec-peak-rate {
    type rt-types:bandwidth-ieee-float32;
    units "Bytes per second";
    description
      "Tspec Token Bucket Peak Data Rate";
    reference
      "RFC2210";
  }
  leaf min-policed-unit {
    type uint32;
    description
      "Tspec Minimum Policed Unit";
    reference
      "RFC2210";
  }
  leaf max-packet-size {
    type uint32;
    description
      "Tspec Maximum Packet Size";
    reference
      "RFC2210";
  }
}

augment "/rt:routing/rt:control-plane-protocols/"
  + "rt:control-plane-protocol/rsvp:rsvp/"
  + "rsvp:sessions/session-te/rsbs/rsb" {
  description
    "RSVP-TE generic data augmentation pertaining to session";
  leaf fspec-average-rate {
    type rt-types:bandwidth-ieee-float32;
    units "Bytes per second";
    description
      "Fspec Token Bucket Average Rate";
    reference
```

```
        "RFC2210";
    }
    leaf fspec-size {
        type rt-types:bandwidth-ieee-float32;
        units "Bytes per second";
        description
            "Fspec Token Bucket Burst Rate";
        reference
            "RFC2210";
    }
    leaf fspec-peak-rate {
        type rt-types:bandwidth-ieee-float32;
        units "Bytes per second";
        description
            "Fspec Token Bucket Peak Data Rate";
        reference
            "RFC2210";
    }
    leaf min-policed-unit {
        type uint32;
        description
            "Fspec Minimum Policed Unit";
        reference
            "RFC2210";
    }
    leaf max-packet-size {
        type uint32;
        description
            "Fspec Maximum Packet Size";
        reference
            "RFC2210";
    }
}

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors" {
    description
        "RSVP-TE generic data augmentation pertaining to neighbors";
    /* To be added */
}

/**
 * RSVP-TE generic augmentations of generic TE model.
 */
/* TE tunnel augmentation */

augment "/te:te/te:tunnels/te:tunnel" {
    when "/te:te/te:tunnels/te:tunnel"
```

```
    + "/te:primary-paths/te:primary-path"
    + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
  description
    "When the path signaling protocol is RSVP-TE ";
}
description
  "RSVP-TE generic data augmentation pertaining to TE tunnels";
uses lsp-properties;
uses tunnel-properties;
}

/* TE LSP augmentation */

grouping rsvp-te-lsp-error-info {
  description
    "Grouping for RSVP-TE error reporting information";
  leaf rsvp-message-type {
    type identityref {
      base rsvp-message-type;
    }
    description
      "The RSVP message type that delivered the error";
  }
  leaf rsvp-error-code {
    type uint8;
    description
      "RSVP error code";
    reference
      "RFC2205";
  }
  leaf rsvp-error-subcode {
    type uint16;
    description
      "RSVP Error sub-codes";
    reference
      "RFC2205";
  }
}

augment "/te:te/te:lsps/te:lsp" {
  when "/te:te/te:lsps/te:lsp"
    + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
    description
      "When the signaling protocol is RSVP-TE ";
  }
  description
    "RSVP-TE generic data augmentation pertaining to specific TE
    LSP";
```

```
leaf associated-rsvp-session {
  type leafref {
    path "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol/rsvp:rsvp/"
      + "rsvp:sessions/session-te/tunnel-id";
  }
  config false;
  description
    "If the signalling protocol specified for this path is
    RSVP-TE, this leaf provides a reference to the associated
    session within the RSVP-TE protocol sessions list, such
    that details of the signaling can be retrieved.";
}
uses lsp-properties;
uses rsvp-te-lsp-error-info;
uses lsp-explicit-route-information-state;
uses lsp-record-route-information-state;
}

/* TE interface augmentation */

augment "/te:te/te-dev:interfaces/te-dev:interface" {
  description
    "RSVP-TE generic data augmentation pertaining to specific TE
    interface";
}
}
<CODE ENDS>
```

Figure 4: RSVP TE generic YANG module

2.3.2. RSVP-TE MPLS YANG Module

The RSVP-TE MPLS YANG module "ietf-rsvp-te-mpls" imports the following module(s):

- o ietf-rsvp defined in [I-D.ietf-teas-yang-rsvp]
- o ietf-routing-types defined in [RFC8294]
- o ietf-te-mpls-types defined in [I-D.ietf-teas-yang-te-types]
- o ietf-te and ietf-te-dev defined in [I-D.ietf-teas-yang-te]

This module references the following documents:

[I-D.ietf-teas-yang-rsvp], [RFC8349], [I-D.ietf-teas-yang-te-types],
[I-D.ietf-teas-yang-te], [RFC3209].

```
<CODE BEGINS> file "ietf-rsvp-te-mpls@2021-02-21.yang"
module ietf-rsvp-te-mpls {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te-mpls";
  prefix rsvp-te-mpls;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "draft-ietf-teas-yang-rsvp: A YANG Data Model for
      Resource Reservation Protocol (RSVP)";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management";
  }
  import ietf-te-packet-types {
    prefix te-packet-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-te {
    prefix te;
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }
  import ietf-te-device {
    prefix te-dev;
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }

  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:tasaad.net@gmail.com>

Editor: Rakesh Gandhi
<mailto:rgandhi@cisco.com>

Editor: Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>

Editor: Igor Bryskin
<mailto:Igor.Bryskin@huawei.com>

Editor: Himanshu Shah
<mailto:hshah@ciena.com>";

description

"Latest update to MPLS RSVP-TE YANG data model.

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 2021-02-21 {  
  description  
    "Update to MPLS RSVP-TE YANG initial revision."  
  reference  
    "RFCXXXX: A YANG Data Model for RSVP-TE Protocol";  
}
```

/* RSVP-TE MPLS LSPs groupings */

```
grouping lsp-attributes-flags-mpls {
```

```
    description
      "Configuration parameters relating to RSVP-TE MPLS LSP
       attribute flags";
  }

  grouping lsp-session-attributes-obj-flags-mpls {
    description
      "Configuration parameters relating to RSVP-TE MPLS LSP
       session attribute flags";
    reference
      "RFC4859: Registry for RSVP-TE Session Flags";
    leaf-list session-attribute {
      when "../session-attribute =
        'te-types:bandwidth-protection-desired' or
        ../session-attribute =
        'te-types:soft-preemption-desired'" {
        description
          "Session attributes applicable to mpls technology";
      }
      type identityref {
        base te-types:session-attributes-flags;
      }
    }
    description
      "RSVP session attributes flags";
    reference
      "RFC4859: Registry for RSVP-TE Session Flags";
  }
}

grouping tunnel-properties-mpls {
  description
    "Top level grouping for LSP properties.";
  uses lsp-session-attributes-obj-flags-mpls;
  uses lsp-attributes-flags-mpls;
}

grouping lsp-properties-mpls {
  description
    "Top level grouping for LSP properties.";
  uses lsp-session-attributes-obj-flags-mpls;
  uses lsp-attributes-flags-mpls;
}

/* End of RSVP-TE MPLS LSPs groupings */
/* MPLS RSVP-TE interface groupings */

grouping rsvp-te-interface-state {
  description
```

```
    "The RSVP-TE interface state grouping";
  leaf over-subscribed-bandwidth {
    type te-packet-types:bandwidth-kbps;
    description
      "The amount of over-subscribed bandwidth on
       the interface";
  }
}

grouping rsvp-te-interface-softpreemption-state {
  description
    "The RSVP-TE interface preeemptions state grouping";
  container interface-softpreemption-state {
    description
      "The RSVP-TE interface preeemptions state grouping";
    leaf soft-preempted-bandwidth {
      type te-packet-types:bandwidth-kbps;
      description
        "The amount of soft-preempted bandwidth on
         this interface";
    }
    list lsps {
      key "source destination tunnel-id lsp-id "
        + "extended-tunnel-id";
      description
        "List of LSPs that are soft-preempted";
      leaf source {
        type leafref {
          path "/te:te/te:lsps/te:lsp/"
            + "te:source";
        }
        description
          "Tunnel sender address extracted from
           SENDER_TEMPLATE object";
        reference
          "RFC3209";
      }
      leaf destination {
        type leafref {
          path "/te:te/te:lsps/te:lsp/"
            + "te:destination";
        }
        description
          "Tunnel endpoint address extracted from
           SESSION object";
        reference
          "RFC3209";
      }
    }
  }
}
```

```
    leaf tunnel-id {
      type leafref {
        path "/te:te/te:lsps/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/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/te:lsp/"
          + "te:extended-tunnel-id";
      }
      description
        "Extended Tunnel ID of the LSP.";
      reference
        "RFC3209";
    }
    leaf type {
      type leafref {
        path "/te:te/te:lsps/te:lsp/"
          + "te:type";
      }
      description
        "LSP type P2P or P2MP";
    }
  }
}

grouping bandwidth-mpls-constraints {
```

```
description
  "Bandwidth constraints.";
container bandwidth-mpls-constraints {
  description
    "Holds the bandwidth constraints properties";
  leaf maximum-reservable {
    type te-packet-types:bandwidth-kbps;
    description
      "The maximum reservable bandwidth on the
       interface in kbps";
  }
  leaf-list bc-value {
    type uint32 {
      range "0..4294967295";
    }
    max-elements 8;
    description
      "The bandwidth constraint type";
  }
}

grouping bandwidth-constraint-values {
  description
    "Packet bandwidth constraints values";
  choice value-type {
    description
      "Value representation";
    case percentages {
      container perc-values {
        uses bandwidth-mpls-constraints;
        description
          "Percentage values";
      }
    }
    case absolutes {
      container abs-values {
        uses bandwidth-mpls-constraints;
        description
          "Absolute values";
      }
    }
  }
}

grouping bandwidth-mpls-reservable {
  description
    "Interface bandwidth reservable configuration grouping";
```

```
container bandwidth-mpls-reservable {
  description
    "Interface bandwidth reservable container";
  choice bandwidth-value {
    description
      "Reservable bandwidth configuration choice";
    case absolute {
      leaf absolute-value {
        type te-packet-types:bandwidth-kbps;
        description
          "Absolute value of the bandwidth";
      }
    }
    case percentage {
      leaf percent-value {
        type uint32 {
          range "0..4294967295";
        }
        description
          "Percentage reservable bandwidth";
      }
    }
    description
      "The maximum reservable bandwidth on the
      interface";
  }
}
choice bc-model-type {
  description
    "Reservable bandwidth percentage capacity
    values.";
  case bc-model-rdm {
    container bc-model-rdm {
      description
        "Russian Doll Model Bandwidth Constraints.";
      uses bandwidth-mpls-constraints;
    }
  }
  case bc-model-mam {
    container bc-model-mam {
      uses bandwidth-mpls-constraints;
      description
        "Maximum Allocation Model Bandwidth
        Constraints.";
    }
  }
  case bc-model-mar {
    container bc-model-mar {
      uses bandwidth-mpls-constraints;
    }
  }
}
```

```
        description
          "Maximum Allocation with Reservation Model
           Bandwidth Constraints.";
      }
    }
  }
}

/* End of RSVP-TE interface groupings */
/* RSVP-TE FRR groupings */

grouping rsvp-te-frr-auto-tunnel-backup {
  description
    "Auto-tunnel backup configuration grouping";
  leaf auto-backup-protection {
    type identityref {
      base te-packet-types:backup-protection-type;
    }
    default "te-packet-types:backup-protection-node-link";
    description
      "Describes whether the backup should offer
       protection against link, node, or either";
  }
  leaf auto-backup-path-computation {
    type identityref {
      base te-types:path-computation-srlg-type;
    }
    description
      "FRR backup computation type";
  }
}

grouping rsvp-te-frr-backups {
  description
    "Top level container for RSVP-TE FRR backup parameters";
  container rsvp-te-frr-backups {
    description
      "RSVP-TE facility backup properties";
    choice type {
      description
        "FRR backup tunnel type";
      case static-tunnel {
        container static-backups {
          description
            "List of static backups";
          list static-backup {
            key "backup-tunnel-name";
          }
        }
      }
    }
  }
}
```

```
        description
            "List of static backup tunnels that
            protect the RSVP-TE interface.";
        leaf backup-tunnel-name {
            type leafref {
                path "/te:te/te:tunnels/te:tunnel/te:name";
            }
            description
                "FRR Backup tunnel name";
        }
    }
}

case auto-tunnel {
    container auto-tunnel-backups {
        description
            "Auto-tunnel choice";
        uses rsvp-te-frr-auto-tunnel-backup;
    }
}

}

}

grouping lsp-backup-info-state {
    description
        "LSP backup information grouping";
    leaf backup-tunnel-name {
        type string;
        description
            "If an LSP has an FRR backup LSP that can protect it,
            this field identifies the tunnel name of the backup LSP.
            Otherwise, this field is empty.";
    }
    leaf backup-frr-on {
        type uint8;
        description
            "Whether currently this backup is carrying traffic";
    }
    leaf backup-protected-lsp-num {
        type uint32;
        description
            "Number of LSPs protected by this backup";
    }
}

grouping lsp-backup-info {
    description
```



```
        "Backup/bypass LSP related information";
    container backup-info {
        description
            "backup information";
        uses lsp-backup-info-state;
    }
}

/**** End of RSVP-TE FRR backup information ****/

/* RSVP-TE global properties */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp" {
    description
        "RSVP-TE augmentation to RSVP globals";
    leaf rsvp-frr-local-revert-delay {
        type uint32;
        description
            "Time to wait after primary link is restored
            before node attempts local revertive
            procedures.";
    }
}

/* Linkage to the base RSVP all interfaces */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
    description
        "Augmentations for RSVP-TE MPLS all interfaces properties";
    /* To be added */
}

/* Linkage to per RSVP interface */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
    + "rsvp:interface" {
    description
        "Augmentations for RSVP-TE MPLS per interface properties";
    /* To be added */
}

/* add augmentation for sessions neighbors */

augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/"
```

```
    + "rsvp:sessions" {
      description
        "Augmentation for RSVP-TE MPLS sessions";
      /* To be added */
    }

    augment "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors" {
        description
          "Augmentations for RSVP-TE MPLS neighbors properties";
        /* To be added */
      }

    /**
     * Augmentation to TE generic module
     */

    augment "/te:te/te:tunnels/te:tunnel" {
      description
        "Augmentations for RSVP-TE MPLS TE tunnel properties";
      uses tunnel-properties-mpls;
    }

    augment "/te:te/te:lsps/te:lsp" {
      when "/te:te/te:lsps/te:lsp"
        + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
        description
          "When the signaling protocol is RSVP-TE ";
      }
      description
        "RSP-TE MPLS LSP state properties";
      uses lsp-properties-mpls;
      uses lsp-backup-info;
    }

    augment "/te:te/te:tunnels/te:tunnel/te:primary-paths"
      + "/te:primary-path/te:lsps/te:lsp" {
      when "/te:te/te:tunnels/te:tunnel"
        + "/te:secondary-paths/te:secondary-path/"
        + "te:signaling-type = 'te-types:path-setup-rsvp'" {
        description
          "When the signaling protocol is RSVP-TE ";
      }
      description
        "RSVP-TE MPLS LSP state properties";
      uses lsp-properties-mpls;
      uses lsp-backup-info;
    }
  }
```

```
augment "/te:te/te:tunnels/te:tunnel/te:secondary-paths"
  + "/te:secondary-path/te:lsps/te:lsp" {
  when "/te:te/te:tunnels/te:tunnel"
  + "/te:secondary-paths/te:secondary-path/"
  + "te:signaling-type = 'te-types:path-setup-rsvp'" {
    description
      "When the signaling protocol is RSVP-TE ";
  }
  description
    "RSVP-TE MPLS LSP state properties";
  uses lsp-properties-mpls;
  uses lsp-backup-info;
}

augment "/te:te/te-dev:interfaces/te-dev:interface" {
  description
    "RSVP reservable bandwidth configuration properties";
  uses bandwidth-mpls-reservable;
}

augment "/te:te/te-dev:interfaces/te-dev:interface" {
  description
    "RSVP reservable bandwidth configuration properties";
  uses rsvp-te-frr-backups;
}
}
<CODE ENDS>
```

Figure 5: RSVP TE MPLS YANG module

3. 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-te
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-te-mpls
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-te
namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-te
prefix:    rsvp-te
reference:  RFCXXXX

name:      ietf-rsvp-te-mpls
namespace: urn:ietf:params:xml:ns:yang:ietf-rsvp-te-mpls
prefix:    rsvp-te-mpls
reference:  RFCXXXX
```

4. 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 [RFC8341] 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(s) defined in this document 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.

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/globals: The data nodes defined in this document and under this branch are applicable device-wide and can affect all RSVP established sessions. Unauthorized access to this container can potentially cause disruptive event(s) on all established sessions.

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/globals/rsvp:sessions: The data nodes defined in this document and under this branch are applicable to one or all RSVP-TE session(s). Unauthorized access to this container can potentially affect the impacted RSVP session(s).

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces: The data nodes defined in this document and under this branch are applicable to one or all RSVP interfaces. Unauthorized access to this container can potentially affect established session(s) over impacted interface(s).

5. Acknowledgement

The authors would like to thank Lou Berger for reviewing and providing valuable feedback on this document.

6. 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

7. References

7.1. Normative References

[I-D.ietf-teas-yang-rsvp]
Beeram, V., Saad, T., Gandhi, R., Liu, X., and I. Bryskin,
"A YANG Data Model for Resource Reservation Protocol
(RSVP)", draft-ietf-teas-yang-rsvp-15 (work in progress),
September 2020.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin,
"A YANG Data Model for Traffic Engineering Tunnels, Label
Switched Paths and Interfaces", draft-ietf-teas-yang-te-25
(work in progress), July 2020.

[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-13 (work in progress), November 2019.

- [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>>.
- [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>>.
- [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>>.
- [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>>.

7.2. Informative References

- [RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <<https://www.rfc-editor.org/info/rfc2210>>.
- [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>>.
- [RFC4859] Farrel, A., "Codepoint Registry for the Flags Field in the Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Session Attribute Object", RFC 4859, DOI 10.17487/RFC4859, April 2007, <<https://www.rfc-editor.org/info/rfc4859>>.
- [RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, DOI 10.17487/RFC4920, July 2007, <<https://www.rfc-editor.org/info/rfc4920>>.
- [RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangar, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, DOI 10.17487/RFC5420, February 2009, <<https://www.rfc-editor.org/info/rfc5420>>.
- [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>>.

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
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Himanshu Shah
Ciena

Email: hshah@ciena.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].

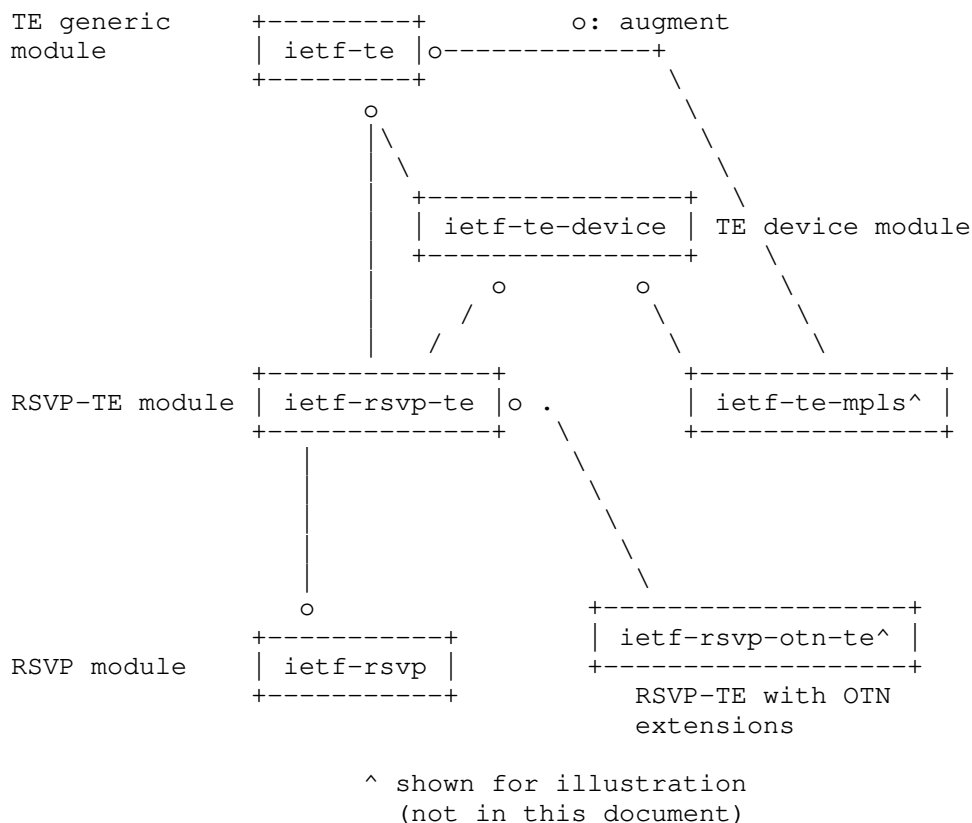


Figure 1: Relationship of TE module(s) with signaling protocol modules

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?
|           |   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 direction?
+--rw rt-types:generalized-label
+--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>

					<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 </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? 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] </pre>

```

+---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
+---ro               uint32
+---ro (type)?
+---: (numbered-node-hop)
+---ro numbered-node-hop
+---ro node-id
+---ro           te-node-id
+---ro hop-type?
+---ro           te-hop-type
+---: (numbered-link-hop)
+---ro numbered-link-hop
+---ro link-tp-id
+---ro           te-tp-id
+---ro hop-type?
+---ro           te-hop-type
+---ro direction?
+---ro           te-link-direction
+---: (unnumbered-link-hop)
+---ro unnumbered-link-hop
+---ro link-tp-id
+---ro           te-tp-id
+---ro node-id
+---ro           te-node-id
+---ro hop-type?
+---ro           te-hop-type
+---ro direction?
+---ro           te-link-direction
+---: (as-number)
+---ro as-number-hop
+---ro as-number
+---ro           inet:as-number
+---ro hop-type?
+---ro           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)

```

+--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 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/lsps/tunnel-name
| | +---ro node -> /te/lsp/lsps/node
| | +---ro lsp-id -> /te/lsp/lsps/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)

```

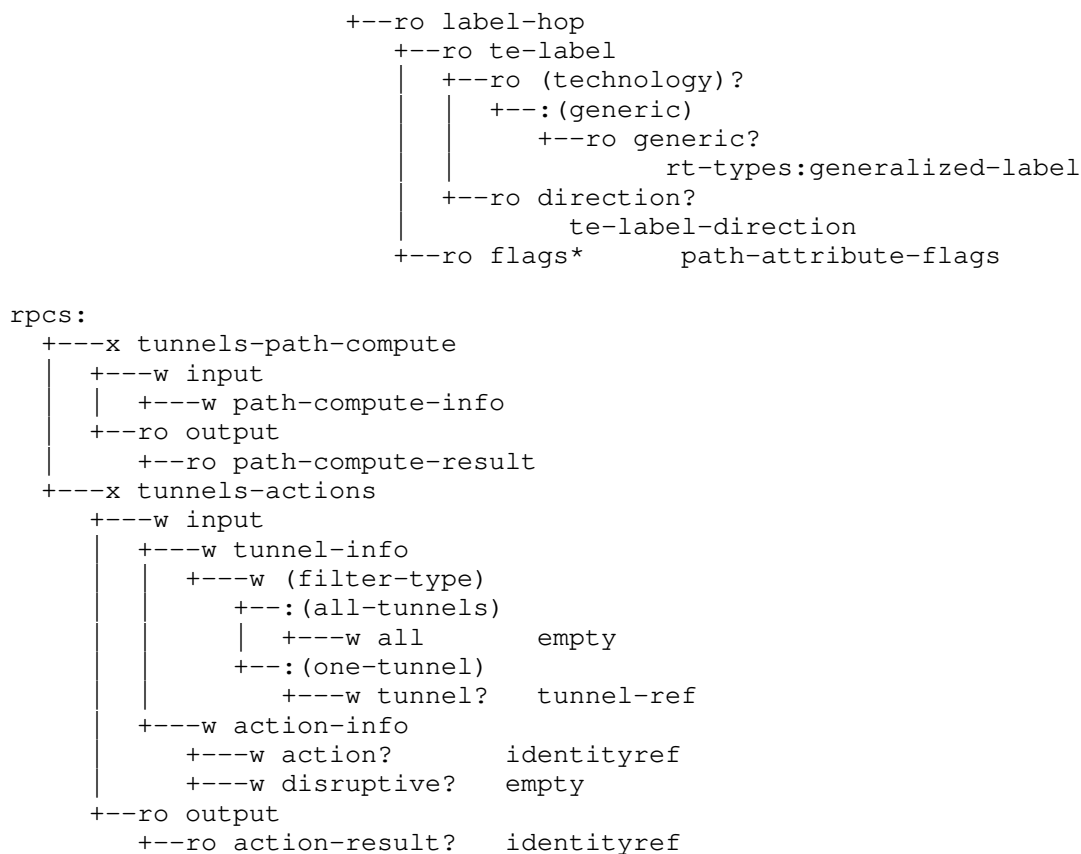


Figure 8: TE Tunnel generic model YANG tree diagram

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 lsp-grouping {
 description
 "LSPs state operational data grouping.";
 container lsp {
 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;
}
```

```

/* TE LSPs device state augmentation */

augment "/te:te/te:lsps/te:lsp" {
 description
 "TE LSP device dependent augmentation.";
 uses lsp-device-info;
}

/* TE interfaces RPCs/execution Data */

rpc link-state-update {
 description
 "Triggers a link state update for the specific interface.";
 input {
 choice filter-type {
 mandatory true;
 description
 "Filter choice.";
 case match-all {
 leaf all {
 type empty;
 mandatory true;
 description
 "Match all TE interfaces.";
 }
 }
 case match-one-interface {
 leaf interface {
 type if:interface-ref;
 description
 "Match a specific TE interface.";
 }
 }
 }
 }
}
}

<CODE ENDS>

```

Figure 12: TE device data model YANG module

## 7. Notifications

Notifications are a key component of any topology data model.

[RFC8639] and [RFC8641] define a subscription mechanism and a push mechanism for YANG datastores. These mechanisms currently allow the user to:

- \* 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.

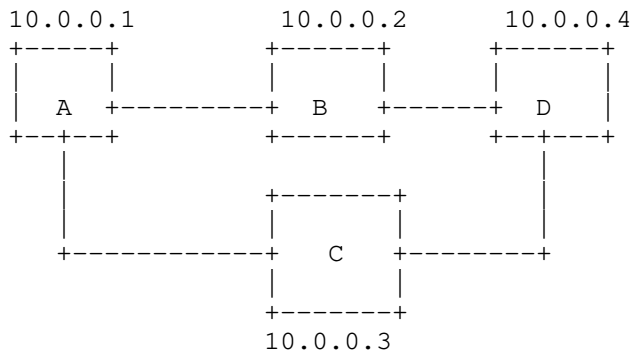


Figure 13: TE network used in data tree examples

### 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  
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

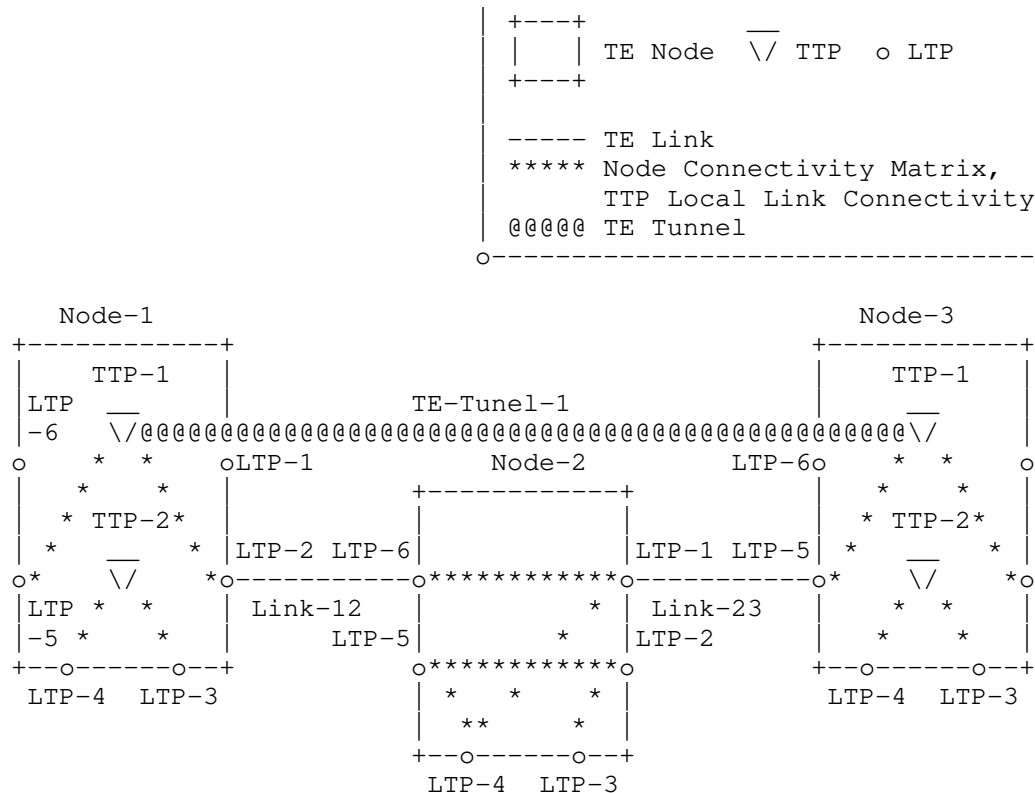


Figure 1: TE Topology Modeling Abstractions

### 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.

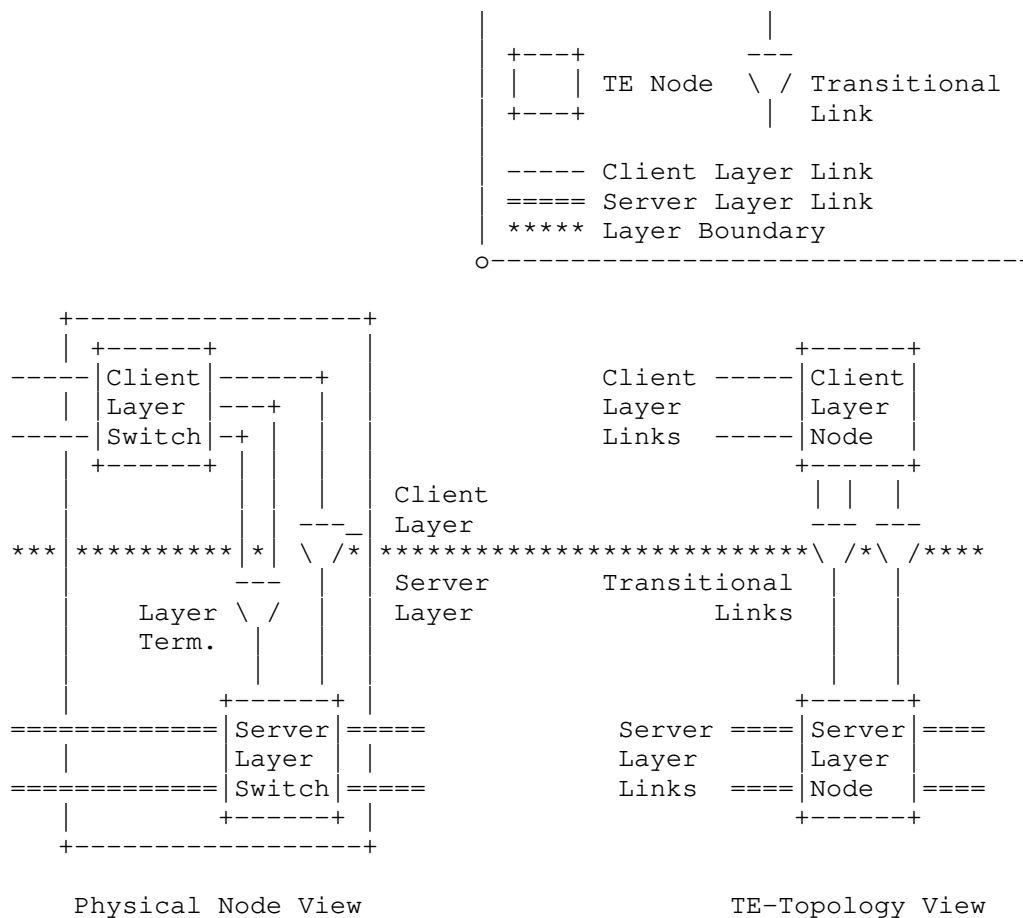


Figure 2: Modeling a Multi-Layer Node (Dual-Layer Example)

### 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.

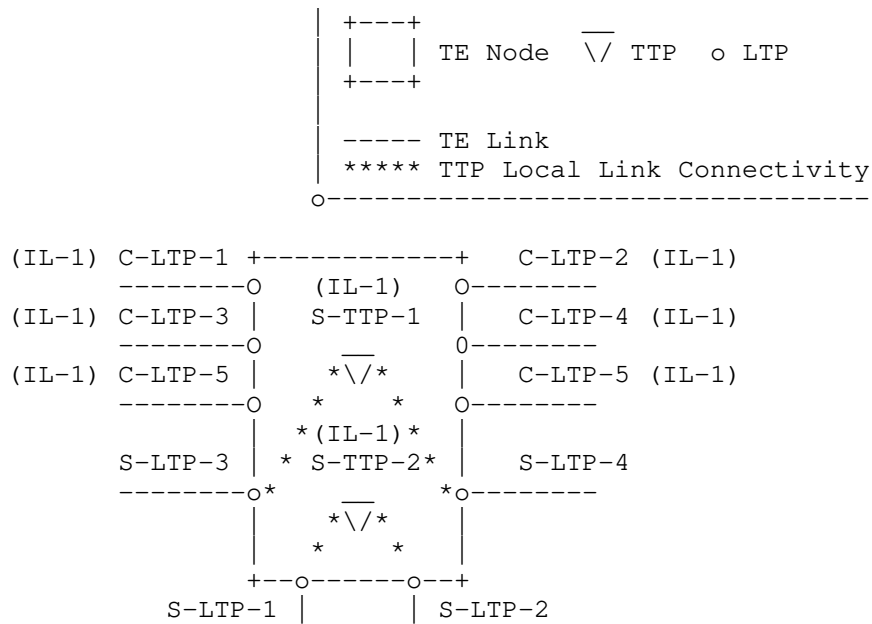


Figure 3: TE Inter-Layer Lock ID Associations

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.

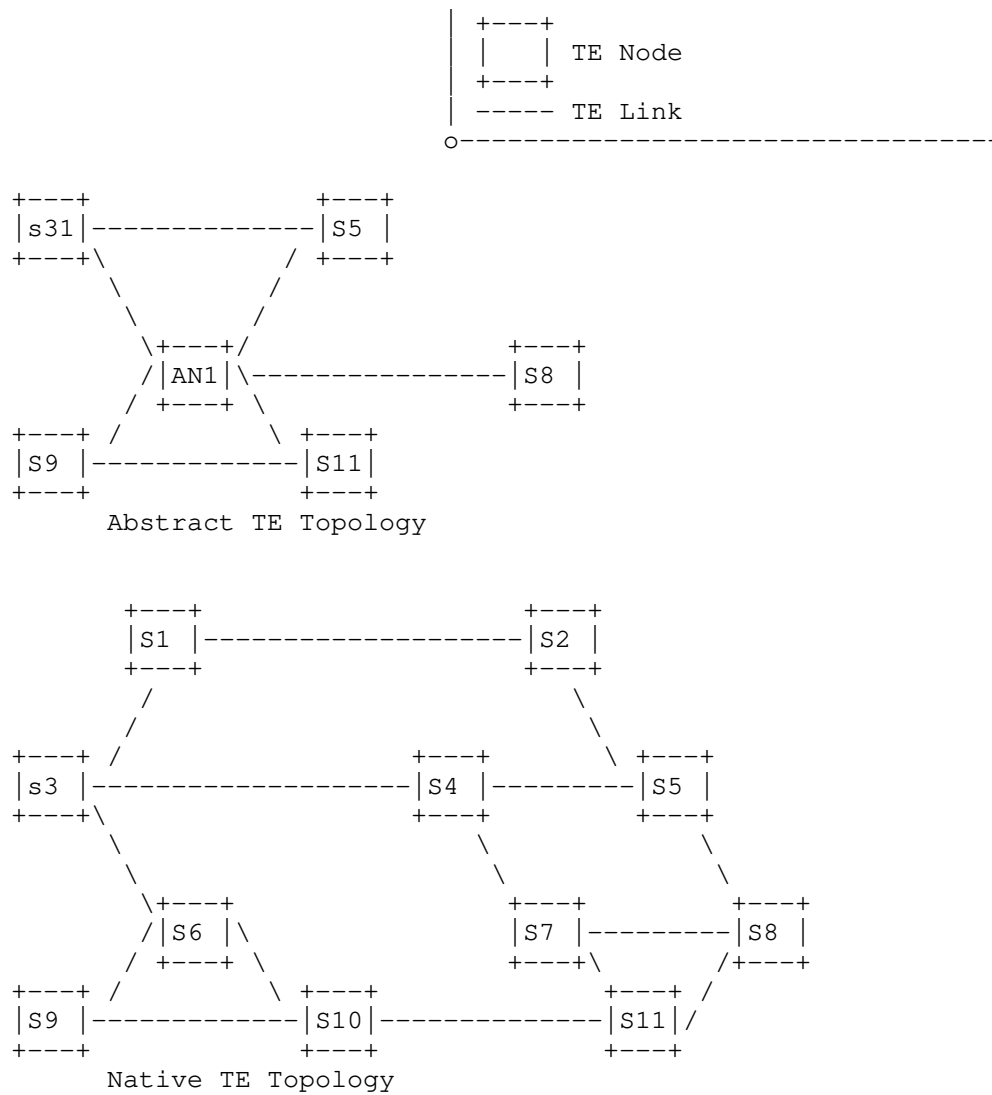


Figure 4: Abstract TE Topology

#### 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.

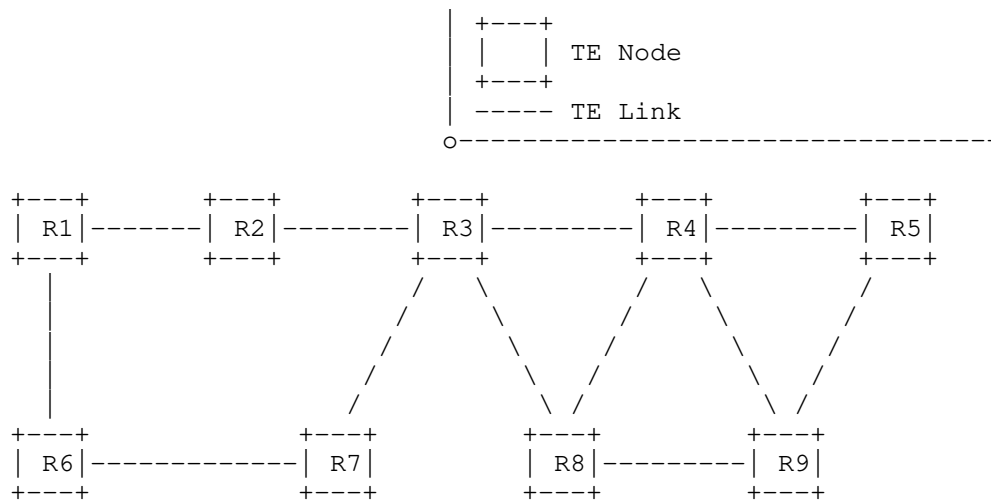


Figure 5a: Example Network Topology

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.

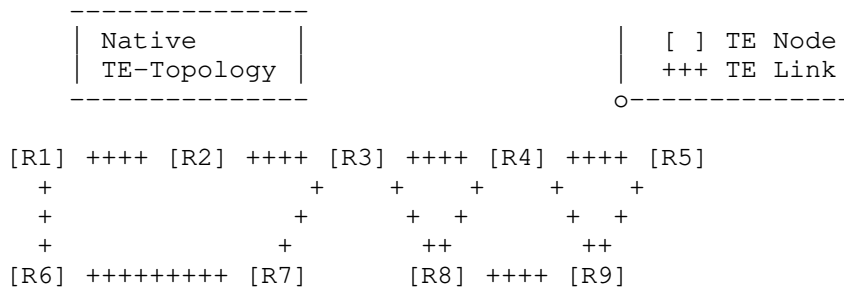


Figure 5b: Native TE Topology as seen on Node R3

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.

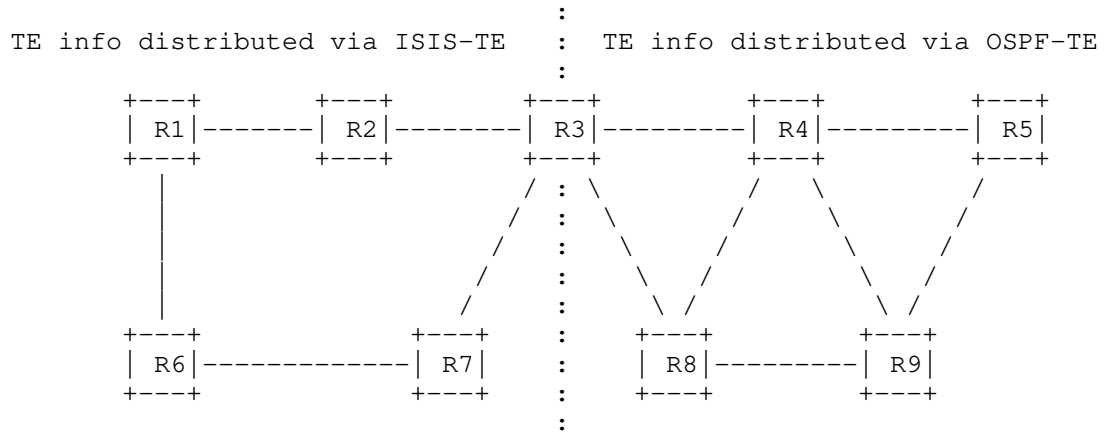


Figure 6a: Example Network Topology

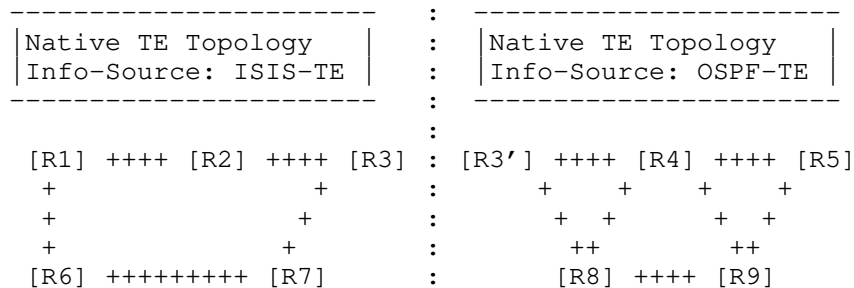


Figure 6b: Native TE Topologies as seen on Node R3

#### 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.

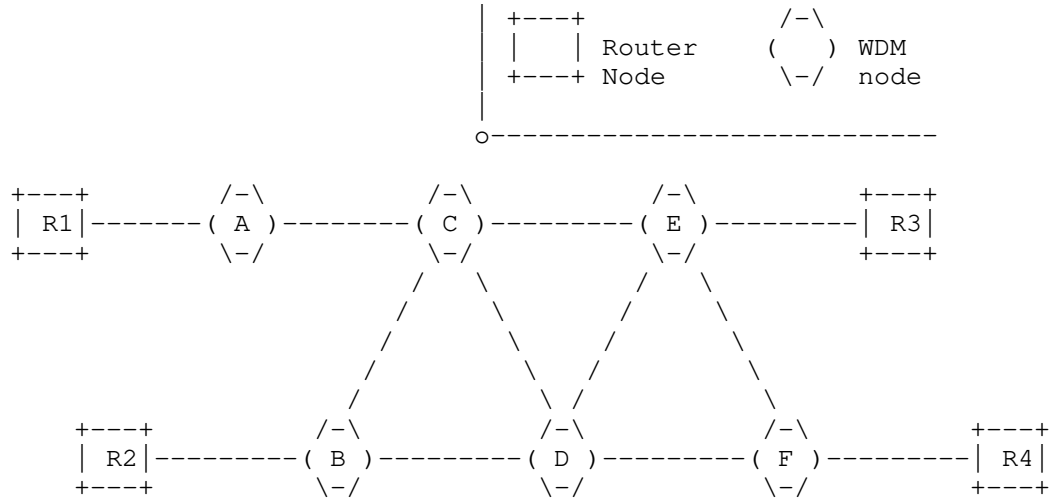


Figure 7: Example packet optical topology

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.

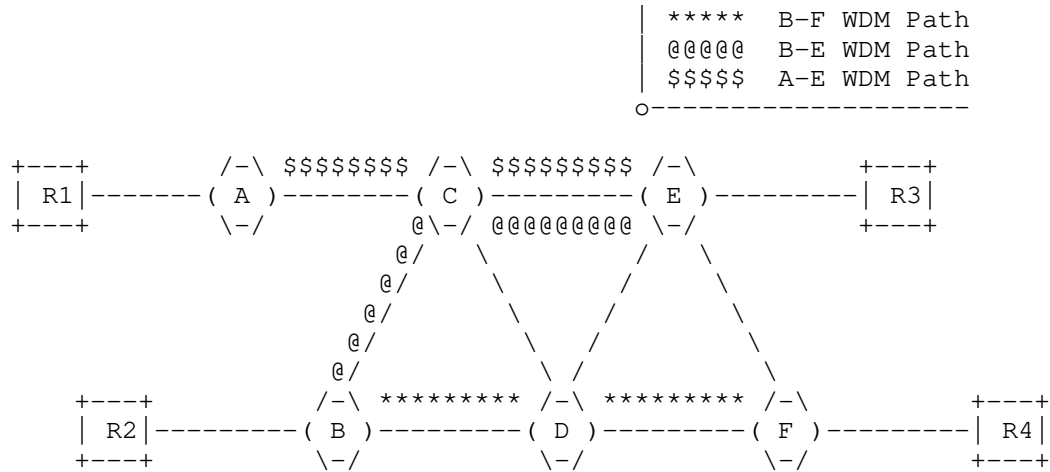


Figure 8a: Paths within the provider domain

```

+++++++ [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.

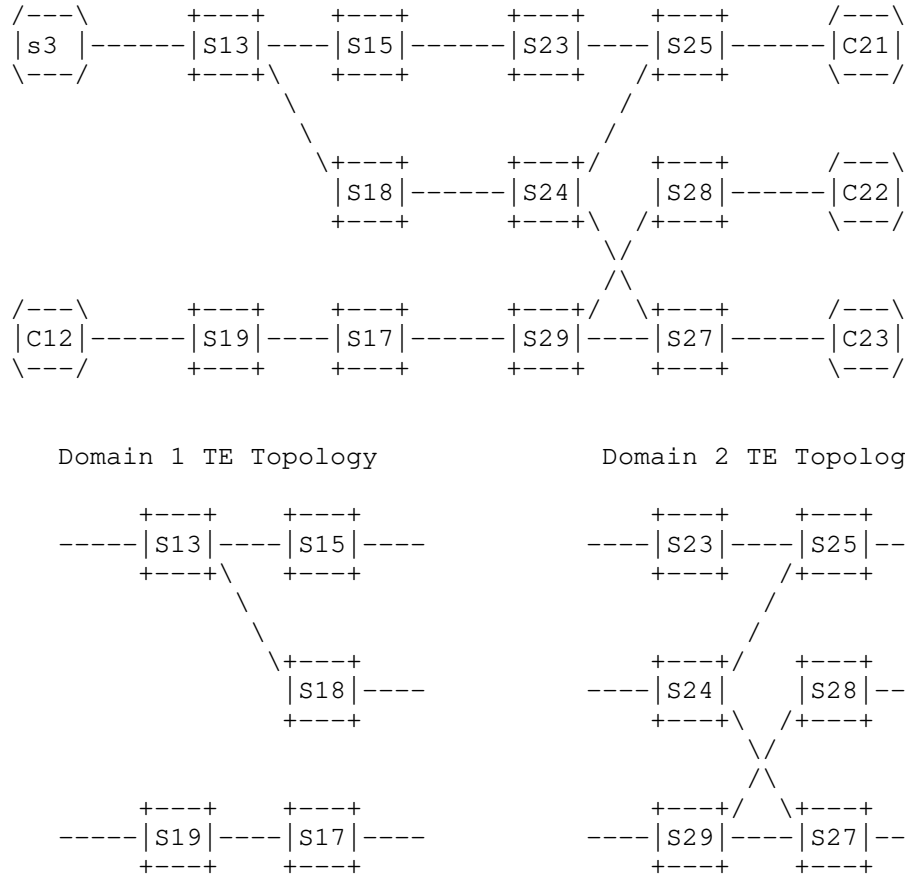


Figure 9: Merging Domain TE Topologies

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

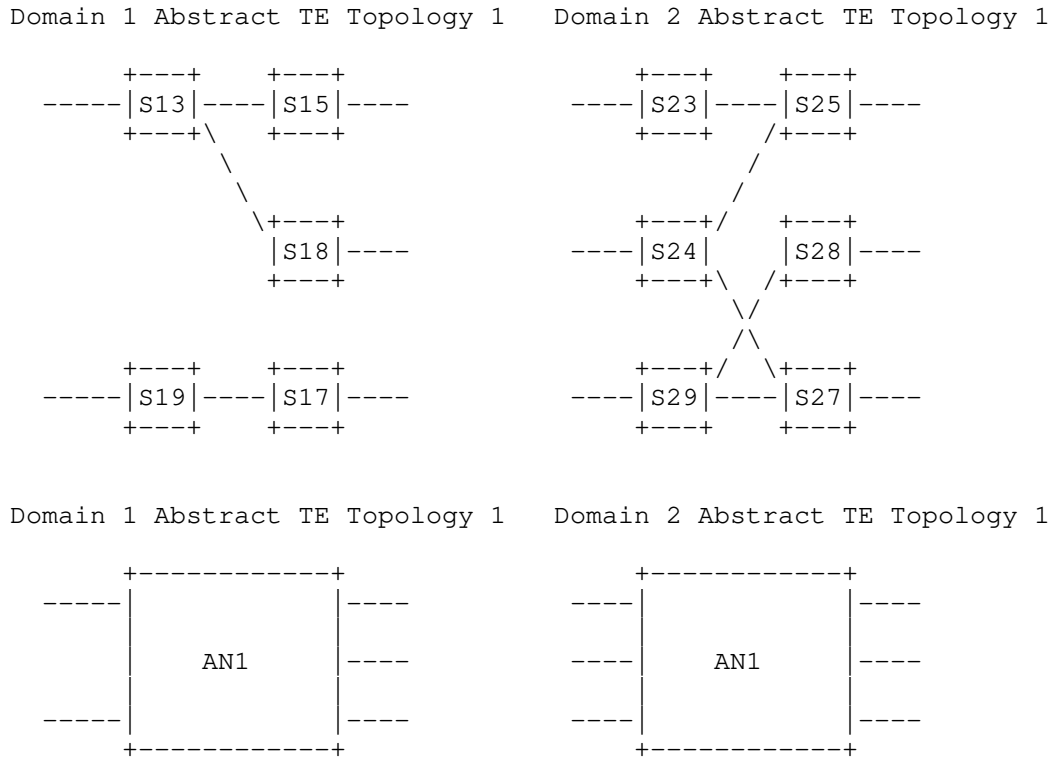


Figure 10: Merging Domain TE Topologies

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.



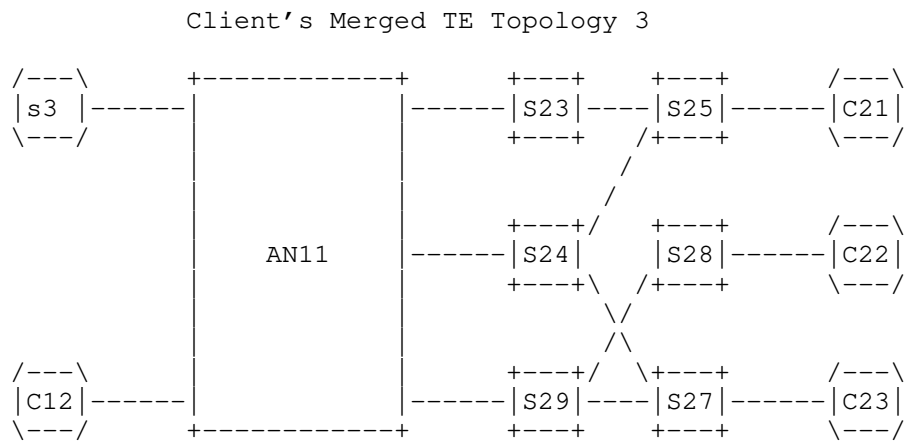
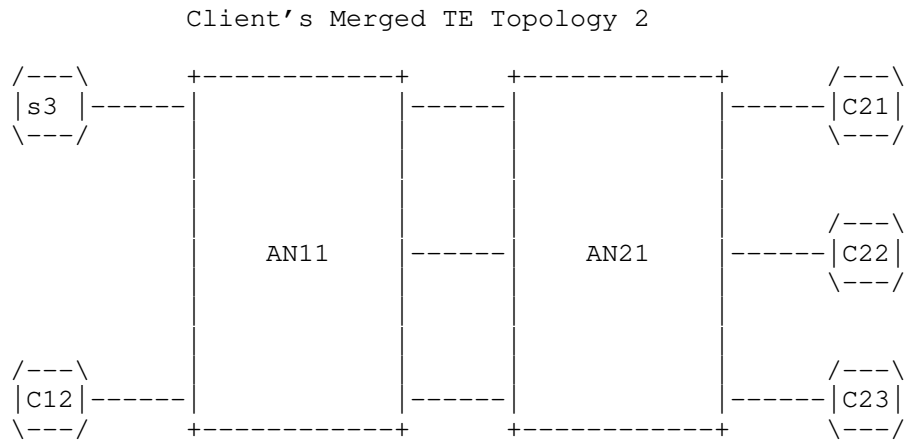


Figure 11: Multiple Native (Merged) Client's TE Topologies

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].

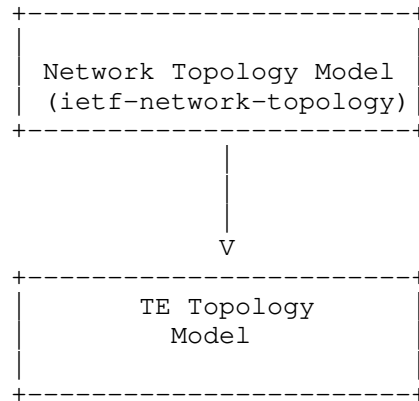


Figure 12: Augmenting the Network Topology Model

### 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.

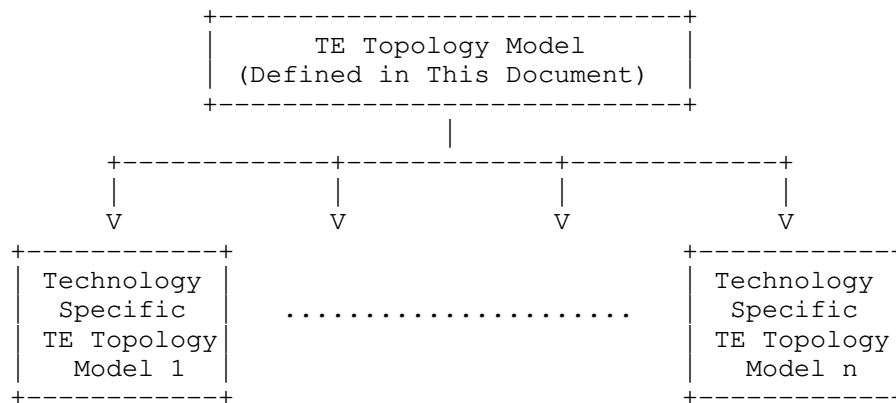


Figure 13: Augmenting the Technology agnostic TE Topology model

### 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-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 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 | 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
| |
| 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

```

```

+---:(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)?
+---rw (generic)
+---rw generic?
+---rw rt-
types:generalized-label
+---rw direction?
+---rw 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)?
+---rw (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)?

```

[illegible]

```

+--:(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                      |                     |



```

| | 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 tiebreakers |
| +---ro tiebreaker* [tiebreaker-type]
| +---ro tiebreaker-type identityref
+---:(objective-function) |
| {path-optimization-objective-function}?
| +---ro objective-function
| +---ro 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

```

|       |  |  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|--|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |  |  | <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

```

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| types:generalized-label | <pre> +--:(label)   +--ro label-hop     +--ro te-label       +--ro (technology)?         +--:(generic)           +--ro generic?             rt-            +--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 </pre> |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

```

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?<br> <br>te-hop-type<br>+--ro direction?<br>te-link-direction                                       |
|                         |  |  |  |                                      | ---:(as-number)<br>+--ro as-number-hop<br>+--ro as-number<br> <br>inet:as-number<br>+--ro hop-type?<br>te-hop-type |
|                         |  |  |  |                                      | ---:(label)<br>+--ro label-hop<br>+--ro te-label<br>+--ro (technology)?<br> <br>---:(generic)<br>+--ro generic?    |
|                         |  |  |  |                                      | rt-                                                                                                                |
| types:generalized-label |  |  |  |                                      |                                                                                                                    |
|                         |  |  |  |                                      | +--ro direction?<br>te-label-                                                                                      |
| direction               |  |  |  |                                      |                                                                                                                    |
|                         |  |  |  |                                      | ---:(srllg)                                                                                                        |
|                         |  |  |  |                                      | +--ro srllg                                                                                                        |
|                         |  |  |  |                                      | +--ro srllg? 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<br>+--ro direction?<br> <br>te-link-direction<br>+---:(unnumbered-link-hop)<br>+--ro unnumbered-link-hop<br>+--ro link-tp-id<br> <br>te-tp-id<br>+--ro node-id<br> <br>te-node-id<br>+--ro hop-type?<br> <br>te-hop-type<br>+--ro direction?<br> <br>te-link-direction<br>+---:(as-number)<br>+--ro as-number-hop<br>+--ro as-number<br> <br>inet:as-number<br>+--ro hop-type?<br> <br>te-hop-type<br>+---:(label)<br>+--ro label-hop<br>+--ro te-label<br>+--ro (technology)?<br>+---:(generic)<br>+--ro generic?<br>rt- |
| types:generalized-label |  |  |  |  |  |  | +--ro direction?<br>te-label-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| direction               |  |  |  |  |  |  | +--ro tiebreakers<br>+--ro tiebreaker* [tiebreaker-type]<br>+--ro tiebreaker-type     identityref<br>+---:(objective-function)<br>{path-optimization-objective-                                                                                                                                                                                                                                                                                                                                                                       |
| function}?              |  |  |  |  |  |  | +--ro objective-function<br>+--ro objective-function-type?<br>identityref<br><br>+--ro path-properties<br>+--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
+--ro domain-id? uint32
+--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]                              |                             |



|  |  |  |  |  |                                     |
|--|--|--|--|--|-------------------------------------|
|  |  |  |  |  | +--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                  |
|  |  |  |  |  | +--rw tiebreakers                   |
|  |  |  |  |  | +--rw tiebreaker* [tiebreaker-type] |

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

```

```

| | | | | +--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 local-link-connectivity* [link-tp-ref]
| | | | | +--rw link-tp-ref
| | | | | | -> ../../../../nt:termination-point/tp-id
+--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 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                        |                     |                  |



|                         |  |  |  |  |  |                                      |
|-------------------------|--|--|--|--|--|--------------------------------------|
|                         |  |  |  |  |  | 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
| +--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 ".../.../.../.../.../.../.../.../.../.../..." /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: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



TEAS WG

Internet Draft  
Intended status: Informational

Young Lee  
Dhruv Dhody  
Huawei

Daniele Ceccarelli  
Ericsson

Oscar Gonzalez de Dios  
Telefonica

Expires: December 2017

June 5, 2017

Abstraction and Control of TE Networks (ACTN) Abstraction Methods  
draft-lee-teas-actn-abstraction-02

#### 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 December 5, 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

Abstraction and Control of Traffic Engineering (TE) Networks (ACTN) refers to the set of virtual network operations needed to orchestrate, control and manage large-scale multi-domain TE networks, so as to facilitate network programmability, automation, and efficient resource sharing.

As the ACTN architecture considers abstraction as one of the important building blocks, this document describes a few alternatives methods of abstraction for both packet and optical networks. This is an important consideration since the choice of the abstraction method impacts protocol design and the information it carries.

## Table of Contents

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| 1. Introduction.....                                                                       | 3  |
| 2. Abstraction Factors in ACTN Architecture.....                                           | 3  |
| 3. Build Method of Grey Topology.....                                                      | 6  |
| 3.1. Automatic generation of abstract topology by configuration                            | 6  |
| 3.2. On-demand generation of supplementary topology via path<br>compute request/reply..... | 6  |
| 4. Protocol/Data Model Requirements.....                                                   | 8  |
| 4.1. Packet Networks.....                                                                  | 8  |
| 4.2. OTN Networks.....                                                                     | 8  |
| 4.3. WSON Networks.....                                                                    | 9  |
| 5. Acknowledgements.....                                                                   | 11 |
| 6. References.....                                                                         | 11 |

|                                  |    |
|----------------------------------|----|
| 6.1. Informative References..... | 11 |
| 7. Contributors.....             | 11 |
| Authors' Addresses.....          | 12 |
| Appendix A:.....                 | 12 |

## 1. Introduction

Abstraction and Control of TE Networks (ACTN) describes a method for operating a Traffic Engineered (TE) network (such as an MPLS-TE network or a layer 1 transport network) to provide connectivity and virtual network services for customers of the TE network. The services provided can be tuned to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers. More details about ACTN can be found in Section 2.

Abstraction is defined in [RFC7926] as:

Abstraction is the process of applying policy to the available TE information within a domain, to produce selective information that represents the potential ability to connect across the domain. Thus, abstraction does not necessarily offer all possible connectivity options, but presents a general view of potential connectivity according to the policies that determine how the domain's administrator wants to allow the domain resources to be used.

Connectivity referred to this document is TE path through a series of connected domains as used in [RFC7926].

As the ACTN architecture considers abstraction as one of the important building blocks, this document discusses a few alternatives for the methods of abstraction for both packet and optical networks. This is an important consideration since the choice of the abstraction method impacts protocol design and the information it carries.

The purpose of this document is to find a common agreement on the factors and methods of abstraction. These abstraction factors and methods may in turn impact implementations and protocol design.

## 2. Abstraction Factors in ACTN Architecture

This section provides abstraction factors in the ACTN architecture. [ACTN-Frame] describes the architecture model for ACTN including the entities (Customer Network Controller (CNC), Multi-domain Service



Coordinator (MDSC), and Physical Network Controller (PNC) and their interfaces.

The MDSC oversees the specific aspects of the different domains and builds a single abstracted end-to-end network topology in order to coordinate end-to-end path computation and path/service provisioning. In order for the MDSC to perform its coordination function, it depends on the coordination with the PNCs which are the domain-level controllers especially as to what level of domain network resource abstraction is agreed upon between the MDSC and the PNCs.

As discussed in [RFC7926], abstraction is tied with policy of the networks. For instance, per an operational policy, the PNC would not be allowed to provide any technology specific details (e.g., optical parameters for WSON) in its update. In such case, the abstraction level of the update will be in a generic nature. In order for the MDSC to get technology specific topology information from the PNC, a request/reply mechanism may be employed.

In some cases, abstraction is also tied with the controller's capability of abstraction as abstraction involves some rules and algorithms to be applied to the actual network resource information (which is also known as network topology).

[TE-Topology] describes YANG models for TE-network abstraction. [PCEP-LS] describes PCEP Link-state mechanism that also allows for transport of abstract topology in the context of Hierarchical PCE.

There are factors that may impact the choice of abstraction and presents a number of abstraction methods. It is important to understand that abstraction depends on several factors:

- The nature of underlying domain networks: Abstraction depends on the nature of the underlying domain networks. For instance, packet networks may have different level of abstraction requirements from that of optical networks. Within optical networks, WSON may have different level of abstraction requirements than the OTN networks.
- The capability of the PNC: Abstraction depends on the capability of the PNCs. As abstraction requires hiding details of the underlying resource network resource information, the PNC capability to run some internal optimization algorithm impacts the feasibility of abstraction. Some PNC may not have the ability to abstract native topology while other PNCs may have such an ability to abstract actual topology by using sophisticated algorithms.

- Scalability factor: Abstraction is a function of scalability. If the actual network resource information is of small size, then the need for abstraction would be less than the case where the native network resource information is of large size. In some cases, abstraction may not be needed at all.
- The frequency of topology updates: The proper abstraction level may depend on the frequency of topology updates and vice versa.
- The capability/nature of the MDSC: The nature of the MDSC impacts the degree/level of abstraction. If the MDSC is not capable of handling optical parameters such as those specific to OTN/WSN, then white topology abstraction may not work well.
- The confidentiality: In some cases where the PNC would like to hide key internal topological data from the MDSC, the abstraction method should consider this aspect.
- The scope of abstraction: All of the aforementioned factors are equally applicable to both the MPI (MDSC-PNC Interface) and the CMI (CNC-MDSC Interface).

[ACTN-Framework] defined the following three levels of topology abstraction and their descriptions:

- . White topology: this is a case where the PNC provides the actual network topology to the MDSC without any hiding or filtering.
- . Black topology: the entire domain network is abstracted as a single virtual node (see the definition of virtual node in [RFC7926]) with the access/egress links without disclosing any node internal connectivity information.
- . Grey topology: this abstraction level is between black topology and white topology from a granularity point of view. we may further differentiate from a perspective of how to abstract internal TE resources between the pairs of border nodes:
  - o Grey topology type A: border nodes with a TE links between them in a full mesh fashion
  - o Grey topology type B: border nodes with some internal abstracted nodes and abstracted links

### 3. Build Method of Grey Topology

This section discusses two different methods of building a grey topology:

- . Automatic generation of abstract topology by configuration (Section 3.1)
- . On-demand generation of supplementary topology via path computation request/reply (Section 3.2)

#### 3.1. Automatic generation of abstract topology by configuration

The "Automatic generation" method is based on the abstraction/summarization of the whole domain by the PNC and its advertisement on MPI interface once the abstraction level is configured. The level of abstraction advertisement can be decided based on some PNC configuration parameters (e.g. provide the potential connectivity between any PE and any ASBR in an MPLS-TE network as described in section 3.3.1)

Note that the configuration parameters for this potential topology can include available B/W, latency, or any combination of defined parameters. How to generate such tunnel information is beyond the scope of this document. Appendix A provides one example of this method for the WSON case.

Such potential topology needs to be periodically or incrementally/asynchronously updated every time that a failure, a recovery or the setup of new VNs causes a change in the characteristics of the advertised grey topology (e.g. in our previous case if due to changes in the network is it now possible to provide connectivity between a given PE and a given ASBR with a higher delay in the update).

#### 3.2. On-demand generation of supplementary topology via path compute request/reply

The "on-demand generation" of supplementary topology is to be distinguished from automatic generation of abstract topology. While abstract topology is generated and updated automatically by configuration as explained in Section 3.1., additional supplementary topology may be obtained by the MDSC via path compute request/reply mechanism. Starting with a black topology advertisement from the

PNCs, the MDSC may need additional information beyond the level of black topology from the PNCs. It is assumed that the black topology advertisement from PNCs would give the MDSC each domain's the border node/link information as described in Figure 2. Under this scenario, when the MDSC needs to allocate a new VN, the MDSC can issue a number of Path Computation requests as described in [ACTN-YANG] to different PNCs with constraints matching the VN request.

An example is provided in Figure 4, where the MDSC is requesting to setup a P2P VN between AP1 and AP2. The MDSC can use two different inter-domain links to get from Domain X to Domain Y, namely the one between ASBRX.1 and ASBRY.1 and the one between ASBRX.2 and ASBRY.2, but in order to choose the best end to end path it needs to know what domain X and Y can offer in term of connectivity and constraints between the PE nodes and the ASBR nodes.

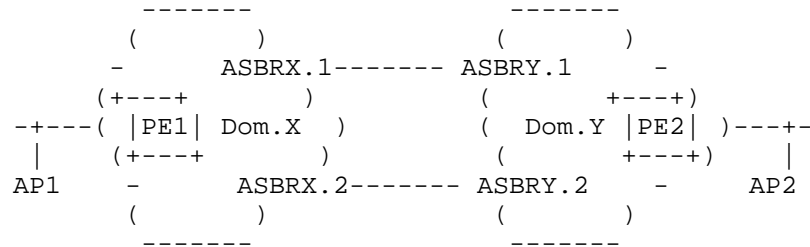


Figure 4: A multi-domain networks example

A path computation request will be issued to PNC.X asking for potential connectivity between PE1 and ASBRX.1 and between PE1 and ASBRX.2 with related objective functions and TE metric constraints. A similar request will be issued to PNC.Y and the results merged together at the MDSC to be able to compute the optimal end-to-end path including the inter domain links.

The info related to the potential connectivity may be cached by the MDSC for subsequent path computation processes or discarded, but in this case the PNCs are not requested to keep the grey topology updated.

#### 4. Protocol/Data Model Requirements

This section provides a set of requirements that may impact the way protocol/data model is designed and the information elements thereof which are carried in the protocol/data model.

It is expected that the abstraction level be negotiated between the CNC and the MDSC (i.e., the CMI) depending on the capability of the CNC. This negotiated level of abstraction on the CMI may also impact the way the MDSC and the PNCs configure and encode the abstracted topology. For example, if the CNC is capable of sophisticated technology specific operation, then this would impact the level of abstraction at the MDSC with the PNCs. On the other hand, if the CNC asks for a generic topology abstraction, then the level of abstraction at the MDSC with the PNCs can be less technology specific than the former case.

The subsequent sections provide a list of possible abstraction levels for various technology domain networks.

##### 4.1. Packet Networks

- For grey abstraction, the type of abstraction and its parameters MUST be defined and configured/negotiated.
  - o Abstraction Level 1: TE-tunnel abstraction for all (S-D) border pairs with:
    - . Maximum B/W available per Priority Level
    - . Minimum Latency
  - o Other Level (TBD)

##### 4.2. OTN Networks

For OTN networks, max bandwidth available may be per ODU 0/1/2/3 switching level or aggregated across all ODU switching levels (i.e., ODUj/k). Clearly, there is a trade-off between these two abstraction methods. Some OTN switches can switch any level of ODUs and in such case there is no need for ODU level abstraction.

- For grey abstraction, the type of abstraction and its parameters MUST be defined and configured/negotiated.
  - o Abstraction Level 1: Per ODU Switching level (i.e., ODU type and number) TE-tunnel abstraction for all (S-D) border pairs with:

- . Maximum B/W available per Priority Level
  - . Minimum Latency
- o Abstraction Level 2: Aggregated TE-tunnel abstraction for all (S-D) border pairs with:
  - . Maximum B/W available per Priority Level
  - . Minimum Latency
- o Other Level (TBD)

#### 4.3. WSON Networks

For WSON networks, max bandwidth available may be per lambda/frequency level (OCh) or aggregated across all lambda/frequency level. Per OCh level abstraction gives more detailed data to the MDSC at the expense of more information processing. Either OCh-level or aggregated level abstraction should factor in the RWA constraint (i.e., wavelength continuity) at the PNC level. This means the PNC should have this capability and advertise it as such. See the Appendix for this abstraction method.

- For grey abstraction, the type of abstraction MUST and its parameters be defined and configured/negotiated.
  - o Abstraction Level 1: Per Lambda/Frequency level TE-tunnel abstraction for all (S-D) border pairs with:
    - . Maximum B/W available per Priority Level
    - . Minimum Latency
  - o Abstraction Level 2: Aggregated TE-tunnel abstraction for all (S-D) border pairs with:
    - . Maximum B/W available per Priority Level
    - . Minimum Latency
  - o Other Level (TBD)

Examples: these examples show how to compute WSON grey topology Abstraction Level 1 and Level 2. These examples illustrate that the encoding of an abstraction topology can be impacted by the configured/negotiated abstraction level in the ACTN interfaces.

This section provides how WSON grey topology abstraction levels 1 and 2 can be computed at a PNC. These examples illustrate that the

encoding of an abstraction topology can be impacted by the configured/negotiated abstraction level at the MPI.

. Abstraction Level 1: Per Lambda/Frequency level TE-tunnel abstraction for all (S-D) border pairs:

For each (S-D) border node pair,

- 1) The concept of a lambda plane: A lambda plane is a confined optical topology with respect to a given lambda value. If an OMS link has the wavelength of the given lambda available, it is included, otherwise excluded.
- 2) Calculate the maximal flow between S and D in every lambda plane. Max flow computation is restricted to each lambda plane is for OCh wavelength continuity.
- 3) Convert each feasible lambda plane with OCh wavelength continuity to B/W equivalent encoding; Send this per lambda level encoding for (S-D) to the MDSC;

. Abstraction Level 2: Aggregated TE-tunnel abstraction for WSON for all (S-D) border pairs

For each (S-D) border node pair,

- 1) The concept of a lambda plane: A lambda plane is a confined optical topology with respect to a given lambda value. If an OMS link has the wavelength of the given lambda available, it is included, otherwise excluded.
- 2) Calculate the maximal flow between S and D in every lambda plane. Max flow computation is restricted to each lambda plane is for OCh wavelength continuity.
- 3) Add up the max flow values across all lambda planes. This is the maximal number of OCh paths that can be setup between S and D at the same time.
- 4) Convert the max number of OCh paths to B/W equivalent encoding; Send this encoding as max B/W for (S-D) to the MDSC;

## 5. Acknowledgements

We thank Adrian Farrel and Italo Busi for providing useful comments and suggestions for this draft.

## 6. References

### 6.1. Informative References

- [RFC7926] A. Farrel, Ed., "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.
- [ACTN-Frame] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, work in progress.
- [TE-Topology] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [PCEP-LS] D. Dhody, Y. Lee and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information," draft-dhodylee-pce-pcep-ls, work in progress.
- [RFC7926] A. Farrel, et. al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.
- [ACTN-YANG] X. Zhang, et. Al., "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-zhang-teas-actn-yang, work in progress

## 7. Contributors

### Contributor's Addresses

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

Xian Zhang  
Huawei

Email: zhang.xian@huawei.com



## Authors' Addresses

Young Lee  
Huawei Technologies  
5340 Legacy Drive  
Plano, TX 75023, USA  
Phone: (469)277-5838

Email: leeyoung@huawei.com

Dhruv Dhody  
Huawei Technologies

Email: dhruv.ietf@gmail.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com



TEAS Working Group  
Internet Draft  
Intended Status: Standard Track  
Expires: November 8, 2019

Y. Lee (Editor)  
Dhruv Dhody  
Satish Karunanithi  
Huawei

Ricard Vilalta  
CTTC

Daniel King  
Lancaster University

Daniele Ceccarelli  
Ericsson

May 8, 2019

YANG models for VN & TE Performance Monitoring Telemetry and Scaling  
Intent Autonomics

draft-lee-teas-actn-pm-telemetry-autonomics-17

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 November 8, 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 provides YANG data models that describe performance monitoring telemetry and scaling intent mechanism for TE-tunnels and Virtual Networks (VN).

The models presented in this draft allow customers to subscribe to and monitor their key performance data of their interest on the level of TE-tunnel or VN. The models also provide customers with the ability to program autonomic scaling intent mechanism on the level of TE-tunnel as well as VN.

## Table of Contents

|                                            |    |
|--------------------------------------------|----|
| 1. Introduction.....                       | 3  |
| 1.1. Terminology.....                      | 4  |
| 1.2. Tree diagram.....                     | 5  |
| 1.3. Prefixes in Data Node Names.....      | 5  |
| 2. Use-Cases.....                          | 5  |
| 3. Design of the Data Models.....          | 7  |
| 3.1. TE KPI Telemetry Model.....           | 7  |
| 3.2. VN KPI Telemetry Model.....           | 8  |
| 4. Autonomic Scaling Intent Mechanism..... | 9  |
| 5. Notification.....                       | 11 |
| 5.1. YANG Push Subscription Examples.....  | 11 |
| 6. YANG Data Tree.....                     | 13 |
| 7. Yang Data Model.....                    | 15 |
| 7.1. ietf-te-kpi-telemetry model.....      | 15 |
| 7.2. ietf-vn-kpi-telemetry model.....      | 21 |
| 8. Security Considerations.....            | 25 |
| 9. IANA Considerations.....                | 26 |

|                                   |    |
|-----------------------------------|----|
| 10. Acknowledgements.....         | 27 |
| 11. References.....               | 27 |
| 11.1. Normative References.....   | 27 |
| 11.2. Informative References..... | 28 |
| 12. Contributors.....             | 29 |
| Authors' Addresses.....           | 29 |

## 1. Introduction

The YANG model discussed in [VN] is used to operate customer-driven Virtual Networks (VNs) during the VN instantiation, VN computation, and its life-cycle service management and operations. YANG model discussed in [TE-Tunnel] is used to operate TE-tunnels during the tunnel instantiation, and its life-cycle management and operations.

The models presented in this draft allow the applications hosted by the customers to subscribe to and monitor their key performance data of their interest on the level of VN [VN] or TE-tunnel [TE-Tunnel]. The key characteristic of the models presented in this document is a top-down programmability that allows the applications hosted by the customers to subscribe to and monitor key performance data of their interest and autonomic scaling intent mechanism on the level of VN as well as TE-tunnel.

According to the classification of [RFC8309], the YANG data models presented in this document can be classified as customer service models, which is mapped to CMI (Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface) of ACTN [RFC8453].

[RFC8233] describes key network performance data to be considered for end-to-end path computation in TE networks. Key performance indicator (KPI) is a term that describes critical performance data that may affect VN/TE-tunnel service. The services provided can be optimized to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers.

This document provides YANG data models generically applicable to any VN/TE-Tunnel service clients to provide an ability to program their customized performance monitoring subscription and publication data models and automatic scaling in/out intent data models. These models can be utilized by a client network controller to initiate these capability to a transport network controller communicating with the client controller via a NETCONF [RFC8341] or a RESTCONF [RFC8040] interface.

The term performance monitoring being used in this document is different from the term that has been used in transport networks for many years. Performance monitoring in this document refers to subscription and publication of streaming telemetry data. Subscription is initiated by the client (e.g., CNC) while publication is provided by the network (e.g., MDSC/PNC) based on the client's subscription. As the scope of performance monitoring in this document is telemetry data on the level of client's VN or TE-tunnel, the entity interfacing the client (e.g., MDSC) has to provide VN or TE-tunnel level information. This would require controller capability to derive VN or TE-tunnel level performance data based on lower-level data collected via PM counters in the Network Elements (NE). How the controller entity derives such customized level data (i.e., VN or TE-tunnel level) is out of the scope of this document.

The data model includes configuration and state data according to the new Network Management Datastore Architecture [RFC8342].

### 1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

**Key Performance Data:** This refers to a set of data the customer is interested in monitoring for their instantiated VNs or TE-tunnels. Key performance data and key performance indicators are interchangeable in this draft.

**Scaling:** This refers to the network ability to re-shape its own resources. Scale out refers to improve network performance by increasing the allocated resources, while scale in refers to decrease the allocated resources, typically because the existing resources are unnecessary.

**Scaling Intent:** To declare scaling conditions, scaling intent is used. Specifically, scaling intent refers to the intent expressed by the client that allows the client to program/configure conditions of their key performance data either for scaling out or scaling in. Various conditions can be set for scaling intent on either VN or TE-tunnel level.

**Network Autonomics:** This refers to the network automation capability that allows client to initiate scaling intent mechanisms and provides the client with the status of the adjusted network

resources based on the client's scaling intent in an automated fashion.

### 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 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   |
|----------|-----------------------|-------------|
| rt       | ietf-routing-types    | [RFC8294]   |
| te       | ietf-te               | [TE-Tunnel] |
| te-types | ietf-te-types         | [TE-Types]  |
| te-tel   | ietf-te-kpi-telemetry | [This I-D]  |
| vn       | ietf-vn               | [VN]        |
| vn-tel   | ietf-vn-kpi-telemetry | [This I-D]  |

Table 1: Prefixes and corresponding YANG modules

## 2. Use-Cases

[PERF] describes use-cases relevant to this draft. It introduces the dynamic creation, modification and optimization of services based on the performance monitoring. Figure 1 shows a high-level workflows for dynamic service control based on traffic monitoring.

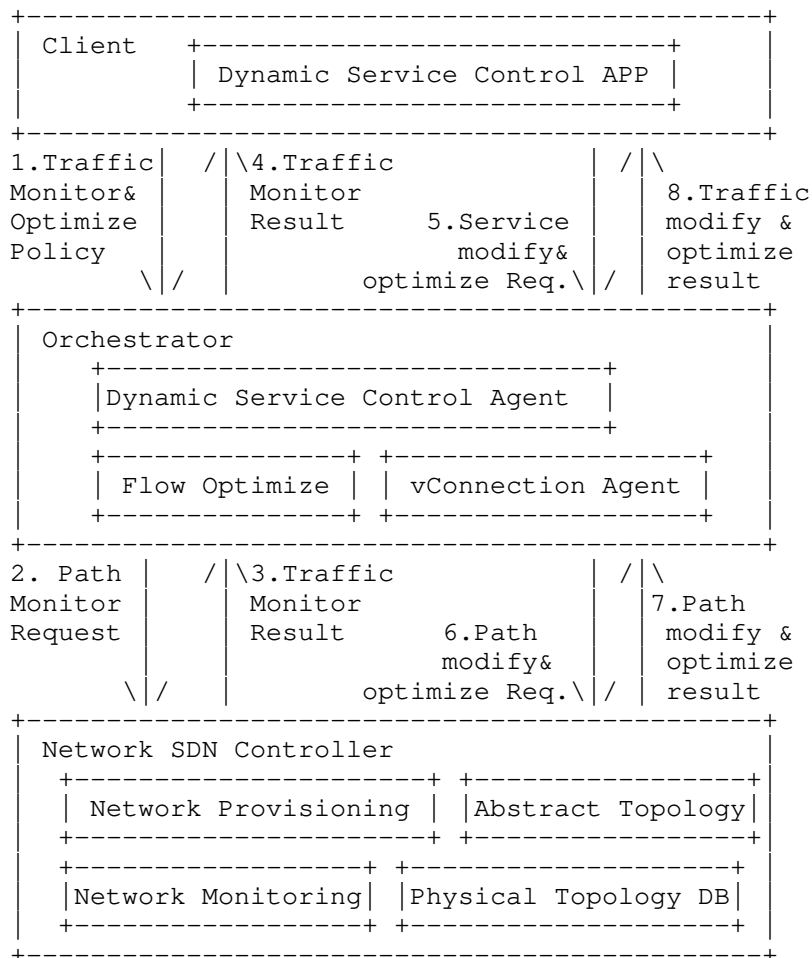


Figure 1 Workflows for dynamic service control based on traffic monitoring

Some of the key points from [PERF] are as follows:

- . Network traffic monitoring is important to facilitate automatic discovery of the imbalance of network traffic, and initiate the network optimization, thus helping the network operator or the virtual network service provider to use the network more efficiently and save the Capital Expense (CAPEX) and the Operating Expense (OPEX).



- . Customer services have various Service Level Agreement (SLA) requirements, such as service availability, latency, latency jitter, packet loss rate, Bit Error Rate (BER), etc. The transport network can satisfy service availability and BER requirements by providing different protection and restoration mechanisms. However, for other performance parameters, there are no such mechanisms. In order to provide high quality services according to customer SLA, one possible solution is to measure the SLA related performance parameters, and dynamically provision and optimize services based on the performance monitoring results.
- . Performance monitoring in a large scale network could generate a huge amount of performance information. Therefore, the appropriate way to deliver the information in the client and network interfaces should be carefully considered.

### 3. Design of the Data Models

The YANG models developed in this document describe two models:

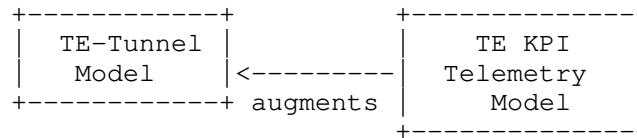
- (i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & 7.1 for details).
- (ii) VN KPI Telemetry Model which provides the VN level of the aggregated performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer (See Section 3.2 & 7.2 for details).

#### 3.1. TE KPI Telemetry Model

This module describes performance telemetry for TE-tunnel model. The telemetry data is augmented to tunnel state. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data (See Section 5 for details)

The TE KPI Telemetry Model augments the TE-Tunnel Model to enhance TE performance monitoring capability. This monitoring capability

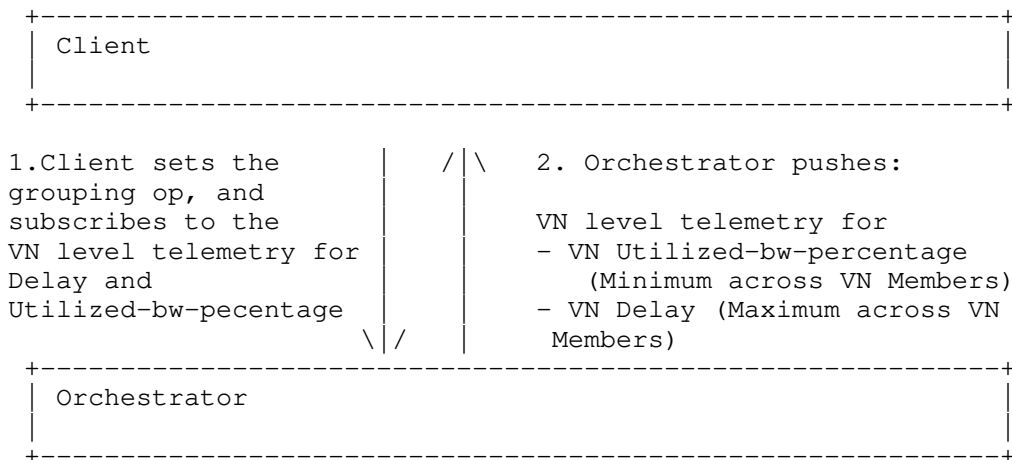
will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.



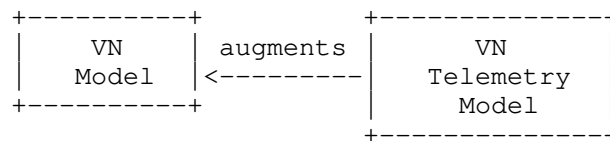
### 3.2. VN KPI Telemetry Model

This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters (See Section 4 for illustrations).

Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters. Grouping operation (such as maximum, mean) could be set at the time of configuration. For example, if maximum grouping operation is used for delay at the VN level, the VN telemetry data is reported as the maximum {delay\_vn\_member\_1, delay\_vn\_member\_2,... delay\_vn\_member\_N}. Thus, this telemetry abstraction mechanism allows the grouping of a certain common set of telemetry values under a grouping operation. This can be done at the VN-member level to suggest how the E2E telemetry be inferred from the per domain tunnel created and monitored by PNCs. One proposed example is the following:



The VN Telemetry Model augments the basic VN model to enhance VN monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of VNs based on the performance monitoring data collected via the VN Telemetry YANG model.



#### 4. Autonomic Scaling Intent Mechanism

Scaling intent configuration mechanism allows the client to configure automatic scale-in and scale-out mechanisms on both the TE-tunnel and the VN level. Various conditions can be set for auto-scaling based on the PM telemetry data.

There are a number of parameters involved in the mechanism:

- . scale-out-intent or scale-in-intent: whether to scale-out or scale-in.
- . performance-type: performance metric type (e.g., one-way-delay, one-way-delay-min, one-way-delay-max, two-way-delay, two-way-delay-min, two-way-delay-max, utilized bandwidth, etc.)

- . threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.
- . scaling-operation-type: in case where scaling condition can be set with one or more performance types, then scaling-operation-type (AND, OR, MIN, MAX, etc.) is applied to these selected performance types and its threshold values.
- . Threshold-time: the duration for which the criteria must hold true.
- . Cooldown-time: the duration after a scaling action has been triggered, for which there will be no further operation.

The following tree is a part of ietf-te-kpi-telemetry tree whose model is presented in full detail in Sections 6 & 7.

```

module: ietf-te-kpi-telemetry
augment /te:te/te:tunnels/te:tunnel:
 +-rw te-scaling-intent
 | +-rw scale-in-intent
 | | +-rw threshold-time? uint32
 | | +-rw cooldown-time? uint32
 | | +-rw scale-in-operation-type? scaling-criteria-operation
 | | +-rw scaling-condition* [performance-type]
 | | +-rw performance-type identityref
 | | +-rw threshold-value? string
 | | +-rw te-telemetry-tunnel-ref?
 | | -> /te:te/tunnels/tunnel/name
 | +-rw scale-out-intent
 | | +-rw threshold-time? uint32
 | | +-rw cooldown-time? uint32
 | | +-rw scale-out-operation-type? scaling-criteria-operation
 | | +-rw scaling-condition* [performance-type]
 | | +-rw performance-type identityref
 | | +-rw threshold-value? string
 | | +-rw te-telemetry-tunnel-ref?
 | | -> /te:te/tunnels/tunnel/name

```

Let say the client wants to set the scaling out operation based on two performance-types (e.g., two-way-delay and utilized-bandwidth for a te-tunnel), it can be done as follows:

- . Set Threshold-time: x (sec) (duration for which the criteria must hold true)

- . Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)
- . Set AND for the scale-out-operation-type

In the scaling condition's list, the following two components can be set:

List 1: Scaling Condition for Two-way-delay

- . performance type: Two-way-delay
- . threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- . performance type: Utilized bandwidth
- . threshold-value: w megabytes

## 5. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [YANG-PUSH] and [Event-Notification] can be used. This mechanism currently allows the user to:

- . Subscribe to 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.

### 5.1. YANG Push Subscription Examples

[YANG-PUSH] allows subscriber applications to request a continuous, customized stream of updates from a YANG datastore.

Below example shows the way for a client to subscribe to the telemetry information for a particular tunnel (Tunnel1). The telemetry parameter that the client is interested in is one-way-delay.

```

<netconf:rpc netconf:message-id="101"
 xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
 <establish-subscription
 xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
 <filter netconf:type="subtree">
 <te xmlns="urn:ietf:params:xml:ns:yang:ietf-te">
 <tunnels>
 <tunnel>
 <name>Tunnel1</name>
 <identifier/>
 <state>
 <te-telemetry xmlns="urn:ietf:params:xml:ns:yang:
 ietf-te-kpi-telemetry">
 <one-way-delay/>
 </te-telemetry>
 </state>
 </tunnel>
 </tunnels>
 </te>
 </filter>
 <period>500</period>
 <encoding>encode-xml</encoding>
 </establish-subscription>
</netconf:rpc>

```

This example shows the way for a client to subscribe to the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way-delay and one-way-utilized-bandwidth.

```

<netconf:rpc netconf:message-id="101"
 xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
 <establish-subscription
 xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
 <filter netconf:type="subtree">
 <vn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-vn">
 <vn>
 <vn-list>
 <vn-id/>
 <vn-name/>
 <vn-telemetry xmlns="urn:ietf:params:xml:ns:yang:
 ietf-vn-kpi-telemetry">
 <one-way-delay/>
 <one-way-utilized-bandwidth/>
 </vn-telemetry >
 </vn-list>
 </vn>
 </vn-state>
 </filter>
 <period>500</period>
 </establish-subscription>
</netconf:rpc>

```

## 6. YANG Data Tree

```

module: ietf-te-kpi-telemetry
augment /te:te/te:tunnels/te:tunnel:
 +--rw te-scaling-intent
 | +--rw scale-in-intent
 | | +--rw threshold-time? uint32
 | | +--rw cooldown-time? uint32
 | | +--rw scale-in-operation-type? scaling-criteria-operation
 | | +--rw scaling-condition* [performance-type]
 | | | +--rw performance-type identityref
 | | | +--rw threshold-value? string
 | | | +--rw te-telemetry-tunnel-ref?
 | | | | -> /te:te/tunnels/tunnel/name
 | | +--rw scale-out-intent
 | | | +--rw threshold-time? uint32
 | | | +--rw cooldown-time? uint32
 | | | +--rw scale-out-operation-type? scaling-criteria-operation
 | | | +--rw scaling-condition* [performance-type]
 | | | | +--rw performance-type identityref
 | | | | +--rw threshold-value? string
 | | | | +--rw te-telemetry-tunnel-ref?
 | | | | | -> /te:te/tunnels/tunnel/name
 | +--ro te-telemetry
 | | +--ro id? string
 | | +--ro performance-metrics-one-way
 | | | +--ro one-way-delay? uint32
 | | | +--ro one-way-delay-normality?
 | | | | te-types:performance-metrics-normality
 | | | +--ro one-way-residual-bandwidth?
 | | | | rt-types:bandwidth-ieee-float32
 | | | +--ro one-way-residual-bandwidth-normality?
 | | | | te-types:performance-metrics-normality
 | | | +--ro one-way-available-bandwidth?
 | | | | rt-types:bandwidth-ieee-float32
 | | | +--ro one-way-available-bandwidth-normality?
 | | | | te-types:performance-metrics-normality
 | | | +--ro one-way-utilized-bandwidth?
 | | | | rt-types:bandwidth-ieee-float32
 | | | +--ro one-way-utilized-bandwidth-normality?
 | | | | te-types:performance-metrics-normality
 | | +--ro performance-metrics-two-way
 | | | +--ro two-way-delay? uint32
 | | | +--ro two-way-delay-normality?
 | | | | te-types:performance-metrics-normality
 | | +--ro te-ref?
 | | | -> /te:te/tunnels/tunnel/name
 +--ro vn-list

```

```

module: ietf-vn-kpi-telemetry
augment /vn:vn/vn:vn-list:

```

```

+--rw vn-scaling-intent
| +--rw scale-in-intent
| | +--rw threshold-time? uint32
| | +--rw cooldown-time? uint32
| | +--rw scale-in-operation-type? scaling-criteria-operation
| | +--rw scaling-condition* [performance-type]
| | | +--rw performance-type identityref
| | | +--rw threshold-value? string
| | | +--rw te-telemetry-tunnel-ref?
| | | | -> /te:te/tunnels/tunnel/name
| | +--rw scale-out-intent
| | | +--rw threshold-time? uint32
| | | +--rw cooldown-time? uint32
| | | +--rw scale-out-operation-type? scaling-criteria-operation
| | | +--rw scaling-condition* [performance-type]
| | | | +--rw performance-type identityref
| | | | +--rw threshold-value? string
| | | | +--rw te-telemetry-tunnel-ref?
| | | | | -> /te:te/tunnels/tunnel/name
| +--ro vn-telemetry
| | +--ro performance-metrics-one-way
| | | +--ro one-way-delay? uint32
| | | +--ro one-way-delay-normality?
| | | | te-types:performance-metrics-normality
| | | +--ro one-way-residual-bandwidth?
| | | | rt-types:bandwidth-ieee-float32
| | | +--ro one-way-residual-bandwidth-normality?
| | | | te-types:performance-metrics-normality
| | | +--ro one-way-available-bandwidth?
| | | | rt-types:bandwidth-ieee-float32
| | | +--ro one-way-available-bandwidth-normality?
| | | | te-types:performance-metrics-normality
| | | +--ro one-way-utilized-bandwidth?
| | | | rt-types:bandwidth-ieee-float32
| | | +--ro one-way-utilized-bandwidth-normality?
| | | | te-types:performance-metrics-normality
| | +--ro performance-metrics-two-way
| | | +--ro two-way-delay? uint32
| | | +--ro two-way-delay-normality?
| | | | te-types:performance-metrics-normality
| | +--ro grouping-operation? grouping-operation
augment /vn:vn/vn:vn-list/vn:vn-member-list:
+--ro vn-member-telemetry
| +--ro performance-metrics-one-way
| | +--ro one-way-delay? uint32
| | +--ro one-way-delay-normality?
| | | te-types:performance-metrics-normality
| | +--ro one-way-residual-bandwidth?
| | | rt-types:bandwidth-ieee-float32
| | +--ro one-way-residual-bandwidth-normality?
| | | te-types:performance-metrics-normality
| | +--ro one-way-available-bandwidth?
| | | rt-types:bandwidth-ieee-float32
| | +--ro one-way-available-bandwidth-normality?
| | | te-types:performance-metrics-normality

```



```

 | +--ro one-way-utilized-bandwidth?
 | | rt-types:bandwidth-ieee-float32
 | +--ro one-way-utilized-bandwidth-normality?
 | te-types:performance-metrics-normality
+--ro performance-metrics-two-way
 | +--ro two-way-delay? uint32
 | +--ro two-way-delay-normality?
 | te-types:performance-metrics-normality
+--ro te-grouped-params*
 | -> /te:te/tunnels/tunnel/te-kpi:te-telemetry/id
+--ro grouping-operation? grouping-operation

```

## 7. Yang Data Model

### 7.1. ietf-te-kpi-telemetry model

The YANG code is as follows:

<CODE BEGINS> file "ietf-te-kpi-telemetry@2019-04-18.yang"

```

module ietf-te-kpi-telemetry {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry";
 prefix te-tel;

 import ietf-te {
 prefix te;
 reference
 "RFC YYYY: A YANG Data Model for Traffic Engineering
 Tunnels and Interfaces";
 }

 /* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-ietf-teas-yang-te
 becomes an RFC.*/

 import ietf-te-types {
 prefix te-types;
 reference
 "RFC YYYY: Traffic Engineering Common YANG Types";
 }

 /* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-ietf-teas-yang-te-types
 becomes an RFC.*/

```

```
organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";
contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
 Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
 Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";
description
 "This module describes YANG data model for performance
 monitoring telemetry for te tunnels.

 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.";

/* Note: The RFC Editor will replace XXXX with the number
 assigned to the RFC once draft-lee-teas-pm-telemetry-
 autonomics becomes an RFC.*/

revision 2019-04-18 {
 description
 "Initial revision. This YANG file defines
 a YANG model for TE telemetry.";
 reference "Derived from earlier versions of base YANG files";
}

identity telemetry-param-type {
 description
 "Base identity for telemetry param types";
}

identity one-way-delay {
 base telemetry-param-type;
 description
 "To specify average Delay in one (forward)
 direction";
```

```
reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

identity two-way-delay {
 base telemetry-param-type;
 description
 "To specify average Delay in both (forward and reverse)
 directions";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

identity one-way-delay-variation {
 base telemetry-param-type;
 description
 "To specify average Delay Variation in one (forward) direction";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

identity two-way-delay-variation {
 base telemetry-param-type;
 description
 "To specify average Delay Variation in both (forward and reverse)
 directions";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

identity utilized-bandwidth {
```

```
 base telemetry-param-type;
 description
 "To specify utilized bandwidth over the specified source
 and destination.";
 reference
 "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
 RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
 RFC7823: Performance-Based Path Selection for Explicitly
 Routed Label Switched Paths (LSPs) Using TE Metric
 Extensions";
}

identity utilized-percentage {
 base telemetry-param-type;
 description
 "To specify utilization percentage of the entity
 (e.g., tunnel, link, etc.)";
}

typedef scaling-criteria-operation {
 type enumeration {
 enum AND {
 description
 "AND operation";
 }
 enum OR {
 description
 "OR operation";
 }
 }
 description
 "Operations to analyze list of scaling criterias";
}

grouping scaling-duration {
 description
 "Base scaling criteria durations";
 leaf threshold-time {
 type uint32;
 units "seconds";
 description
 "The duration for which the criteria must hold true";
 }
 leaf cooldown-time {
 type uint32;
 units "seconds";
 description
```

```
 "The duration after a scaling-in/scaling-out action has been
 triggered, for which there will be no further operation";
 }
}

grouping scaling-criteria {
 description
 "Grouping for scaling criteria";
 leaf performance-type {
 type identityref {
 base telemetry-param-type;
 }
 description
 "Reference to the tunnel level telemetry type";
 }
 leaf threshold-value {
 type string;
 description
 "Scaling threshold for the telemetry parameter type";
 }
 leaf te-telemetry-tunnel-ref {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te:name";
 }
 description
 "Reference to tunnel";
 }
}

grouping scaling-in-intent {
 description
 "Basic scaling in intent";
 uses scaling-duration;
 leaf scale-in-operation-type {
 type scaling-criteria-operation;
 default "AND";
 description
 "Operation to be applied to check between
 scaling criterias to check if the scale in
 threshold condition has been met.
 Defaults to AND";
 }
 list scaling-condition {
 key "performance-type";
 description
 "Scaling conditions";
 uses scaling-criteria;
 }
}
```

```
 }
 }

 grouping scaling-out-intent {
 description
 "Basic scaling out intent";
 uses scaling-duration;
 leaf scale-out-operation-type {
 type scaling-criteria-operation;
 default "OR";
 description
 "Operation to be applied to check between
 scaling criterias to check if the scale out
 threshold condition has been met.
 Defaults to OR";
 }
 list scaling-condition {
 key "performance-type";
 description
 "Scaling conditions";
 uses scaling-criteria;
 }
 }

 augment "/te:te/te:tunnels/te:tunnel" {
 description
 "Augmentation parameters for config scaling-criteria
 TE tunnel topologies. Scale in/out criteria might be used
 for network autonomics in order the controller
 to react to a certain set of monitored params.";
 container te-scaling-intent {
 description
 "scaling intent";
 container scale-in-intent {
 description
 "scale-in";
 uses scaling-in-intent;
 }
 container scale-out-intent {
 description
 "scale-out";
 uses scaling-out-intent;
 }
 }
 container te-telemetry {
 config false;
 description
```

```

 "telemetry params";
 leaf id {
 type string;
 description
 "Id of telemetry param";
 }
 uses te-types:performance-metrics-attributes;
 leaf te-ref {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te:name";
 }
 description
 "Reference to measured te tunnel";
 }
}
}
}
<CODE ENDS>

```

## 7.2. ietf-vn-kpi-telemetry model

The YANG code is as follows:

```

<CODE BEGINS> file "ietf-vn-kpi-telemetry@2019-04-18.yang"

module ietf-vn-kpi-telemetry {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry";
 prefix vn-tel;

 import ietf-vn {
 prefix vn;
 reference
 "RFC YYYY: A YANG Data Model for VN Operation";
 }

 /* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-ietf-teas-actn-vn-yang
 becomes an RFC.*/

 import ietf-te {
 prefix te;
 reference
 "RFC YYYY: A YANG Data Model for Traffic Engineering

```

```
 Tunnels and Interfaces";
}

/* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-ietf-teas-yang-te
 becomes an RFC.*/

import ietf-te-types {
 prefix te-types;
 reference
 "RFC YYYY: Traffic Engineering Common YANG Types";
}

/* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-ietf-teas-yang-te-types
 becomes an RFC.*/

import ietf-te-kpi-telemetry {
 prefix te-kpi;
 reference
 "RFC YYYY: YANG models for VN & TE Performance Monitoring
 Telemetry and Scaling Intent Autonomics";
}

/* Note: The RFC Editor will replace YYYY with the number
 assigned to the RFC once draft-lee-teas-actn-pm-telemetry
 -autonomics becomes an RFC.*/

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";
contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
 Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
 Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";

description
 "This module describes YANG data models for performance
 monitoring telemetry for vn.

 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.";

/\* Note: The RFC Editor will replace XXXX with the number assigned to the RFC once draft-lee-teas-pm-telemetry-autonomics becomes an RFC.\*/

```
revision 2019-04-18 {
 description
 "Initial revision. This YANG file defines
 the VN telemetry.";
 reference "Derived from earlier versions of base YANG files";
}

typedef grouping-operation {
 type enumeration {
 enum MINIMUM {
 description
 "Select the minimum param";
 }
 enum MAXIMUM {
 description
 "Select the maximum param";
 }
 enum MEAN {
 description
 "Select the MEAN of the params";
 }
 enum STD_DEV {
 description
 "Select the standard deviation of the
 monitored params";
 }
 enum AND {
 description
 "Select the AND of the params";
 }
 enum OR {
 description
 "Select the OR of the params";
 }
 }
 description
```

```
 "Operations to analyze list of monitored params";
}

grouping vn-telemetry-param {
 description
 "augment of te-kpi:telemetry-param for VN specific params";
 leaf-list te-grouped-params {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te-kpi:te-telemetry/te-kpi:id";
 }
 description
 "Allows the definition of a vn-telemetry param
 as a grouping of underlying TE params";
 }
 leaf grouping-operation {
 type grouping-operation;
 description
 "describes the operation to apply to
 te-grouped-params";
 }
}

augment "/vn:vn/vn:vn-list" {
 description
 "Augmentation parameters for state TE VN topologies.";
 container vn-scaling-intent {
 description
 "scaling intent";
 container scale-in-intent {
 description
 "VN scale-in";
 uses te-kpi:scaling-in-intent;
 }
 container scale-out-intent {
 description
 "VN scale-out";
 uses te-kpi:scaling-out-intent;
 }
 }
 container vn-telemetry {
 config false;
 description
 "VN telemetry params";
 uses te-types:performance-metrics-attributes;
 leaf grouping-operation {
 type grouping-operation;
 description
```

```
 "describes the operation to apply to the VN-members";
 }
}
}
augment "/vn:vn/vn:vn-list/vn:vn-member-list" {
 description
 "Augmentation parameters for state TE vn member topologies.";
 container vn-member-telemetry {
 config false;
 description
 "VN member telemetry params";
 uses te-types:performance-metrics-attributes;
 uses vn-telemetry-param;
 }
}
}
<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 NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. 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 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.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true"). These data nodes may be considered sensitive or vulnerable in some network environments.

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:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent
/te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent

/vn:vn/vn:vn-list/vn-scaling-intent/scale-in-intent
/vn:vn/vn:vn-list/vn-scaling-intent/scale-out-intent
```

## 9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```

URI: urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry
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 [RFC7950]:

```

name: ietf-te-kpi-telemetry
namespace: urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry
prefix: te-tel
reference: RFC XXXX (TDB)

```

```


name: ietf-vn-kpi-telemetry
namespace: urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry
prefix: vn-tel
reference: RFC XXXX (TDB)

```

## 10. Acknowledgements

We thank Rakesh Gandhi, Tarek Saad and Igor Bryskin for useful discussions and their suggestions for this work.

## 11. References

### 11.1. Normative References

- [RFC6242] M. Wasserman, "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.
- [RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", August 2016.
- [RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8233] D. Dhody, et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", RFC 8233, September 2017.
- [RFC8341] A. Bierman, M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.
- [RFC8342] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

- [RFC8446] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", RFC8446, August 2018.
- [TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [TE-Types] T. Saad, et.al, "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types, work in progress.
- [VN] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

## 11.2. Informative References

- [RFC3688] M. Mealling, "The IETF XML Registry", RFC 3688, January 2004.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.
- [RFC7823] Atlas, A., Drake, J., Giacalone, S., and S. Previdi, "Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions", RFC 7823, May 2016.
- [RFC8294] X. Liu, et al, "Routing Area Common YANG Data Types", RFC 8294, December 2017.
- [RFC8340] M. Bjorklund and L. Berger (Editors), "YANG Tree Diagrams", RFC 8340, March 2018.
- [YANG-PUSH] A. Clemm, et al, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push, work in progress.
- [Event-Notification] E. Voit, et al, "Dynamic subscription to YANG Events and Datastores over NETCONF", draft-ietf-netconf-netconf-event-notifications, work in progress.

- [PERF] Y. XU, et al., "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control, work in progress.
- [RFC8309] Q. Wu, W. Cheng, and A. Farrel. "Service Models Explained", RFC 8309, January 2018.
- [RFC8453] D. Ceccarelli and Y. Lee (Editors), "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.
- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, March 2019.

## 12. Contributors

### Authors' Addresses

Young Lee  
Huawei Technologies  
5340 Legacy Drive Suite 173  
Plano, TX 75024, USA

Email: leeyoung@huawei.com

Dhruv Dhody  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
India

Email: dhruv.dhody@huawei.com

Satish Karunanithi  
Huawei Technology  
Leela Palace  
Bangalore, Karnataka 560008  
India

Email: satish.karunanithi@gmail.com

Ricard Vilalta  
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)  
Av. Carl Friedrich Gauss 7  
08860 - Castelldefels  
Barcelona (Spain)  
Email: ricard.vilalta@cttc.es

Daniel King  
Lancaster University  
  
Email: d.king@lancaster.ac.uk

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden  
  
Email: daniele.ceccarelli@ericsson.com





TEAS Working Group  
Internet Draft  
Intended Status: Standard Track  
Expires: November 29, 2018

Y. Lee (Editor)  
Dhruv Dhody  
Huawei  
D. Ceccarelli  
Ericsson  
Igor Bryskin  
Huawei  
Bin Yeong Yoon  
ETRI

May 29, 2018

## A Yang Data Model for ACTN VN Operation

draft-lee-teas-actn-vn-yang-13

### Abstract

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network Service (VNS) operation.

### 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 November 29, 2018.

Copyright Notice  
Lee, et al.

Expires November 2018

[Page 1]

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  |
| 1.1. Terminology.....                                 | 4  |
| 2. ACTN CMI context.....                              | 4  |
| 2.1. Type 1 VN.....                                   | 4  |
| 2.2. Type 2 VN.....                                   | 5  |
| 3. High-Level Control Flows with Examples.....        | 6  |
| 3.1. Type 1 VN Illustration.....                      | 6  |
| 3.2. Type 2 VN Illustration.....                      | 8  |
| 4. Justification of the ACTN VN Model on the CMI..... | 10 |
| 4.1. Customer view of VN.....                         | 10 |
| 4.2. Innovative Services.....                         | 10 |
| 4.2.1. VN Compute.....                                | 10 |
| 4.2.2. Multi-sources and Multi-destinations.....      | 11 |
| 4.2.3. Others.....                                    | 11 |
| 4.3. Summary.....                                     | 12 |
| 5. ACTN VN YANG Model (Tree Structure).....           | 12 |
| 6. ACTN-VN YANG Code.....                             | 15 |
| 7. JSON Example.....                                  | 27 |
| 7.1. ACTN VN JSON.....                                | 28 |
| 7.2. TE-topology JSON.....                            | 33 |
| 8. Security Considerations.....                       | 49 |
| 9. IANA Considerations.....                           | 50 |
| 10. Acknowledgments.....                              | 50 |
| 11. References.....                                   | 51 |
| 11.1. Normative References.....                       | 51 |
| 11.2. Informative References.....                     | 51 |
| 12. Contributors.....                                 | 52 |

|                         |    |
|-------------------------|----|
| Authors' Addresses..... | 52 |
|-------------------------|----|

## 1. Introduction

This document provides a YANG data model for the Abstraction and Control of Traffic Engineered (TE) networks (ACTN) Virtual Network Service (VNS) operation that is going to be implemented for the Customer Network Controller (CNC)- Multi-Domain Service Coordinator (MSDC) interface (CMI).

The YANG model on the CMI is also known as customer service model in [Service-YANG]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN computation, VN instantiation and its life-cycle management and operations.

The YANG model discussed in this document basically provides the following:

- o Characteristics of Access Points (APs) that describe customer's end point characteristics;
- o Characteristics of Virtual Network Access Points (VNAP) that describe How an AP is partitioned for multiple VNs sharing the AP and its reference to a Link Termination Point (LTP) of the Provider Edge (PE) Node;
- o Characteristics of Virtual Networks (VNs) that describe the customer's VNs in terms of VN Members comprising a VN, multi-source and/or multi-destination characteristics of VN Member, the VN's reference to TE-topology's Abstract Node;

The actual VN instantiation is performed with Connectivity Matrices sub-module of TE-Topology Model [TE-Topo] which interacts with the VN YANG module presented in this draft. Once TE-topology Model is used in triggering VN instantiation over the networks, TE-tunnel [TE-tunnel] Model will inevitably interact with TE-Topology model for setting up actual tunnels and LSPs under the tunnels.

The ACTN VN operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture (NMDA) [NMDA]. The origin of the data is indicated as per the origin metadata annotation.

### 1.1. Terminology

Refer to [ACTN-Frame] and [RFC7926] for the key terms used in this document.

### 2. ACTN CMI context

The model presented in this document has the following ACTN context.

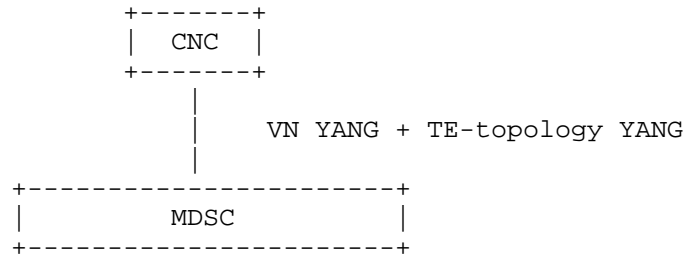


Figure 1. ACTN CMI

Both ACTN VN YANG and TE-topology models are used over the CMI to establish a VN over TE networks.

#### 2.1. Type 1 VN

As defined in [ACTN-FW], a Virtual Network is a customer view of the TE network. To recapitulate VN types from [ACTN-FW], Type 1 VN is defined as follows:

The VN can be seen as a set of edge-to-edge links (a Type 1 VN). Each link is referred to as a VN member and is formed as an end-to-end tunnel across the underlying networks. Such tunnels may be constructed by recursive slicing or abstraction of paths in the underlying networks and can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.

If we were to create a VN where we have four VN-members as follows:

|             |       |
|-------------|-------|
| VN-Member 1 | L1-L4 |
| VN-Member 2 | L1-L7 |
| VN-Member 3 | L2-L4 |
| VN-Member 4 | L3-L8 |

Where L1, L2, L3, L4, L7 and L8 correspond to a Customer End-Point, respectively.

This VN can be modeled as one abstract node representation as follows in Figure 2:

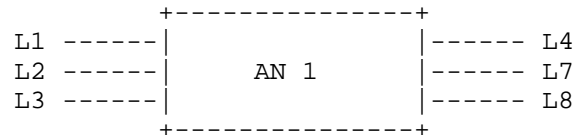


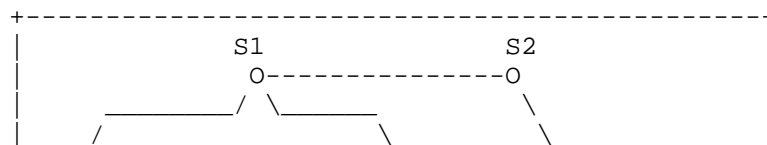
Figure 2. Abstract Node (One node topology)

Modeling a VN as one abstract node is the easiest way for customers to express their end-to-end connectivity; however, customers are not limited to express their VN only with one abstract node. In some cases, more than one abstract nodes can be employed to express their VN.

## 2.2. Type 2 VN

For some VN members of a VN, the customers are allowed to configure the actual path (i.e., detailed virtual nodes and virtual links) over the VN/abstract topology agreed mutually between CNC and MDSC prior to or a topology created by the MDSC as part of VN instantiation. Type 2 VN is always built on top of a Type 1 VN.

If a Type 2 VN is desired for some or all of VN members of a type 1 VN (see the example in Section 2.1), the TE-topology model can provide the following abstract topology (that consists of virtual nodes and virtual links) which is built on top of the Type 1 VN.



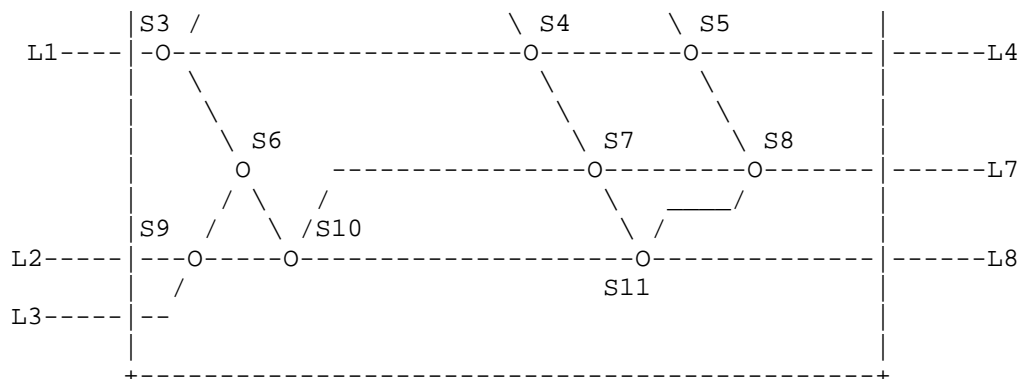


Figure 3. Type 2 topology

As you see from Figure 3, the Type 1 abstract node is depicted as a Type 1 abstract topology comprising of detailed virtual nodes and virtual links.

As an example, if VN-member 1 (L1-L4) is chosen to configure its own path over Type 2 topology, it can select, say, a path that consists of the ERO {S3,S4,S5} based on the topology and its service requirement. This capability is enacted via TE-topology configuration by the customer.

### 3. High-Level Control Flows with Examples

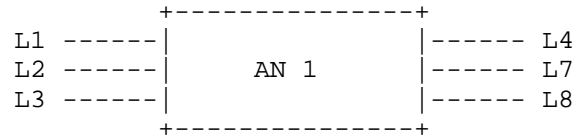
#### 3.1. Type 1 VN Illustration

If we were to create a VN where we have four VN-members as follows:

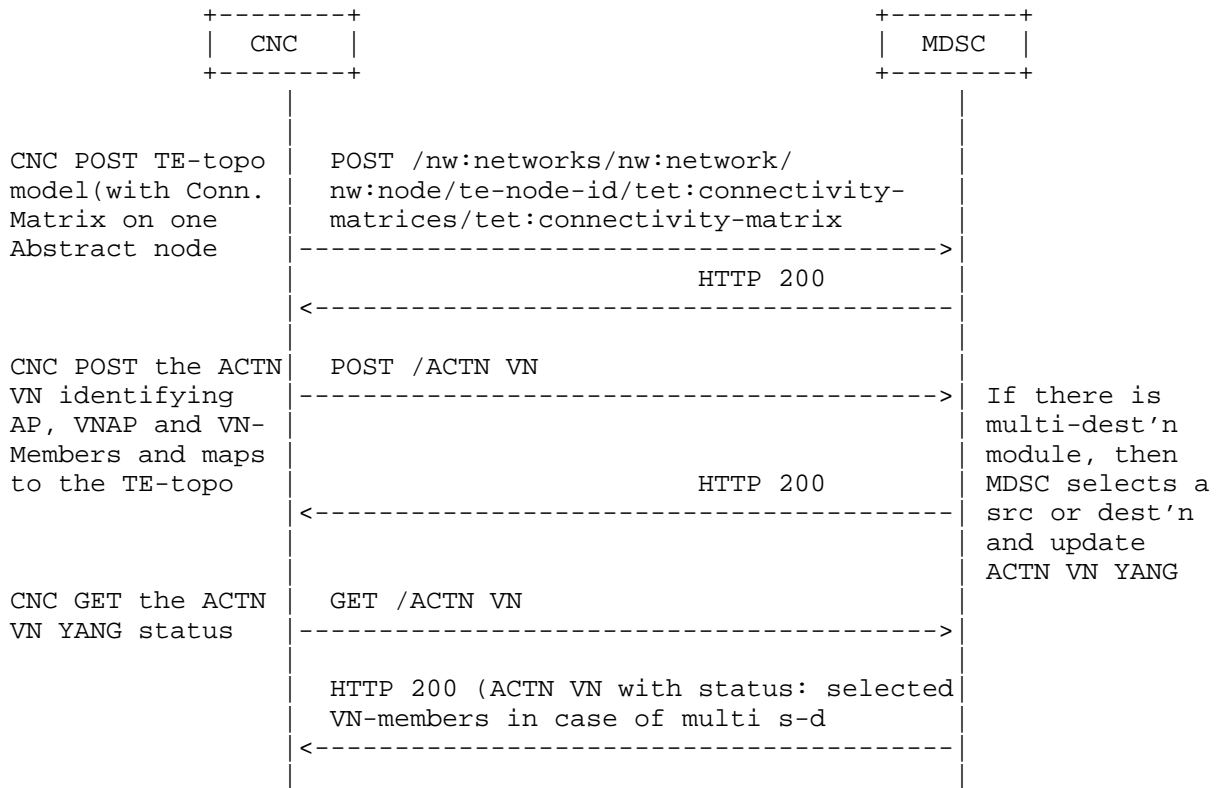
|             |       |
|-------------|-------|
| VN-Member 1 | L1-L4 |
| VN-Member 2 | L1-L7 |
| VN-Member 3 | L2-L4 |
| VN-Member 4 | L3-L8 |

Where L1, L2, L3, L4, L7 and L8 correspond to Customer End-Point, respectively.

This VN can be modeled as one abstract node representation as follows:



If this VN is Type 1, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using ACTN VN and TE-Topology Model.



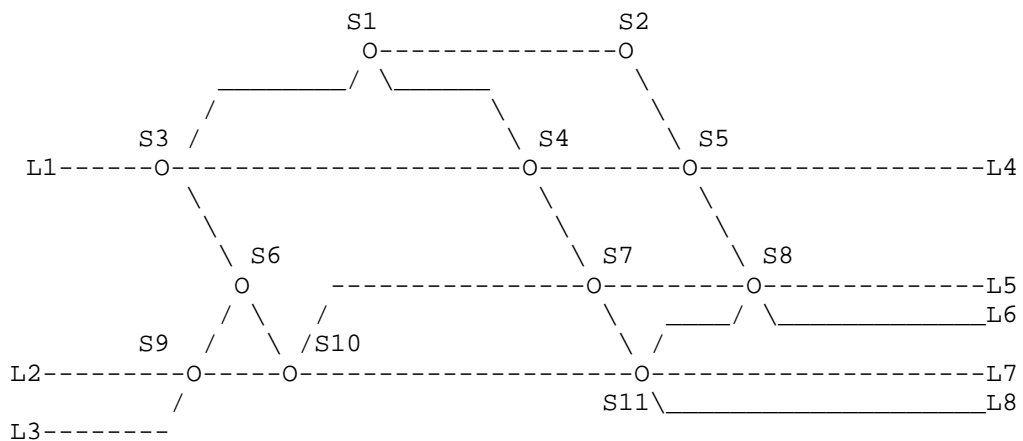


### 3.2. Type 2 VN Illustration

For some VN members, the customer may want to "configure" explicit routes over the path that connects its two end-points. Let us consider the following example.

|             |                       |
|-------------|-----------------------|
| VN-Member 1 | L1-L4                 |
| VN-Member 2 | L1-L7 (via S4 and S7) |
| VN-Member 3 | L2-L4                 |
| VN-Member 4 | L3-L8 (via S10)       |

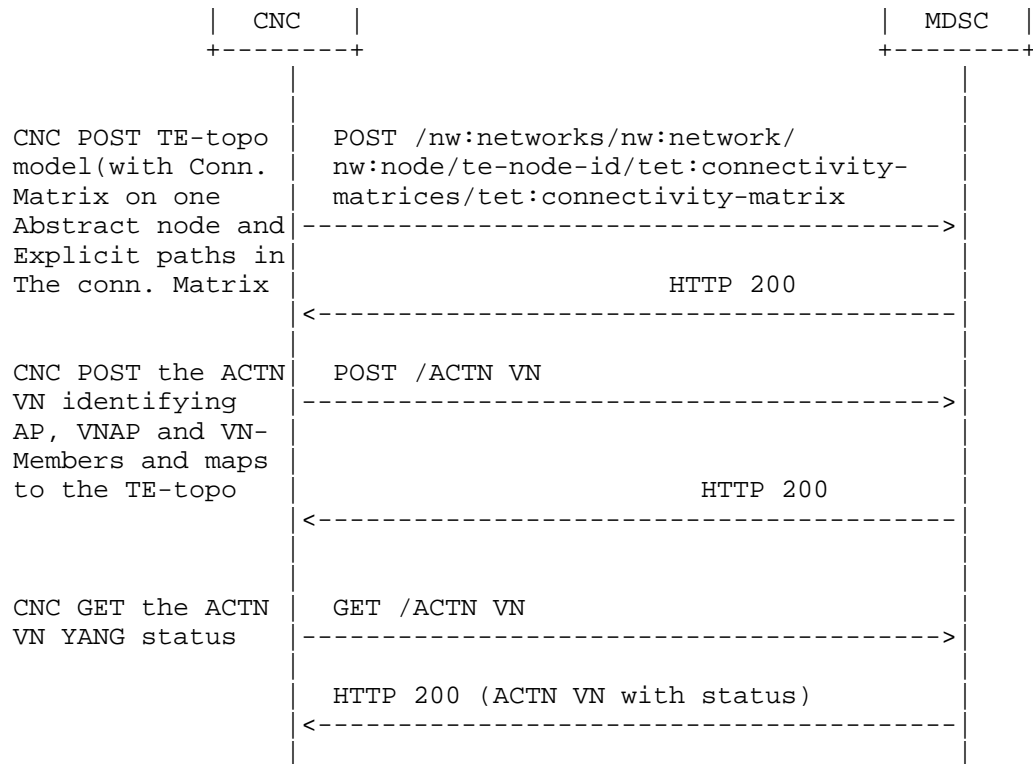
Where the following topology is the underlay for Abstraction Node 1 (AN1).



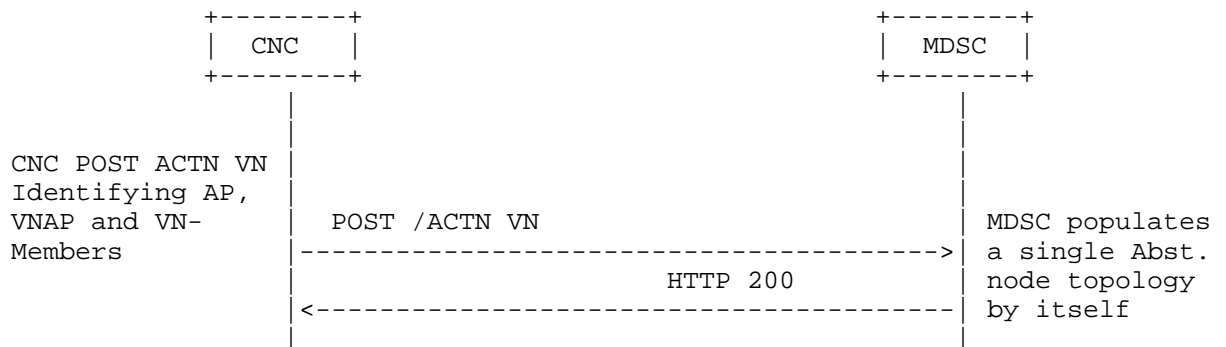
If CNC creates the single abstract topology, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using ACTN VN and TE-Topology Model.

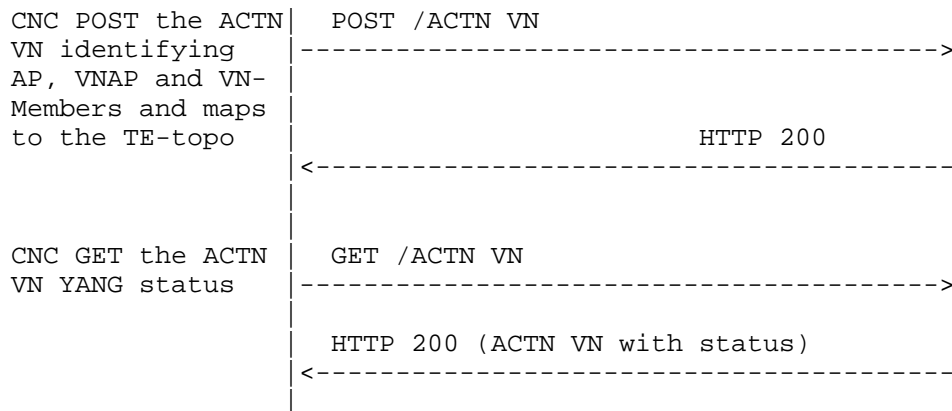
+-----+

+-----+



On the other hand, if MDSC create single node topology based ACTN VN YANG posted by the CNC, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using ACTN VN and TE-Topology Model.





#### 4. Justification of the ACTN VN Model on the CMI.

##### 4.1. Customer view of VN

The VN-Yang model allows to define a customer view, and allows the customer to communicate using the VN constructs as described in the [ACTN-INFO]. It also allows to group the set of edge-to-edge links (i.e., VN members) under a common umbrella of VN. This allows the customer to instantiate and view the VN as one entity, making it easier for some customers to work on VN without worrying about the details of the provider based YANG models.

This is similar to the benefits of having a separate YANG model for the customer services as described in [SERVICE-YANG], which states that service models do not make any assumption of how a service is actually engineered and delivered for a customer.

##### 4.2. Innovative Services

###### 4.2.1. VN Compute

ACTN VN supports VN compute (pre-instantiation mode) to view the full VN as a single entity before instantiation. Achieving this via path computation or "compute only" tunnel setup does not provide the same functionality.

#### 4.2.2. Multi-sources and Multi-destinations

In creating a virtual network, the list of sources or destinations or both may not be pre-determined by the customer. For instance, for a given source, there may be a list of multiple-destinations to which the optimal destination may be chosen depending on the network resource situations. Likewise, for a given destination, there may also be multiple-sources from which the optimal source may be chosen. In some cases, there may be a pool of multiple sources and destinations from which the optimal source-destination may be chosen. The following YANG module is shown for describing source container and destination container. The following YANG tree shows how to model multi-sources and multi-destinations.

```

+--rw actn
 . . .
 +--rw vn
 +--rw vn-list* [vn-id]
 +--rw vn-id uint32
 +--rw vn-name? string
 +--rw vn-topology-id? te-types:te-topology-id
 +--rw abstract-node? -> /nw:networks/network/node/tet:te-node-id
 +--rw vn-member-list* [vn-member-id]
 | +--rw vn-member-id uint32
 | +--rw src
 | | +--rw src? -> /actn/ap/access-point-list/access-po
int-id | | +--rw src-vn-ap-id? -> /actn/ap/access-point-list/vn-ap/vn-
ap-id | | +--rw multi-src? boolean {multi-src-dest}?
 | | +--rw dest
 | | +--rw dest? -> /actn/ap/access-point-list/access-p
oint- | | +--rw dest-vn-ap-id? -> /actn/ap/access-point-list/vn-ap/vn
id | | +--rw multi-dest? boolean {multi-src-dest}?
 | | +--rw connetivity-matrix-id? -> /nw:networks/network/node/tet:
te/te- | +--ro oper-status? identityref
node-attributes/connectivity-matrices/connectivity-matrix/id
 | +--ro if-selected? boolean {multi-src-dest}?
 +--rw admin-status? identityref
 +--ro oper-status? identityref

```

#### 4.2.3. Others

The VN Yang model can be easily augmented to support the mapping of VN to the Services such as L3SM and L2SM as described in [TE-MAP].

The VN Yang model can be extended to support telemetry, performance monitoring and network autonomies as described in [ACTN-PM].

#### 4.3. Summary

This section summarizes the innovative service features of the ACTN VN Yang.

- o Maintenance of AP and VNAP along with VN.
- o VN construct to group of edge-to-edge links
- o VN Compute (pre-instantiate)
- o Multi-Source / Multi-Destination
- o Ability to support various VN and VNS Types
  - \* VN Type 1: Customer configures the VN as a set of VN Members.  
No other details need to be set by customer, making for a simplified operations for the customer.
  - \* VN Type 2: Along with VN Members, the customer could also provide an abstract topology, this topology is provided by the Abstract TE Topology Yang Model.

#### 5. ACTN VN YANG Model (Tree Structure)

```
module: ietf-actn-vn
 +--rw actn
```

```

 +--rw ap
 | +--rw access-point-list* [access-point-id]
 | | +--rw access-point-id uint32
 | | +--rw access-point-name? string
 | | +--rw max-bandwidth? te-types:te-bandwidth
 | | +--rw avl-bandwidth? te-types:te-bandwidth
 | | +--rw vn-ap* [vn-ap-id]
 | | | +--rw vn-ap-id uint32
 | | | +--rw vn? -> /actn/vn/vn-list/vn-id
 | | | +--rw abstract-node? ->
 | | +--rw ltp? te-types:te-tp-id
 | +--rw vn
 | | +--rw vn-list* [vn-id]
 | | | +--rw vn-id uint32
 | | | +--rw vn-name? string
 | | | +--rw vn-topology-id? te-types:te-topology-id
 | | | +--rw abstract-node? ->
 | | +--rw vn-member-list* [vn-member-id]
 | | | +--rw vn-member-id uint32
 | | | +--rw src
 | | | | +--rw src? -> /actn/ap/access-point-
 | | | | | +--rw src-vn-ap-id? -> /actn/ap/access-point-
 | | | | | | +--rw multi-src? boolean {multi-src-dest}?
 | | | | | | +--rw dest
 | | | | | | | +--rw dest? -> /actn/ap/access-point-
 | | | | | | | | +--rw dest-vn-ap-id? -> /actn/ap/access-point-
 | | | | | | | | | +--rw multi-dest? boolean {multi-src-dest}?
 | | | | | | | | | +--rw connetivity-matrix-id? ->
 | | | | | | | | | | +--ro oper-status? identityref
 | | | | | | | | | | +--ro if-selected? boolean {multi-src-dest}?
 | | | | | | | | | | +--ro admin-status? identityref
 | | | | | | | | | | +--ro oper-status? identityref
 | | | | | | | | | | +--rw vn-level-diversity? vn-disjointness
 | | +--rw vn-level-diversity? vn-disjointness
 | +--rw vn-level-diversity? vn-disjointness
 +--rw vn-level-diversity? vn-disjointness

```

```

rpcs:
 +---x vn-compute
 +---w input
 | +---w abstract-node? ->
/nw:networks/network/node/tet:te-node-id
 | +---w vn-member-list* [vn-member-id]
 | | +---w vn-member-id uint32
 | | +---w src
 | | | +---w src? -> /actn/ap/access-point-
list/access-point-id
 | | | +---w src-vn-ap-id? -> /actn/ap/access-point-
list/vn-ap/vn-ap-id
 | | | +---w multi-src? boolean {multi-src-dest}?
 | | | +---w dest
 | | | +---w dest? -> /actn/ap/access-point-
list/access-point-id
 | | | +---w dest-vn-ap-id? -> /actn/ap/access-point-
list/vn-ap/vn-ap-id
 | | | +---w multi-dest? boolean {multi-src-dest}?
 | | | +---w connetivity-matrix-id? ->
/nw:networks/network/node/tet:te/te-node-attributes/connectivity-
matrices/connectivity-matrix/id
 | +---w vn-level-diversity? vn-disjointness
 +--ro output
 +--ro vn-member-list* [vn-member-id]
 +--ro vn-member-id uint32
 +--ro src
 | +--ro src? -> /actn/ap/access-point-
list/access-point-id
 | +--ro src-vn-ap-id? -> /actn/ap/access-point-
list/vn-ap/vn-ap-id
 | +--ro multi-src? boolean {multi-src-dest}?
 +--ro dest
 | +--ro dest? -> /actn/ap/access-point-
list/access-point-id
 | +--ro dest-vn-ap-id? -> /actn/ap/access-point-
list/vn-ap/vn-ap-id
 | +--ro multi-dest? boolean {multi-src-dest}?

```

```

 +--ro connetivity-matrix-id? ->
/nw:networks/network/node/tet:te/te-node-attributes/connectivity-
matrices/connectivity-matrix/id
 +--ro if-selected? boolean {multi-src-
dest}?
 +--ro compute-status? identityref

```

## 6. ACTN-VN YANG Code

The YANG code is as follows:

```

<CODE BEGINS> file "ietf-actn-vn@2018-02-27.yang"

module ietf-actn-vn {
 namespace "urn:ietf:params:xml:ns:yang:ietf-actn-vn";
 prefix "vn";

 /* Import network */
 import ietf-network {
 prefix "nw";
 }

 /* Import TE generic types */
 import ietf-te-types {
 prefix "te-types";
 }

 /* Import Abstract TE Topology */
 import ietf-te-topology {
 prefix "tet";
 }

 organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";
 contact
 "Editor: Young Lee <leeyoung@huawei.com>
 : Dhruv Dhody <dhruv.ietf@gmail.com>";
 description
 "This module contains a YANG module for the ACTN VN. It
 describes a VN operation module that takes place in the
 context of the CNC-MDSC Interface (CMI) of the ACTN
 architecture where the CNC is the actor of a VN

```



```
 instantiation/modification /deletion.";
revision 2018-02-27 {
 description
 "initial version.";
 reference
 "TBD";
}
/*
 * Features
 */
feature multi-src-dest {
 description
 "Support for selection of one src or destination
 among multiple.";
}

/*identity path-metric-delay {
 base te-types:path-metric-type;
 description
 "delay path metric";
}
identity path-metric-delay-variation {
 base te-types:path-metric-type;
 description
 "delay-variation path metric";
}
identity path-metric-loss {
 base te-types:path-metric-type;
 description
 "loss path metric";
}*/

identity vn-state-type {
 description
 "Base identity for VN state";
}
identity vn-state-up {
 base vn-state-type;
 description "VN state up";
}
identity vn-state-down {
 base vn-state-type;
 description "VN state down";
}
identity vn-admin-state-type {
```

```
 description
 "Base identity for VN admin states";
 }
 identity vn-admin-state-up {
 base vn-admin-state-type;
 description "VN administratively state up";
 }
 identity vn-admin-state-down {
 base vn-admin-state-type;
 description "VN administratively state down";
 }
 identity vn-compute-state-type {
 description
 "Base identity for compute states";
 }
 identity vn-compute-state-computing {
 base vn-compute-state-type;
 description
 "State path compute in progress";
 }
 identity vn-compute-state-computation-ok {
 base vn-compute-state-type;
 description
 "State path compute successful";
 }
 identity vn-compute-state-computatione-failed {
 base vn-compute-state-type;
 description
 "State path compute failed";
 }
}
/*
 * Groupings
 */

typedef vn-disjointness {
 type bits {
 bit node {
 position 0;
 description "node disjoint";
 }
 bit link {
 position 1;
 description "link disjoint";
 }
 bit srlg {
```

```
 position 2;
 description "srlg disjoint";
 }
}
description
 "type of the resource disjointness for
 VN level applied across all VN members
 in a VN";
}

grouping vn-ap {
 description
 "VNAP related information";
 leaf vn-ap-id {
 type uint32;
 description
 "unique identifier for the referred
 VNAP";
 }
 leaf vn {
 type leafref {
 path "/actn/vn/vn-list/vn-id";
 }
 description
 "reference to the VN";
 }
 leaf abstract-node {
 type leafref {
 path "/nw:networks/nw:network/nw:node/"
 + "tet:te-node-id";
 }
 description
 "a reference to the abstract node in TE
 Topology";
 }
 leaf ltp {
 type te-types:te-tp-id;
 description
 "Reference LTP in the TE-topology";
 }
}

grouping access-point {
 description
 "AP related information";
 leaf access-point-id {
```

```
 type uint32;
 description
 "unique identifier for the referred
 access point";
 }
 leaf access-point-name {
 type string;
 description
 "ap name";
 }

 leaf max-bandwidth {
 type te-types:te-bandwidth;
 description
 "max bandwidth of the AP";
 }
 leaf avl-bandwidth {
 type te-types:te-bandwidth;
 description
 "available bandwidth of the AP";
 }
 /*add details and any other properties of AP,
 not associated by a VN
 CE port, PE port etc.
 */
 list vn-ap {
 key vn-ap-id;
 uses vn-ap;
 description
 "list of VNAP in this AP";
 }
} //access-point
grouping vn-member {
 description
 "vn-member is described by this container";
 leaf vn-member-id {
 type uint32;
 description
 "vn-member identifier";
 }
}
container src
{
 description
 "the source of VN Member";
 leaf src {
```

```
 type leafref {
 path "/actn/ap/access-point-list/access-point-id";
 }
 description
 "reference to source AP";
 }
 leaf src-vn-ap-id{
 type leafref {
 path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
 }
 description
 "reference to source VNAP";
 }
 leaf multi-src {
 if-feature multi-src-dest;
 type boolean;
 description
 "Is source part of multi-source, where
 only one of the source is enabled";
 }
}
container dest
{
 description
 "the destination of VN Member";
 leaf dest {
 type leafref {
 path "/actn/ap/access-point-list/access-point-id";
 }
 description
 "reference to destination AP";
 }
 leaf dest-vn-ap-id{
 type leafref {
 path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
 }
 description
 "reference to dest VNAP";
 }
 leaf multi-dest {
 if-feature multi-src-dest;
 type boolean;
 description
 "Is destination part of multi-destination, where
 only one of the destination is enabled";
 }
}
```

```
 }
 }
 leaf connectivity-matrix-id{
 type leafref {
 path "/nw:networks/nw:network/nw:node/tet:te/"
 + "tet:te-node-attributes/"
 + "tet:connectivity-matrices/"
 + "tet:connectivity-matrix/tet:id";
 }
 description
 "reference to connectivity-matrix";
 }
} //vn-member
/*
grouping policy {
 description
 "policy related to vn-member-id";
 leaf local-reroute {
 type boolean;
 description
 "Policy to state if reroute
 can be done locally";
 }
 leaf push-allowed {
 type boolean;
 description
 "Policy to state if changes
 can be pushed to the customer";
 }
 leaf incremental-update {
 type boolean;
 description
 "Policy to allow only the
 changes to be reported";
 }
}
} //policy
*/
grouping vn-policy {
 description
 "policy for VN-level diverisity";
 leaf vn-level-diversity {
 type vn-disjointness;
 description
 "the type of disjointness on the VN level
 (i.e., across all VN members)";
 }
}
```

```
 }
 }
 /*
 grouping metrics-op {
 description
 "metric related information";
 list metric{
 key "metric-type";
 config false;
 description
 "The list of metrics for VN";
 leaf metric-type {
 type identityref {
 base te-types:path-metric-type;
 }
 description
 "The VN metric type.";
 }
 leaf value{
 type uint32;
 description
 "The limit value";
 }
 }
 }
 */
 /*
 grouping metrics {
 description
 "metric related information";
 list metric{
 key "metric-type";
 description
 "The list of metrics for VN";
 uses te:path-metrics-bounds_config;
 container optimize{
 description
 "optimizing constraints";
 leaf enabled{
 type boolean;
 description
 "Metric to optimize";
 }
 leaf value{
 type uint32;
 }
 }
 }
 }
 */
```

```

 description
 "The computed value";
 }
}
}
}
/*
/*
grouping service-metric {
 description
 "service-metric";
 uses te:path-objective-function_config;
 uses metrics;
 uses te-types:common-constraints_config;
 uses te:protection-restoration-params_config;
 uses policy;
} //service-metric
/*
/*
* Configuration data nodes
*/
container actn {
 description
 "actn is described by this container";
 container ap {
 description
 "AP configurations";
 list access-point-list {
 key "access-point-id";
 description
 "access-point identifier";
 uses access-point {
 description
 "access-point information";
 }
 }
 }
}
container vn {
 description
 "VN configurations";
 list vn-list {
 key "vn-id";
 description
 "a virtual network is identified by a vn-id";
 leaf vn-id {

```



```
 type uint32;
 description
 "a unique vn identifier";
 }
 leaf vn-name {
 type string;
 description "vn name";
 }
 leaf vn-topology-id{
 type te-types:te-topology-id;
 description
 "An optional identifier to the TE Topology
 Model where the abstract nodes and links
 of the Topology can be found for Type 2
 VNS";
 }
 leaf abstract-node {
 type leafref {
 path "/nw:networks/nw:network/nw:node/"
 + "tet:te-node-id";
 }
 description
 "a reference to the abstract node in TE
 Topology";
 }
 list vn-member-list{
 key "vn-member-id";
 description
 "List of VN-members in a VN";
 uses vn-member;
 /*uses metrics-op;*/
 leaf oper-status {
 type identityref {
 base vn-state-type;
 }
 config false;
 description
 "VN-member operational state.";
 }
 }
}
leaf if-selected{
 if-feature multi-src-dest;
 type boolean;
 default false;
```

```

 config false;
 description
 "Is the vn-member is selected among the
 multi-src/dest options";
 }
/*
container multi-src-dest{
 if-feature multi-src-dest;
 config false;
 description
 "The selected VN Member when multi-src
 and/or mult-destination is enabled.";
 leaf selected-vn-member{
 type leafref {
 path "/actn/vn/vn-list/vn-member-list"
 + "/vn-member-id";
 }
 description
 "The selected VN Member along the set
 of source and destination configured
 with multi-source and/or multi-destination";
 }
}
*/
/*uses service-metric;*/
leaf admin-status {
 type identityref {
 base vn-admin-state-type;
 }
 default vn-admin-state-up;
 description "VN administrative state.";
}
leaf oper-status {
 type identityref {
 base vn-state-type;
 }
 config false;
 description "VN operational state.";
}
 uses vn-policy;
} //vn-list
} //vn
} //actn
/*
* Notifications - TBD

```

```
*/
/*
* RPC
*/
rpc vn-compute{
 description
 "The VN computation without actual
 instantiation";
 input {
 leaf abstract-node {
 type leafref {
 path "/nw:networks/nw:network/nw:node/"
 + "tet:te-node-id";
 }
 description
 "a reference to the abstract node in TE
 Topology";
 }
 list vn-member-list{
 key "vn-member-id";
 description
 "List of VN-members in a VN";
 uses vn-member;
 }
 uses vn-policy;
 /*uses service-metric;*/
 }
 output {
 list vn-member-list{
 key "vn-member-id";
 description
 "List of VN-members in a VN";
 uses vn-member;
 leaf if-selected{
 if-feature multi-src-dest;
 type boolean;
 default false;
 description
 "Is the vn-member is selected among
 the multi-src/dest options";
 }
 /*uses metrics-op;*/
 leaf compute-status {
 type identityref {
 base vn-compute-state-type;
 }
 }
 }
 }
}
```

```

 }
 description
 "VN-member compute state.";
 }
}
/*
container multi-src-dest{
 if-feature multi-src-dest;
 description
 "The selected VN Member when multi-src
 and/or mult-destination is enabled.";
 leaf selected-vn-member-id{
 type uint32;
 description
 "The selected VN Member-id from the
 input";
 }
}*/
}
}
}

```

<CODE ENDS>

## 7. JSON Example

This section provides json implementation examples as to how ACTN VN YANG model and TE topology model are used together to instantiate virtual networks.

The example in this section includes following VN

- o VN1 (Type 1): Which maps to the single node topology abstract1 (node D1) and consist of VN Members 104 (L1 to L4), 107 (L1 to L7), 204 (L2 to L4), 308 (L3 to L8) and 108 (L1 to L8). We also show how disjointness (node, link, srlg) is supported in the example on the global level (i.e., connectivity matrices level).

- o VN2 (Type 2): Which maps to the single node topology abstract2 (node D2), this topology has an underlay topology (absolute) (see figure in section 3.2). This VN has a single VN member 105 (L1 to L5) and an underlay path (S4 and S7) has been set in the connectivity matrix of abstract2 topology;
- o VN3 (Type 1): This VN has a multi-source, multi-destination feature enable for VN Member 104 (L1 to L4)/107 (L1 to L7) [multi-src] and VN Member 204 (L2 to L4)/304 (L3 to L4) [multi-dest] usecase. The selected VN-member is known via the field "if-selected" and the corresponding connectivity-matrix-id.

Note that the ACTN VN YANG model also include the AP and VNAP which shows various VN using the same AP.

#### 7.1. ACTN VN JSON

```
{
 "actn": {
 "ap": {
 "access-point-list": [
 {
 "access-point-id": 101,
 "access-point-name": "101",
 "vn-ap": [
 {
 "vn-ap-id": 10101,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "1-0-1"
 },
 {
 "vn-ap-id": 10102,
 "vn": 2,
 "abstract-node": "D2",
 "ltp": "1-0-1"
 },
 {
 "vn-ap-id": 10103,
 "vn": 3,
 "abstract-node": "D3",
 "ltp": "1-0-1"
 }
]
 },
 {
 "access-point-id": 202,
 "access-point-name": "202",
 "vn-ap": [

```

```
 {
 "vn-ap-id": 20201,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "2-0-2"
 }
],
},
{
 "access-point-id": 303,
 "access-point-name": "303",
 "vn-ap": [
 {
 "vn-ap-id": 30301,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "3-0-3"
 },
 {
 "vn-ap-id": 30303,
 "vn": 3,
 "abstract-node": "D3",
 "ltp": "3-0-3"
 }
]
},
{
 "access-point-id": 440,
 "access-point-name": "440",
 "vn-ap": [
 {
 "vn-ap-id": 44001,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "4-4-0"
 }
]
},
{
 "access-point-id": 550,
 "access-point-name": "550",
 "vn-ap": [
 {
 "vn-ap-id": 55002,
 "vn": 2,
 "abstract-node": "D2",
 "ltp": "5-5-0"
 }
]
}
```

```

 },
 {
 "access-point-id": 770,
 "access-point-name": "770",
 "vn-ap": [
 {
 "vn-ap-id": 77001,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "7-7-0"
 },
 {
 "vn-ap-id": 77003,
 "vn": 3,
 "abstract-node": "D3",
 "ltp": "7-7-0"
 }
]
 },
 {
 "access-point-id": 880,
 "access-point-name": "880",
 "vn-ap": [
 {
 "vn-ap-id": 88001,
 "vn": 1,
 "abstract-node": "D1",
 "ltp": "8-8-0"
 },
 {
 "vn-ap-id": 88003,
 "vn": 3,
 "abstract-node": "D3",
 "ltp": "8-8-0"
 }
]
 }
],
 "vn": {
 "vn-list": [
 {
 "vn-id": 1,
 "vn-name": "vn1",
 "vn-topology-id": "te-topology:abstract1",
 "abstract-node": "D1",
 "vn-member-list": [
 {
 "vn-member-id": 104,

```

```
 "src": {
 "src": 101,
 "src-vn-ap-id": 10101,
 },
 "dest": {
 "dest": 440,
 "dest-vn-ap-id": 44001,
 },
 "connectivity-matrix-id": 104
 },
 {
 "vn-member-id": 107,
 "src": {
 "src": 101,
 "src-vn-ap-id": 10101,
 },
 "dest": {
 "dest": 770,
 "dest-vn-ap-id": 77001,
 },
 "connectivity-matrix-id": 107
 },
 {
 "vn-member-id": 204,
 "src": {
 "src": 202,
 "dest-vn-ap-id": 20401,
 },
 "dest": {
 "dest": 440,
 "dest-vn-ap-id": 44001,
 },
 "connectivity-matrix-id": 204
 },
 {
 "vn-member-id": 308,
 "src": {
 "src": 303,
 "src-vn-ap-id": 30301,
 },
 "dest": {
 "dest": 880,
 "src-vn-ap-id": 88001,
 },
 "connectivity-matrix-id": 308
 },
 {
 "vn-member-id": 108,
 "src": {
```



```

 "src": 101,
 "src-vn-ap-id": 10101,
 },
 "dest": {
 "dest": 880,
 "dest-vn-ap-id": 88001,
 },
 "connectivity-matrix-id": 108
 }
]
},
{
 "vn-id": 2,
 "vn-name": "vn2",
 "vn-topology-id": "te-topology:abstract2",
 "abstract-node": "D2",
 "vn-member-list": [
 {
 "vn-member-id": 105,
 "src": {
 "src": 101,
 "src-vn-ap-id": 10102,
 },
 "dest": {
 "dest": 550,
 "dest-vn-ap-id": 55002,
 },
 "connectivity-matrix-id": 105
 }
]
},
{
 "vn-id": 3,
 "vn-name": "vn3",
 "vn-topology-id": "te-topology:abstract3",
 "abstract-node": "D3",
 "vn-member-list": [
 {
 "vn-member-id": 104,
 "src": {
 "src": 101,
 },
 "dest": {
 "dest": 440,
 "multi-dest": true
 }
 }
],
 {
 "vn-member-id": 107,

```

```

 "src": {
 "src": 101,
 "src-vn-ap-id": 10103,
 },
 "dest": {
 "dest": 770,
 "dest-vn-ap-id": 77003,
 "multi-dest": true
 },
 "connectivity-matrix-id": 107,
 "if-selected": true,
 },
 {
 "vn-member-id": 204,
 "src": {
 "src": 202,
 "multi-src": true,
 },
 "dest": {
 "dest": 440,
 },
 },
 {
 "vn-member-id": 304,
 "src": {
 "src": 303,
 "src-vn-ap-id": 30303,
 "multi-src": true,
 },
 "dest": {
 "dest": 440,
 "src-vn-ap-id": 44003,
 },
 "connectivity-matrix-id": 304,
 "if-selected": true,
 },
],
},
]
}
}
}

```

## 7.2. TE-topology JSON

```

{
 "networks": {

```

```
"network": [
 {
 "network-types": {
 "te-topology": {}
 },
 "network-id": "abstract1",
 "provider-id": 201,
 "client-id": 600,
 "te-topology-id": "te-topology:abstract1",
 "node": [
 {
 "node-id": "D1",
 "te-node-id": "2.0.1.1",
 "te": {
 "te-node-attributes": {
 "domain-id": 1,
 "is-abstract": [null],
 "connectivity-matrices": {
 "is-allowed": true,
 "path-constraints": {
 "bandwidth-generic": {
 "te-bandwidth": {
 "generic": [
 {
 "generic": "0x1p10",
 }
]
 }
 }
 }
 }
 }
 },
 "disjointness": "node link srlg",
 },
],
 "connectivity-matrix": [
 {
 "id": 104,
 "from": "1-0-1",
 "to": "4-4-0"
 },
 {
 "id": 107,
 "from": "1-0-1",
 "to": "7-7-0"
 },
 {
 "id": 204,
 "from": "2-0-2",
 "to": "4-4-0"
 },
]
 }
]
```

```

 "id": 308,
 "from": "3-0-3",
 "to": "8-8-0"
 },
 {
 "id": 108,
 "from": "1-0-1",
 "to": "8-8-0"
 }
],
 }
},
"termination-point": [
 {
 "tp-id": "1-0-1",
 "te-tp-id": 10001,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "1-1-0",
 "te-tp-id": 10100,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "2-0-2",
 "te-tp-id": 20002,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
]
}

```

```
 },
 {
 "tp-id": "2-2-0",
 "te-tp-id": 20200,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
],
 {
 "tp-id": "3-0-3",
 "te-tp-id": 30003,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
],
{
 "tp-id": "3-3-0",
 "te-tp-id": 30300,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "4-0-4",
 "te-tp-id": 40004,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
}
```

```
{
 "tp-id": "4-4-0",
 "te-tp-id": 40400,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "5-0-5",
 "te-tp-id": 50005,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "5-5-0",
 "te-tp-id": 50500,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "6-0-6",
 "te-tp-id": 60006,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
}
}
```

```
"tp-id": "6-6-0",
"te-tp-id": 60600,
"te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
},
{
 "tp-id": "7-0-7",
 "te-tp-id": 70007,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "7-7-0",
 "te-tp-id": 70700,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "8-0-8",
 "te-tp-id": 80008,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "8-8-0",
```

```

 "te-tp-id": 80800,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
]
},
{
 "network-types": {
 "te-topology": {}
 },
 "network-id": "abstract2",
 "provider-id": 201,
 "client-id": 600,
 "te-topology-id": "te-topology:abstract2",
 "node": [
 {
 "node-id": "D2",
 "te-node-id": "2.0.1.2",
 "te": {
 "te-node-attributes": {
 "domain-id": 1,
 "is-abstract": [null],
 "connectivity-matrices": {
 "is-allowed": true,
 "underlay": {
 "enabled": true
 }
 },
 "path-constraints": {
 "bandwidth-generic": {
 "te-bandwidth": {
 "generic": [
 {
 "generic": "0x1p10"
 }
]
 }
 }
 }
 },
 "optimizations": {
 "objective-function": {

```



```

 "objective-function-type": "of-maximize-residual-
bandwidth"
 },
 "connectivity-matrix": [
 {
 "id": 105,
 "from": "1-0-1",
 "to": "5-5-0",
 "underlay": {
 "enabled": true,
 "primary-path": {
 "network-ref": "absolute",
 "path-element": [
 {
 "path-element-id": 1,
 "index": 1,
 "numbered-hop": {
 "address": "4.4.4.4",
 "hop-type": "STRICT"
 }
 },
 {
 "path-element-id": 2,
 "index": 2,
 "numbered-hop": {
 "address": "7.7.7.7",
 "hop-type": "STRICT"
 }
 }
]
 }
 }
 }
],
 "termination-point": [
 {
 "tp-id": "1-0-1",
 "te-tp-id": 10001,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
]
}

```

```
 }
 },
 {
 "tp-id": "1-1-0",
 "te-tp-id": 10100,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "2-0-2",
 "te-tp-id": 20002,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "2-2-0",
 "te-tp-id": 20200,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "3-0-3",
 "te-tp-id": 30003,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
}
```

```
 },
 {
 "tp-id": "3-3-0",
 "te-tp-id": 30300,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
],
 {
 "tp-id": "4-0-4",
 "te-tp-id": 40004,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
],
{
 "tp-id": "4-4-0",
 "te-tp-id": 40400,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "5-0-5",
 "te-tp-id": 50005,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
}
```

```

{
 "tp-id": "5-5-0",
 "te-tp-id": 50500,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "6-0-6",
 "te-tp-id": 60006,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "6-6-0",
 "te-tp-id": 60600,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "7-0-7",
 "te-tp-id": 70007,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{

```

```

 "tp-id": "7-7-0",
 "te-tp-id": 70700,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "8-0-8",
 "te-tp-id": 80008,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "8-8-0",
 "te-tp-id": 80800,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
]
},
{
 "network-types": {
 "te-topology": {}
 },
 "network-id": "abstract3",
 "provider-id": 201,
 "client-id": 600,
 "te-topology-id": "te-topology:abstract3",
 "node": [
 {

```

```

"node-id": "D3",
"te-node-id": "3.0.1.1",
"te": {
 "te-node-attributes": {
 "domain-id": 3,
 "is-abstract": [null],
 "connectivity-matrices": {
 "is-allowed": true,
 "path-constraints": {
 "bandwidth-generic": {
 "te-bandwidth": {
 "generic": [
 {
 "generic": "0x1p10",
 }
]
 }
 }
 },
 "connectivity-matrix": [
 {
 "id": 107,
 "from": "1-0-1",
 "to": "7-7-0"
 },
 {
 "id": 308,
 "from": "3-0-3",
 "to": "8-8-0"
 }
],
 }
 },
 "termination-point": [
 {
 "tp-id": "1-0-1",
 "te-tp-id": 10001,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "1-1-0",

```

```
"te-tp-id": 10100,
"te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
},
{
 "tp-id": "2-0-2",
 "te-tp-id": 20002,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "2-2-0",
 "te-tp-id": 20200,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "3-0-3",
 "te-tp-id": 30003,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
},
{
 "tp-id": "3-3-0",
 "te-tp-id": 30300,
```

```

 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "4-0-4",
 "te-tp-id": 40004,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "4-4-0",
 "te-tp-id": 40400,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "5-0-5",
 "te-tp-id": 50005,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "5-5-0",
 "te-tp-id": 50500,
 "te": {

```



```
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 },
 {
 "tp-id": "6-0-6",
 "te-tp-id": 60006,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "6-6-0",
 "te-tp-id": 60600,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "7-0-7",
 "te-tp-id": 70007,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "7-7-0",
 "te-tp-id": 70700,
 "te": {
 "interface-switching-capability": [
```

```

 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "8-0-8",
 "te-tp-id": 80008,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 },
 {
 "tp-id": "8-8-0",
 "te-tp-id": 80800,
 "te": {
 "interface-switching-capability": [
 {
 "switching-capability": "switching-otn",
 "encoding": "lsp-encoding-oduk"
 }
]
 }
 }
]
}

```

## 8. Security Considerations

TDB

## 9. IANA Considerations

TDB

## 10. Acknowledgments

The authors would like to thank Xufeng Liu for his helpful comments and valuable suggestions.

## 11. References

### 11.1. Normative References

- [TE-TOPO] X. Liu, et al., "YANG Data Model for TE Topologies", work in progress: draft-ietf-teas-yang-te-topo.
- [TE-tunnel] T. Saad, et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", work in progress: draft-ietf-teas-yang-te.

### 11.2. Informative References

- [RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.
- [ACTN-REQ] Lee, et al., "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.
- [ACTN-FWK] D. Ceccarelli, Y. Lee [Editors], "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ceccarelli-teas-actn-framework, work in progress.
- [TE-MAP] Y. Lee, D. Dhody, and D. Ceccarelli, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.
- [SERVICE-YANG] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained, work in progress.
- [ACTN-PM] Y. Lee, et al., "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.
- [OIF-VTNS] Virtual Transport Network Services 1.0 Specification, IA OIF-VTNS-1.0, April 2017.

## 12. Contributors

### Contributor's Addresses

Haomian Zheng  
Huawei Technologies  
Email: zhenghaomian@huawei.com

Xian Zhang  
Huawei Technologies  
Email: zhang.xian@huawei.com

Sergio Belotti  
Nokia  
Email: sergio.belotti@nokia.com

Qin Wu  
Huawei Technologies  
Email: bill.wu@huawei.com

Takuya Miyasaka  
KDDI  
Email: ta-miyasaka@kddi.com

Peter Park  
KT  
Email: peter.park@kt.com

### Authors' Addresses

Young Lee (ed.)  
Huawei Technologies  
Email: leeyoung@huawei.com

Dhruv Dhody  
Huawei Technologies  
Email: dhruv.ietf@gmail.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden  
Email: daniele.ceccarelli@ericsson.com

Igor Bryskin  
Huawei  
Email: Igor.Bryskin@huawei.com

Bin Yeong Yoon  
ETRI  
Email: byyun@etri.re.kr



TEAS WG  
Internet Draft  
Intended status: standard track  
Expires: June 30, 2019

Young Lee  
Dhruv Dhody  
Huawei

Daniele Ceccarelli  
Ericsson

Jeff Tantsura  
Apstra

Giuseppe Fioccola  
Huawei

Qin Wu  
Huawei

December 30, 2018

## Traffic Engineering and Service Mapping Yang Model

draft-lee-teas-te-service-mapping-yang-13

### Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPM Service Model) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Abstraction and Control of Traffic Engineered Networks Virtual Network model). This model is referred to as TE Service Mapping Model and is applicable generically to the operator's need for seamless control and management of their VPN services with TE tunnel support.

The model is principally used to allow monitoring and diagnostics of the management systems to show how the service requests are mapped onto underlying network resource and TE models.

### 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 June 30, 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 (<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  |
| 1.1. Terminology.....                                          | 4  |
| 1.2. Tree diagram.....                                         | 4  |
| 1.3. Prefixes in Data Node Names.....                          | 4  |
| 2. TE & Service Related Parameters.....                        | 5  |
| 2.1. VN/Tunnel Selection Requirements.....                     | 5  |
| 2.2. Availability Requirement.....                             | 7  |
| 3. YANG Modeling Approach.....                                 | 7  |
| 3.1. Forward Compatibility.....                                | 8  |
| 4. L3VPN Architecture in the ACTN Context.....                 | 8  |
| 4.1. Service Mapping.....                                      | 11 |
| 4.2. Site Mapping.....                                         | 12 |
| 5. Applicability of TE-Service Mapping in Generic context..... | 12 |
| 6. YANG Data Trees.....                                        | 13 |

|                                   |    |
|-----------------------------------|----|
| 7. YANG Data Models.....          | 14 |
| 8. Security.....                  | 24 |
| 9. IANA Considerations.....       | 24 |
| 10. Acknowledgements.....         | 26 |
| 11. References.....               | 26 |
| 11.1. Informative References..... | 26 |
| 12. Contributors.....             | 27 |
| Authors' Addresses.....           | 27 |

## 1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modeling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [RFC8199] and [RFC8309].

Framework for Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453] introduces an architecture to support virtual network services and connectivity services. [ACTN-VN-YANG] defines a YANG model and describes how customers or end-to-end orchestrators can request and/or instantiate a generic virtual network service. [ACTN-Applicability] describes the way IETF YANG models of different classifications can be applied to the ACTN interfaces. In particular, it describes how customer service models can be mapped into the CNC-MDSC Interface (CMI) of the ACTN architecture.

The models presented in this document are also applicable in generic context [RFC8309] as part of Customer Service Model used between Service Orchestrator and Customer.

[RFC8299] provides a L3VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[L2SM] provides a L2VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[L1CSM] provides a L1 connectivity service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

While the IP/MPLS Provisioning Network Controller (PNC) is responsible for provisioning the VPN service on the Provider Edge (PE) nodes, the Multi-Domain Service Coordinator (MDSC) can coordinate how to map the VPN services onto Traffic Engineering (TE) tunnels. This is consistent with the two of the core functions of the MDSC specified in [RFC8453]:

- . Customer mapping/translation function: This function is to map customer requests/commands into network provisioning requests that can be sent to the PNC according to the business policies that have been provisioned statically or dynamically. Specifically, it provides mapping and translation of a customer's service request into a set of parameters that are specific to a network type and technology such that the network configuration process is made possible.
- . Virtual service coordination function: This function translates customer service-related information into virtual network service operations in order to seamlessly operate virtual networks while meeting a customer's service requirements. In the context of ACTN, service/virtual service coordination includes a number of service orchestration functions such as multi-destination load balancing, guarantees of service quality, bandwidth and throughput. It also includes notifications for service fault and performance degradation and so forth.

Section 2 describes a set of TE & service related parameters that this document addresses as new and advanced parameters that are not included in generic service models. Section 3 discusses YANG modeling approach.

### 1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

### 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

### 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   |
|-----------|-------------------------------|-------------|
| tsm-types | ietf-te-service-mapping-types | [RFCXXXX]   |
| l1        | ietf-l1csm                    | [L1CSM]     |
| l2vpn-svc | ietf-l2vpn-svc                | [L2SM]      |
| l3vpn-svc | ietf-l3vpn-svc                | [RFC8299]   |
| l1-tsm    | ietf-l1csm-te-service-mapping | [RFCXXXX]   |
| l2-tsm    | ietf-l2sm-te-service-mapping  | [RFCXXXX]   |
| l3-tsm    | ietf-l3sm-te-service-mapping  | [RFCXXXX]   |
| vn        | ietf-vn                       | [ACTN-VN]   |
| nw        | ietf-network                  | [RFC8345]   |
| te-types  | ietf-te-types                 | [TE-Types]  |
| te-topo   | ietf-te-topology              | [TE-Topo]   |
| te        | ietf-te                       | [TE-Tunnel] |

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

## 2. TE & Service Related Parameters

While L1/2/3 service models [L1CSM, L2SM, L3SM] are intended to provide service-specific parameters for VPN service instances, there are a number of TE & Service related parameters that are not included in the generic service models.

Additional service parameters and policies that are not included in the aforementioned service models are addressed in the YANG models defined in this document.

### 2.1. VN/Tunnel Selection Requirements

In some cases, the service requirements may need addition TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example, for network slicing. The establishment of TE tunnels is subject to the network operator's policies.

To summarize, there are three modes of VN/Tunnel selection operations to be supported as follows. Additional modes may be defined in the future.

- o New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN.

Under this mode, the following sub-categories can be supported:

1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.
  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.
  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.
- o VN/Tunnel Sharing - A customer could request a VPN service where new tunnels (or a VN) do not need to be created for each VPN and can be shared across multiple VPNs. Further, the mapping YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the tunnels in use. No modification of the properties of a tunnel (or VN) is allowed in this mode: an existing tunnel can only be selected.
  - o VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

## 2.2. Availability Requirement

Availability is another service requirement or intent that may influence the selection or provisioning of TE tunnels or a VN to support the requested service. Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure.

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this draft.

## 3. YANG Modeling Approach

This section provides how the TE & Service mapping parameters are supported using augmentation of the existing service models (i.e., [L1CSM], [L2SM], and [L3SM]). Figure 1 shows the scope of the Augmented LxSM Model.

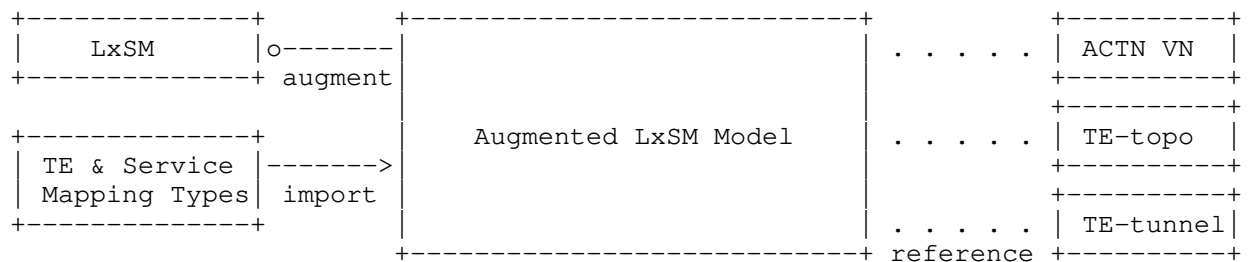


Figure 1. Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE & Service related parameters (defined in Section 2) grouping information from TE & Service Mapping Types. The TE & Service Mapping Types (ietf-te-service-mapping-types) module is the repository of all common groupings imported by each augmented LxSM model. Any future service models would import this grouping file.

The role of the augmented LxSm service model is to expose the mapping relationship between service models and TE models so that VN/VPN service instantiations provided by the underlying TE networks

can be viewed outside of the MDSC, for example by an operator who is diagnosing the behavior of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various constraints via TE-topology model [TE-topo], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

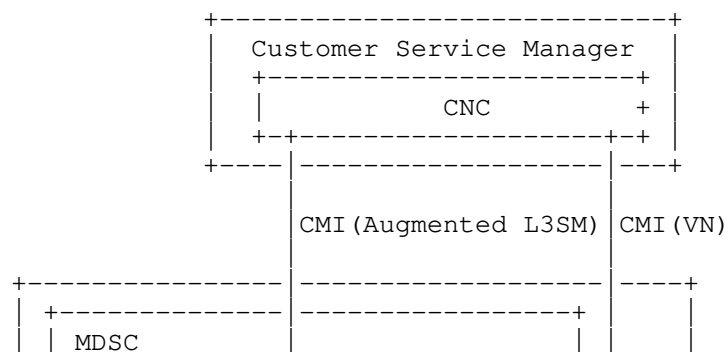
### 3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE & Service Mapping Types.

It is possible that new service models will be defined at some future time and that it will be desirable to map them to underlying TE constructs in the same way as the three existing models are augmented.

## 4. L3VPN Architecture in the ACTN Context

Figure 2 shows the architectural context of this document referencing the ACTN components and interfaces.



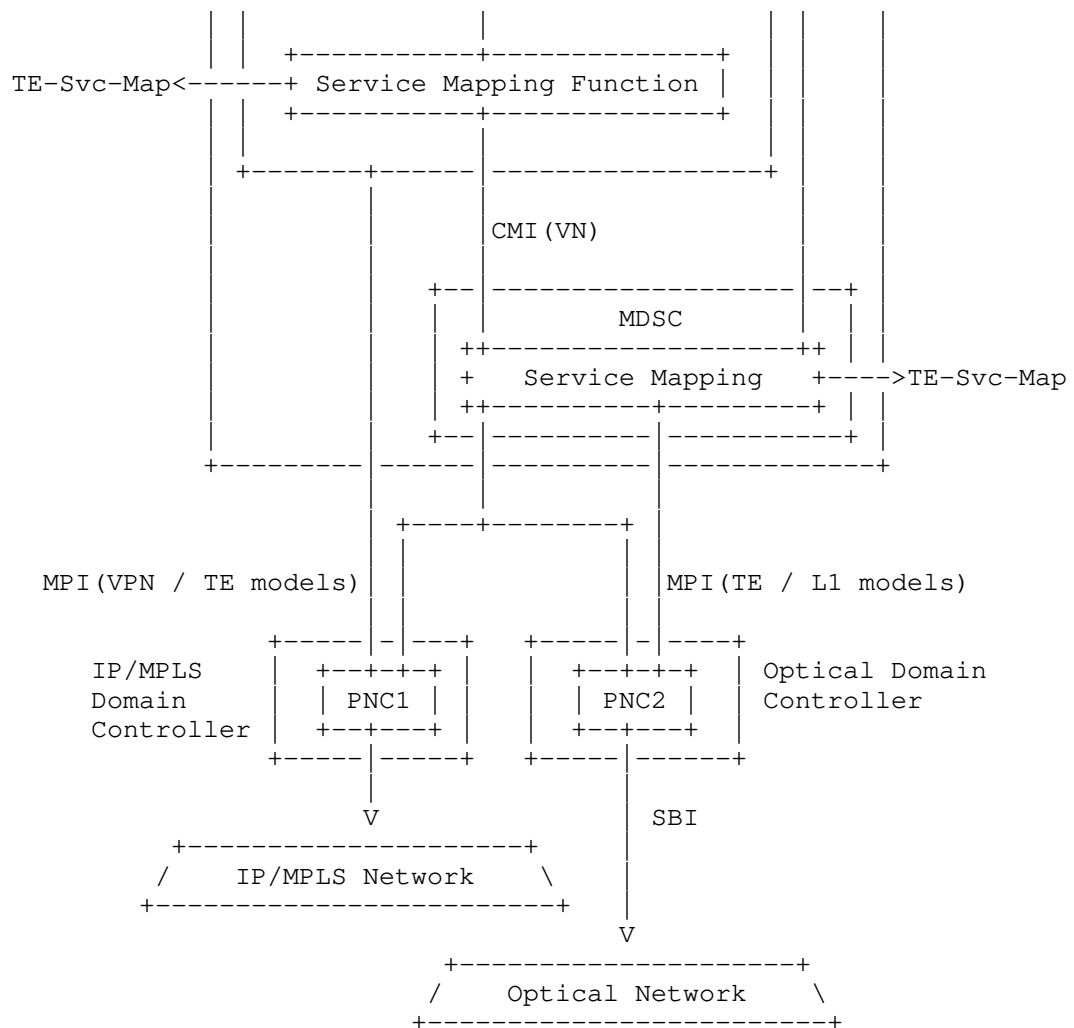


Figure 2: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 2.

- . CNC: The Customer Network Controller is responsible for generating service requests. In the context of an L3VPN, the CNC uses the Augmented L3SM to express the service request and communicate it to the network operator.



- . MDSC: This entity is responsible for coordinating a L3VPN service request (expressed via the Augmented L3SM) with the IP/MPLS PNC and the Transport PNC. For TE services, one of the key responsibilities of the MDSC is to coordinate with both the IP PNC and the Transport PNC for the mapping of the Augmented L3VPN Service Model to the ACTN VN model. In the VN/TE-tunnel binding case, the MDSC will need to coordinate with the Transport PNC to dynamically create the TE-tunnels in the transport network as needed. These tunnels are added as links in the IP/MPLS Layer topology. The MDSC coordinates with IP/MPLS PNC to create the TE-tunnels in the IP/MPLS layer, as part of the ACTN VN creation.
- . PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 2 shows two distinct PNCs.
  - o IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.
  - o Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

There are four main interfaces shown in Figure 2.

- . CMI: The CNC-MDSC Interface is used to communicate service requests from the customer to the operator. The requests may be expressed as Augmented VPN service requests (L2SM, L3SM), as connectivity requests (L1CSM), or as virtual network requests (ACTN VN).
- . MPI: The MDSC-PNC Interface is used by the MDSC to orchestrate networks under the control of PNCs. The requests on this interface may use TE tunnel models, TE topology models, VPN network configuration models or layer one connectivity models.
- . SBI: The Southbound Interface is used by the PNC to control network devices and is out of scope for this document.
- . The TE Service Mapping Model as described in this document can be used to see the mapping between service models and VN models and TE Tunnel/Topology models. That mapping may occur in the CNC if a service request is mapped to a VN request. Or it may occur in the MDSC where a service request is mapped to a TE tunnel, TE topology, or VPN network configuration model. The TE Service Mapping Model may be read from the CNC or MDSC to understand how the mapping has been made and to see the purpose for which network resources are used.

As shown in Figure 2, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

The high-level control flows for one example are as follows:

1. A customer asks for an L3VPN between CE1 and CE2 using the Augmented L3SM model.
2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [ACTN-VN-YANG] is used to configure a new VN based on this VPN and map the VPN service to the ACTN VN. In case an existing tunnel is to be used, each device will select which tunnel to use and populate this mapping information.
3. The MDSC interacts with both the IP/MPLS PNC and the Transport PNC to create a PE-PE tunnel in the IP network mapped to a TE tunnel in the transport network by providing the inter-layer access points and tunnel requirements. The specific service information is passed to the IP/MPLS PNC for the actual VPN configuration and activation.
  - a. The Transport PNC creates the corresponding TE tunnel matching with the access point and egress point.
  - b. The IP/MPLS PNC maps the VPN ID with the corresponding TE tunnel ID to bind these two IDs.
4. The IP/MPLS PNC creates/updates a VRF instance for this VPN customer. This is not in the scope of this document.

#### 4.1. Service Mapping

Augmented L3SM and L2SM can be used to request VPN service creation including the creation of sites and corresponding site network access connection between CE and PE. A VPN-ID is used to identify each VPN service ordered by the customer. The ACTN VN can be used further to establish PE-to-PE connectivity between VPN sites belonging to the same VPN service. A VN-ID is used to identify each virtual network established between VPN sites.

Once the ACTN VN has been established over the TE network (maybe a new VN, maybe modification of an existing VN, or maybe the use of an unmodified existing VN), the mapping between the VPN service and the ACTN VN service can be created.

## 4.2. Site Mapping

The elements in Augmented L3SM and L2SM define site location parameters and constraints such as distance and access diversity that can influence the placement of network attachment points (i.e., virtual network access points (VNAP)). To achieve this, a central directory can be set up to establish the mapping between location parameters and constraints and network attachment point location. Suppose multiple attachment points are matched, the management system can use constraints or other local policy to select the best candidate network attachment points.

After a network attachment point is selected, the mapping between VPN site and VNAP can be established as shown in Table 1.

| Site  | Site Network Access | Location<br>(Address, Postal Code, State, City, Country Code) | Access Diversity<br>(Constraint-Type, Group-id, Target Group-id) | PE  |
|-------|---------------------|---------------------------------------------------------------|------------------------------------------------------------------|-----|
| SITE1 | ACCESS1             | (,,US,NewYork,)                                               | (10,PE-Diverse,10)                                               | PE1 |
| SITE2 | ACCESS2             | (,,CN,Beijing,)                                               | (10,PE-Diverse,10)                                               | PE2 |
| SITE3 | ACCESS3             | (,,UK,London, )                                               | (12,same-PE,12)                                                  | PE4 |
| SITE4 | ACCESS4             | (,,FR,Paris,)                                                 | (20,Bearer-Diverse,20)                                           | PE7 |

Table 1 : Mapping Between VPN Site and VNAP

## 5. Applicability of TE-Service Mapping in Generic context

As discussed in the Introduction Section, the models presented in this document are also applicable generically outside of the ACTN architecture. [RFC8309] defines Customer Service Model between Customer and Service Orchestrator and Service Delivery Model between Service Orchestrator and Network Orchestrator(s). TE-Service mapping models defined in this document can be regarded primarily as Customer Service Model and secondarily as Service Deliver Model.

## 6. YANG Data Trees

```

module: ietf-l1csm-te-service-mapping
 augment /l1:l1-connectivity/l1:services/l1:service:
 +-rw te-service-mapping!
 augment /l1:l1-connectivity/l1:services/l1:service:
 +-rw te-mapping
 +-rw map-type? identityref
 +-rw availability-type? identityref
 +-rw (te)?
 +--:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/vn/vn-list/vn-id
 +--:(te-topo)
 | +-rw vn-topology-id? te-types:te-topology-id
 | +-rw abstract-node? -> /nw:networks/network/node/node-id
 +--:(te-tunnel)
 +-rw te-tunnel-list* te:tunnel-ref
 augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1:
 +-rw (te)?
 +--:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/ap/access-point-list/access-point-id
 +--:(te)
 +-rw ltp? te-types:te-tp-id
 augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2:
 +-rw (te)?
 +--:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/ap/access-point-list/access-point-id
 +--:(te)
 +-rw ltp? te-types:te-tp-id

module: ietf-l2sm-te-service-mapping
 augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
 +-rw te-service-mapping!
 augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
 +-rw te-mapping
 +-rw map-type? identityref
 +-rw availability-type? identityref
 +-rw (te)?
 +--:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/vn/vn-list/vn-id
 +--:(te-topo)
 | +-rw vn-topology-id? te-types:te-topology-id
 | +-rw abstract-node? -> /nw:networks/network/node/node-id
 +--:(te-tunnel)
 +-rw te-tunnel-list* te:tunnel-ref

```

```

 augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site/l2vpn-svc:site-net
work-
accesses/l2vpn-svc:site-network-access:
 +-rw (te)?
 +-:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/ap/access-point-list/access-point-id
 +-:(te)
 | +-rw ltp? te-types:te-tp-id

module: ietf-l3sm-te-service-mapping
 augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
 +-rw te-service-mapping!
 augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
 +-rw te-mapping
 +-rw map-type? identityref
 +-rw availability-type? identityref
 +-rw (te)?
 +-:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/vn/vn-list/vn-id
 +-:(te-topo)
 | +-rw vn-topology-id? te-types:te-topology-id
 | +-rw abstract-node? -> /nw:networks/network/node/node-id
 +-:(te-tunnel)
 | +-rw te-tunnel-list* te:tunnel-ref
 augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site/l3vpn-svc:site-
network-accesses/l3vpn-svc:site-network-access:
 +-rw (te)?
 +-:(actn-vn)
 | +-rw actn-vn-ref? -> /vn:actn/ap/access-point-list/access-point-i
d
 +-:(te)
 | +-rw ltp? te-types:te-tp-id

```

## 7. YANG Data Models

The YANG codes are as follows:

```
<CODE BEGINS> file "ietf-te-service-mapping-types@2018-12-30.yang"
```

```

module ietf-te-service-mapping-types {

 namespace "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";

 prefix "tsm";

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

```

```
 }

 import ietf-network {
 prefix "nw";
 }

 import ietf-te {
 prefix "te";
 }

 import ietf-vn {
 prefix "vn";
 }

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

 contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Dhruv Dhody <dhruv.ietf@gmail.com>
 Qin Wu <bill.wu@huawei.com>";

 description
 "This module contains a YANG module for TE & Service mapping
 parameters and policies as a common grouping applicable to
 various service models (e.g., L1CSM, L2SM, L3SM, etc.)";

 revision 2018-12-30 {
 description
 "initial version.";
 reference
 "TBD";
 }

 /*
 * Identity for map-type
 */
 identity map-type {
 description
 "Base identity from which specific map types are
 derived.";
 }

 identity new {
 base map-type;
 description
 "The new VN/tunnels are binded to the service.";
 }
```

```
identity detnet-hard-isolation {
 base new;
 description
 "Hard isolation with deterministic characteristics.";
}

identity hard-isolation {
 base new;
 description
 "Hard isolation.";
}

identity soft-isolation {
 base new;
 description
 "Soft-isolation.";
}

identity select {
 base map-type;
 description
 "The VPN service selects an existing tunnel with no
 modification.";
}

identity modify {
 base map-type;
 description
 "The VPN service selects an existing tunnel and allows
 to modify the properties of the tunnel (e.g., b/w)";
}

/*
 * Identity for availability-type
 */
identity availability-type {
 description
 "Base identity from which specific map types are
 derived.";
}

identity level-1 {
 base availability-type;
 description
 "level 1: 99.9999%";
}

identity level-2 {
```

```
 base availability-type;
 description
 "level 2: 99.999%";
 }

 identity level-3 {
 base availability-type;
 description
 "level 3: 99.99%";
 }

 identity level-4 {
 base availability-type;
 description
 "level 4: 99.9%";
 }

 identity level-5 {
 base availability-type;
 description
 "level 5: 99%";
 }

 /*
 * Groupings
 */

 grouping te-ref {
 description
 "The reference to TE.";
 choice te {
 description
 "The TE";
 case actn-vn {
 leaf actn-vn-ref {
 type leafref {
 path "/vn:actn/vn:vn/vn:vn-list/vn:vn-id";
 }
 description
 "The reference to ACTN VN";
 }
 }
 case te-topo {
 leaf vn-topology-id {
 type te-types:te-topology-id;
 description
 "An identifier to the TE Topology Model
 where the abstract nodes and links of
 the Topology can be found for Type 2";
 }
 }
 }
 }
}
```



```
 VNS";
 }
 leaf abstract-node {
 type leafref {
 path "/nw:networks/nw:network/nw:node/"
 + "nw:node-id";
 }
 description
 "a reference to the abstract node in TE
 Topology";
 }
}
case te-tunnel {
 leaf-list te-tunnel-list {
 type te:tunnel-ref;
 description
 "Reference to TE Tunnels";
 }
}
}

grouping te-endpoint-ref {
 description
 "The reference to TE endpoints.";
 choice te {
 description
 "The TE";
 case actn-vn {
 leaf actn-vn-ref {
 type leafref {
 path "/vn:actn/vn:ap/vn:access-point-list"
 + "/vn:access-point-id";
 }
 description
 "The reference to ACTN VN";
 }
 }
 case te {
 leaf ltp {
 type te-types:te-tp-id;
 description
 "Reference LTP in the TE-topology";
 }
 }
 }
}
```

```
 }

 grouping te-mapping {
 description
 "Mapping between Services and TE";
 container te-mapping {
 description
 "Mapping between Services and TE";
 leaf map-type {
 type identityref {
 base map-type;
 }
 description
 "Isolation Requirements, Tunnel Bind or
 Tunnel Selection";
 }
 leaf availability-type {
 type identityref {
 base availability-type;
 }
 description
 "Availability Requirement for the Service";
 }
 uses te-ref;
 }
 }
 }
}
<CODE ENDS>
```

<CODE BEGINS> file "ietf-llcsm-te-service-mapping@2018-10-05.yang"

```
module ietf-llcsm-te-service-mapping {

 namespace "urn:ietf:params:xml:ns:yang:ietf-llcsm-te-service-mapping";

 prefix "tm";

 import ietf-te-service-mapping-types {
 prefix "tsm-types";
 }

 import ietf-llcsm {
 prefix "ll";
 }
}
```

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

contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Dhruv Dhody <dhruv.ietf@gmail.com>
 Qin Wu <bill.wu@huawei.com>";
description
 "This module contains a YANG module for the mapping of
 Layer 1 Connectivity Service Module (L1CSM) to the TE and VN ";

revision 2018-10-05 {
 description
 "initial version.";
 reference
 "TBD";
}

/*
 * Configuration data nodes
 */
augment "/l1:l1-connectivity/l1:services/l1:service" {
 description
 "l1csm augmented to include TE parameters and mapping";
 container te-service-mapping {
 presence "indicates l1 service to te mapping";
 description
 "Container to augment l1csm to TE parameters and mapping";
 }
}

augment "/l1:l1-connectivity/l1:services/l1:service" {
 description
 "This augment is only valid for TE mapping --
 te mapping is added";
 uses tsm-types:te-mapping;
}

augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1" {
 description
 "This augment is only valid for TE mapping --
 endpoint-1 te-reference is added";
 uses tsm-types:te-endpoint-ref;
}

augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2" {
 description
```

```
 "This augment is only valid for TE mapping --
 endpoint-2 te-reference is added";
 uses tsm-types:te-endpoint-ref;
}
}
```

<CODE ENDS>

<CODE BEGINS> file "ietf-l2sm-te-service-mapping@2018-10-05.yang"

```
module ietf-l2sm-te-service-mapping {

 namespace "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";

 prefix "tm";

 import ietf-te-service-mapping-types {
 prefix "tsm-types";
 }

 import ietf-l2vpn-svc {
 prefix "l2vpn-svc";
 }

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

 contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Dhruv Dhody <dhruv.ietf@gmail.com>
 Qin Wu <bill.wu@huawei.com>";
 description
 "This module contains a YANG module for the mapping of
 Layer 2 Service Model (L1CSM) to the TE and VN ";

 revision 2018-10-05 {
 description
 "initial version.";
 reference
 "TBD";
 }
}
```

```
 /*
 * Configuration data nodes
 */
 augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
 description
 "l2sm augmented to include TE parameters and mapping";
 container te-service-mapping {
 presence "indicates l2 service to te mapping";
 description
 "Container to augment l2sm to TE parameters and mapping";
 }
}

 augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
 description
 "This augment is only valid for TE mapping --
 te mapping is added";
 uses tsm-types:te-mapping;
}

 augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
 +"/l2vpn-svc:site-network-accesses/l2vpn-svc:site-network-access" {
 description
 "This augment is only valid for TE mapping --
 network-access te-reference is added";
 uses tsm-types:te-endpoint-ref;
}
}
```

<CODE ENDS>

<CODE BEGINS> file "ietf-l3sm-te-service-mapping@2018-10-05.yang"

```
module ietf-l3sm-te-service-mapping {

 namespace "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";

 prefix "tm";

 import ietf-te-service-mapping-types {
 prefix "tsm-types";
 }
}
```

```

 }

 import ietf-l3vpn-svc {
 prefix "l3vpn-svc";
 }

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

 contact
 "Editor: Young Lee <leeyoung@huawei.com>
 Dhruv Dhody <dhruv.ietf@gmail.com>
 Qin Wu <bill.wu@huawei.com>";

 description
 "This module contains a YANG module for the mapping of
 Layer 3 Service Model (L3SM) to the TE and VN ";

 revision 2018-10-05 {
 description
 "initial version.";
 reference
 "TBD";
 }

 /*
 * Configuration data nodes
 */
 augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
 description
 "l3sm augmented to include TE parameters and mapping";
 container te-service-mapping {
 presence "indicates l3 service to te mapping";
 description
 "Container to augment l3sm to TE parameters and mapping";
 }
 }

 augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
 description
 "This augment is only valid for TE mapping --
 te mapping is added";
 uses tsm-types:te-mapping;
 }

 augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
 +"/l3vpn-svc:site-network-accesses/l3vpn-svc:site-network-access" {

```

```
 description
 "This augment is only valid for TE mapping --
 network-access te-reference is added";
 uses tsm-types:te-endpoint-ref;
 }
}
```

<CODE ENDS>

## 8. Security

The configuration, state, and action data defined in this document are designed to be accessed via a management protocol with a secure transport layer, such as NETCONF [RFC6241]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true") These data nodes may be considered sensitive or vulnerable in some network environments.

## 9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

```

URI: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-llcsm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

```

```

URI: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
Registrant Contact: The IESG.

```

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

---

---

URI: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping  
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 [RFC7950]:

---

name: ietf-te-service-mapping-types  
namespace: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types  
reference: RFC XXXX (TDB)

---

---

name: ietf-l1csm-te-service-mapping  
namespace: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping  
reference: RFC XXXX (TDB)

---

---

name: ietf-l2sm-te-service-mapping  
namespace: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping  
reference: RFC XXXX (TDB)

---

---

name: ietf-l3sm-te-service-mapping  
namespace: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping  
reference: RFC XXXX (TDB)

---



## 10. Acknowledgements

We thank Diego Caviglia and Igor Bryskin for useful discussions and motivation for this work.

## 11. References

### 11.1. Informative References

- [RFC4110] R. Callon and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.
- [RFC6020] M. Bjorklund, Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC8309] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", RFC 8309, January 2018.
- [RFC8199] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", RFC 8199, July 2017.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.
- [RFC8453] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.
- [TE-Topo] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [TE-Types] T. Saad (Editor), "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types, work in progress.
- [ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

- [ACTN-Applicability] Y. Lee, et al, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks, draft-ietf-teas-actn-yang, work in progress.
- [RFC8299] Q. Wu, S. Litkowski, L.Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN service delivery", RFC 8299, January 2018.
- [L2SM] B. Wen, et al, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.
- [L1CSM] G. Fioccola, et al, "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang, work in progress.

## 12. Contributors

Adrian Farrel  
Old Dog Consulting

Email: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

Italo Busi  
Huawei Technologies

Email: [Italo.Busi@huawei.com](mailto:Italo.Busi@huawei.com)

## Authors' Addresses

Young Lee  
Huawei Technologies  
5340 Legacy Drive  
Plano, TX 75023, USA  
Phone: (469)277-5838

Email: [leeyoung@huawei.com](mailto:leeyoung@huawei.com)

Dhruv Dhody  
Huawei Technologies

Email: [dhruv.ietf@gmail.com](mailto:dhruv.ietf@gmail.com)

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura  
Apstra

Email: jefftant@gmail.com

Giuseppe Fioccola  
Huawei  
Email: giuseppe.fioccola@huawei.com

Qin Wu  
Huawei  
Email: bill.wu@huawei.com



TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 21, 2017

H. Sitaraman, Ed.  
V. Beeram  
Juniper Networks  
I. Minei  
Google, Inc.  
S. Sivabalan  
Cisco Systems, Inc.  
February 17, 2017

Recommendations for RSVP-TE and Segment Routing LSP co-existence  
draft-sitaraman-sr-rsvp-coexistence-rec-02.txt

## Abstract

Operators are looking to introduce services over Segment Routing (SR) LSPs in networks running Resource Reservation Protocol (RSVP-TE) LSPs. In some instances, operators are also migrating existing services from RSVP-TE to SR LSPs. For example, there might be certain services that are well suited for SR and need to co-exist with RSVP-TE in the same network. In other cases, services running on RSVP-TE might be migrated to run over SR. Such introduction or migration of traffic to SR might require co-existence with RSVP-TE in the same network for an extended period of time depending on the operator's intent. The following document provides solution options for keeping the traffic engineering database (TED) consistent across the network, accounting for the different bandwidth utilization between SR and RSVP-TE.

## 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 August 21, 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 |
| 3. Solution options . . . . .                               | 3 |
| 3.1. Static partitioning of bandwidth . . . . .             | 3 |
| 3.2. Centralized management of available capacity . . . . . | 4 |
| 3.3. Flooding SR utilization in IGP . . . . .               | 4 |
| 3.4. Running SR over RSVP-TE . . . . .                      | 5 |
| 3.5. TED consistency by reflecting SR traffic . . . . .     | 5 |
| 4. Acknowledgements . . . . .                               | 7 |
| 5. Contributors . . . . .                                   | 7 |
| 6. IANA Considerations . . . . .                            | 8 |
| 7. Security Considerations . . . . .                        | 8 |
| 8. References . . . . .                                     | 8 |
| 8.1. Normative References . . . . .                         | 8 |
| 8.2. Informative References . . . . .                       | 8 |
| Authors' Addresses . . . . .                                | 9 |

## 1. Introduction

Introduction of SR [I-D.ietf-spring-segment-routing] in the same network domain as RSVP-TE [RFC3209] presents the problem of accounting for SR traffic and making RSVP-TE aware of the actual available bandwidth on the network links. RSVP-TE is not aware of how much bandwidth is being consumed by SR services on the network links and hence both at computation time (for a distributed computation) and at signaling time RSVP-TE LSPs will incorrectly place loads. This is true where RSVP-TE paths are distributed or centrally computed without a common entity managing both SR and RSVP-TE computation for the entire network domain.

The problem space can be generalized as a dark bandwidth problem to cases where any other service exists in the network that runs in parallel across common links and whose bandwidth is not reflected in the available and reserved values in the TED. The general problem is management of dark bandwidth pools and can be generalized to cases where any other service exists in the network that runs in parallel across common links and whose bandwidth is not reflected in the available and reserved values in the TED. In most practical instances given the static nature of the traffic demands, limiting the available reservable bandwidth available to RSVP-TE has been an acceptable solution. However, in the case of SR traffic, there is assumed to be very dynamic traffic demands and there is considerable risk associated with stranding capacity or overbooking service traffic resulting in traffic drops.

The high level requirements or assumptions to consider are:

1. Placement of SR LSPs in the same domain as RSVP-TE LSPs MUST NOT introduce inaccuracies in the TED used by distributed or centralized path computation engines.
2. Engines that compute RSVP-TE paths MAY have no knowledge of the existence of the SR paths in the same domain.
3. Engines that compute RSVP-TE paths SHOULD NOT require a software upgrade or change to their path computation logic.
4. Protocol extensions SHOULD be avoided or be minimal as in many cases this co-existence of RSVP-TE and SR MAY be needed only during a transition phase.
5. Placement of SR LSPs in the same domain as RSVP-TE LSPs that are computed in a distributed fashion MUST NOT require migration to a central controller architecture for the RSVP-TE LSPs.

## 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. Solution options

### 3.1. Static partitioning of bandwidth

In this model, the static reservable bandwidth of an interface can be statically partitioned between SR and RSVP-TE and each can operate within that bandwidth allocation and SHOULD NOT preempt each other.

While it is possible to configure RSVP-TE to only reserve up to a certain maximum link bandwidth and manage the remaining link bandwidth for other services, this is a deployment where SR and RSVP-TE are separated in the same network (ships in the night) and can lead to suboptimal link bandwidth utilization not allowing each to consume more, if required and constraining the respective deployments.

The downside of this approach is the inability to use the reservable bandwidth effectively and inability to use bandwidth left unused by the other protocol.

### 3.2. Centralized management of available capacity

In this model, a central controller performs path placement for both RSVP-TE and SR LSPs. The controller manages and updates its own view of the in-use and the available capacity. As the controller is a single common entity managing the network it can have a unified and consistent view of the available capacity at all times.

A practical drawback of this model is that it requires the introduction of a central controller managing the RSVP-TE LSPs as a prerequisite to the deployment of any SR LSPs. Therefore, this approach is not practical for networks where distributed TE with RSVP-TE LSPs is already deployed, as it requires a redesign of the network and is not backwards compatible. This does not satisfy requirement 5.

Note that it is not enough for the controller to just maintain the unified view of the available capacity, it must also perform the path computation for the RSVP-TE LSPs, as the reservations for the SR LSPs are not reflected in the TED. This does not fit with assumption 2 mentioned earlier.

### 3.3. Flooding SR utilization in IGP

Using techniques in [RFC7810], [RFC7471] and [RFC7823], the SR utilization information can be flooded in IGP-TE and the RSVP-TE path computation engine (CSPF) can be changed to consider this information. This requires changes to the RSVP-TE path computation logic and would require upgrades in deployments where distributed computation is done across the network.

This does not fit with requirements 3 and 4 mentioned earlier.



### 3.4. Running SR over RSVP-TE

SR can run over dedicated RSVP-TE LSPs that carry only SR traffic. In this model, the LSPs can be one-hop or multi-hop and can provide bandwidth reservation for the SR traffic based on functionality such as auto-bandwidth. The model of deployment would be similar in nature to running LDP over RSVP-TE. This would allow the TED to stay consistent across the network and any other RSVP-TE LSPs will also be aware of the SR traffic reservations. In this approach, non-SR traffic MUST NOT take the SR-dedicated RSVP-TE LSPs, unless required by policy.

The drawback of this solution is that it requires SR to rely on RSVP-TE for deployment. Furthermore, the accounting accuracy/frequency of this method is dependent on performance of auto-bandwidth for RSVP-TE. Note that for this method to work, the SR-dedicated RSVP-TE LSPs must be set up with the best setup and hold priorities in the network.

### 3.5. TED consistency by reflecting SR traffic

The solution relies on dynamically measuring SR traffic utilization on each TE interface and reducing the bandwidth allowed for use by RSVP-TE. It is assumed that SR traffic receives precedence in terms of the placement on the path over RSVP traffic (that is, RSVP traffic can be preempted from the path in case of insufficient resources). This is logically equivalent to SR traffic having the best preemption priority in the network. Note that this does not necessarily mean that SR traffic has higher QoS priority, in fact, SR and RSVP traffic may be in the same QoS class. The following methodology can be used at every TE node for this solution:

- o T: Traffic statistics collection time interval
- o N: Traffic averaging calculation (adjustment) interval such that  $N = k * T$ , where k is a constant integer multiplier greater or equal to 1. Its purpose is to provide a smoothing function to the statistics collection.
- o Maximum-Reservable-Bandwidth: The maximum available bandwidth for TE (this is the maximum available bandwidth on the interface, before any LSP reservations).

If Differentiated-Service (Diffserv)-aware MPLS Traffic Engineering (DS-TE) [RFC4124] is enabled, the Maximum-Reservable-Bandwidth SHOULD be interpreted as the aggregate bandwidth constraint across all Class-Types independent of the Bandwidth Constraints model.

- o RSVP-unreserved-bandwidth-at-priority-X: Maximum-Reservable-Bandwidth - sum of (existing reservations at priority X and all priorities better than X)
- o SR traffic threshold percentage: The percentage difference of traffic demand that when exceeded can result in a change to the RSVP-TE Maximum-Reservable-Bandwidth
- o IGP-TE update threshold: Specifies the frequency at which IGP-TE updates should be triggered based on TE bandwidth updates on a link
- o M: An optional multiplier that can be applied to the SR traffic average. This multiplier provides the ability to grow or shrink the bandwidth used by SR

At every interval T, each node SHOULD collect the SR traffic statistics for each of its TE interfaces. Further, at every interval N, given a configured SR traffic threshold percentage and a set of collected SR traffic statistics samples across the interval N, the SR traffic average (or any other traffic metric depending on the algorithm used) over this period is calculated.

If the difference between the new calculated SR traffic average and the current SR traffic average (that was computed in the prior adjustment) is at least SR traffic threshold percentage, then two values MUST be updated:

- o New Maximum-Reservable-Bandwidth = Current Maximum-Reservable-Bandwidth - (SR traffic average \* M)
- o New RSVP-unreserved-bandwidth-at-priority-X = New Maximum-Reservable-Bandwidth - sum of (existing reservations at priority X and all priorities better than X)

A DS-TE LSR that advertises Bandwidth Constraints TLV should update the bandwidth constraints for class-types based on operator policy. For example, when Russian Dolls Model (RDM) [RFC4127] is in use, then only BC0 may be updated. Whereas, when Maximum Allocation Model (MAM) [RFC4125] is in use, then all BCs may be updated equally such that the total value updated is equal to the newly calculated SR traffic average.

Note that the computation of the new RSVP-unreserved-bandwidth-at-priority-X MAY result in RSVP-TE LSPs being hard or soft preempted. Such preemption will be based on relative priority (e.g. low to high) between RSVP-TE LSPs. It is RECOMMENDED that the IGP-TE update threshold SHOULD be lower in order to flood unreserved bandwidth

updates often. From an operational point of view, an implementation SHOULD be able to expose both the configured and the actual values of the Maximum-Reservable-Bandwidth.

If LSP preemption is not acceptable, then the RSVP-TE Maximum-Reservable-Bandwidth cannot be reduced below what is currently reserved by RSVP-TE on that interface. This may result in bandwidth not being available for SR traffic. Thus, it is required that any external controller managing SR LSPs SHOULD be able to detect this situation (for example by subscribing to TED updates [RFC7752]) and SHOULD take action to reroute existing SR paths.

Generically, SR traffic (or any non-RSVP-TE traffic) should have its own priority allocated from the available priorities. This would allow SR to preempt other traffic according to the preemption priority order.

In this solution, the logic to retrieve the statistics, calculating averages and taking action to change the Maximum-Reservable-Bandwidth is an implementation choice, and all changes are local in nature. However, note that this is a new network trigger for RSVP-TE preemption and thus is a consideration for the operator.

The above solution offers the advantage of not introducing new network-wide mechanisms especially during scenarios of migrating to SR in an existing RSVP-TE network and reusing existing protocol mechanisms.

#### 4. Acknowledgements

The authors would like to thank Steve Ulrich for his detailed review and comments.

#### 5. Contributors

The following individuals contributed to this document:

Chandra Ramachandran  
Juniper Networks  
Email: csekar@juniper.net

Raveendra Torvi  
Juniper Networks  
Email: rtorvi@juniper.net

Sudharsana Venkataraman  
Juniper Networks  
Email: sudharsana@juniper.net

## 6. IANA Considerations

This draft does not have any request for IANA.

## 7. Security Considerations

No new security issues are introduced in this document beyond is already part of RSVP-TE and Segment routing architectures.

## 8. References

### 8.1. Normative 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.
- [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>>.
- [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>>.

### 8.2. Informative References

- [RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of  
Diffserv-aware MPLS Traffic Engineering", RFC 4124,  
DOI 10.17487/RFC4124, June 2005,  
<<http://www.rfc-editor.org/info/rfc4124>>.
- [RFC4125] Le Faucheur, F. and W. Lai, "Maximum Allocation Bandwidth  
Constraints Model for Diffserv-aware MPLS Traffic  
Engineering", RFC 4125, DOI 10.17487/RFC4125, June 2005,  
<<http://www.rfc-editor.org/info/rfc4125>>.
- [RFC4127] Le Faucheur, F., Ed., "Russian Dolls Bandwidth Constraints  
Model for Diffserv-aware MPLS Traffic Engineering",  
RFC 4127, DOI 10.17487/RFC4127, June 2005,  
<<http://www.rfc-editor.org/info/rfc4127>>.

- [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, <<http://www.rfc-editor.org/info/rfc7471>>.
- [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, <<http://www.rfc-editor.org/info/rfc7752>>.
- [RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <<http://www.rfc-editor.org/info/rfc7810>>.
- [RFC7823] Atlas, A., Drake, J., Giacalone, S., and S. Previdi, "Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions", RFC 7823, DOI 10.17487/RFC7823, May 2016, <<http://www.rfc-editor.org/info/rfc7823>>.

#### Authors' Addresses

Harish Sitaraman (editor)  
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)

Ina Minei  
Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94043  
US

Email: inaminei@google.com

Siva Sivabalan  
Cisco Systems, Inc.  
2000 Innovation Drive  
Kanata, Ontario K2K 3E8  
Canada

Email: msiva@cisco.com

TEAS Working Group  
Internet Draft

A.Wang  
China Telecom  
Quintin Zhao  
Boris Khasanov  
HuaiMo Chen  
Huawei Technologies  
Penghui Mi  
Tencent Company

Intended status: Experimental Track  
Expires: July 24, 2018

January 25, 2018

PCE in Native IP Network  
draft-wang-teas-pce-native-ip-07.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). 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 24, 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 carefully, as they describe your rights and restrictions with respect to this document.

#### Abstract

This document defines the framework for CCDR traffic engineering within Native IP network, using Dual/Multi-BGP session strategy and PCE-based central control architecture.

<A.Wang>

Expires July 24, 2018

[Page 1]

Internet-Draft                      PCE in Native IP Network                      January 25, 2017  
The proposed central mode control framework conforms to the concept  
that defined in RFC " An Architecture for Use of PCE and the PCE  
Communication Protocol (PCEP) in a Network with Central Control".

The scenario and simulation results of CCDD traffic engineering is  
described in draft "CCDD Scenario, Simulation and Suggestion".

## Table of Contents

|                                                                |    |
|----------------------------------------------------------------|----|
| 1. Introduction .....                                          | 2  |
| 2. Dual-BGP framework for simple topology. ....                | 3  |
| 3. Dual-BGP in large Scale Topology .....                      | 4  |
| 4. Multi-BGP for Extended Traffic Differentiation .....        | 5  |
| 5. CCDD based framework for Multi-BGP strategy deployment..... | 6  |
| 6. PCEP extension for key parameters delivery. ....            | 7  |
| 7. CCDD Deployment Consideration .....                         | 7  |
| 8. Security Considerations.....                                | 8  |
| 9. IANA Considerations .....                                   | 8  |
| 10. Conclusions .....                                          | 8  |
| 11. References .....                                           | 9  |
| 11.1. Normative References.....                                | 9  |
| 11.2. Informative References.....                              | 9  |
| 12. Acknowledgments .....                                      | 10 |

## 1. Introduction

Draft [I-D.draft-wang-teas-ccdd] describes the scenario and simulation  
results for the CCDD traffic engineering. In summary, the requirements for  
CCDD traffic engineering in Native IP network are the following:

- 1) No complex MPLS signaling procedure.
- 2) End to End traffic assurance, determined QoS behavior.
- 3) Identical deployment method for intra- and inter- domain.
- 4) No influence to existing router forward behavior.
- 5) Can utilize the power of centrally control(PCE) and  
flexibility/robustness of distributed control protocol.
- 6) Coping with the differentiation requirements for large amount  
traffic and prefixes.
- 7) Flexible deployment and automation control.

This document defines the framework for CCDD traffic engineering  
within Native IP network, using Dual/Multi-BGP session strategy and  
CCDD architecture, to meet the above requirements in dynamical and  
central control mode. Future PCEP protocol extensions to transfer the  
key parameters between PCE and the underlying network devices(PCC)  
are provided in draft [draft-wang-pcep-extension-native-IP]



## 2. Dual-BGP framework for simple topology.

Dual-BGP framework for simple topology is illustrated in Fig.1, which is comprised by SW1, SW2, R1, R2. There are multiple physical links between R1 and R2. Traffic between IP11 and IP21 is normal traffic, traffic between IP12 and IP22 is priority traffic that should be treated differently.

Only Native IGP/BGP protocol is deployed between R1 and R2. The traffic between each address pair may change timely and the corresponding source/destination addresses of the traffic may also change dynamically.

The key idea of the Dual-BGP framework for this simple topology is the following:

- 1) Build two BGP sessions between R1 and R2, via the different loopback address lo0, lo1 on these routers.
- 2) Send different prefixes via the two BGP sessions. (For example, IP11/IP21 via the BGP pair 1 and IP12/IP22 via the BGP pair 2).
- 3) Set the explicit peer route on R1 and R2 respectively for BGP next hop of lo0, lo1 to different physical link address between R1 and R2.

So, the traffic between the IP11 and IP21, and the traffic between IP12 and IP22 will go through different physical links between R1 and R2, each type of traffic occupy the different dedicated physical links.

If there is more traffic between IP12 and IP22 that needs to be assured, one can add more physical links on R1 and R2 to reach the loopback address lo1(also the next hop for BGP Peer pair2). In this cases the prefixes that advertised by two BGP peer need not be changed.

If, for example, there is traffic from another address pair that needs to be assured (for example IP13/IP23), but the total volume of assured traffic does not exceed the capacity of the previous appointed physical links, then one need only to advertise the newly added source/destination prefixes via the BGP peer pair2, then the traffic between IP13/IP23 will go through the assigned dedicated physical links as the traffic between IP12/IP22.

Such decouple philosophy gives the network operator more flexible control ability on the network traffic, get the determined QoS assurance effect to meet the application's requirement. No complex MPLS signal procedures is introduced, the router need only support native IP protocol.

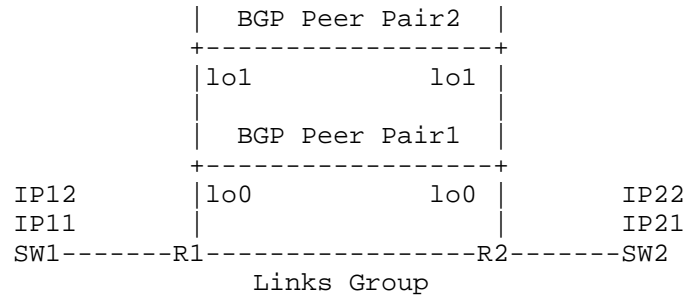
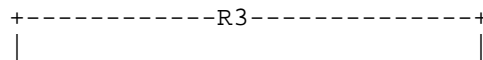


Fig.1 Design Philosophy for Dual-BGP Framework

### 3. Dual-BGP in large Scale Topology

When the assured traffic spans across one large scale network, as that illustrated in Fig.2, the dual BGP sessions cannot be established hop by hop especially for the iBGP within one AS. For such scenario, we should consider to use the Route Reflector (RR) to achieve the similar Dual-BGP effect, select one router which performs the role of RR (for example R3 in Fig.2), every other edge router will establish two BGP peer sessions with the RR, using their different loopback addresses respectively. The other two steps for traffic differentiation are same as one described in the Dual-BGP simple topology usage case.

For the example shown in Fig.2, if we select the R1-R2-R4-R7 as the dedicated path, then we should set the explicit peer routes on these routers respectively, pointing to the BGP next hop (loopback addresses of R1 and R7, which are used to send the prefix of the assured traffic) to the actual address of the physical link



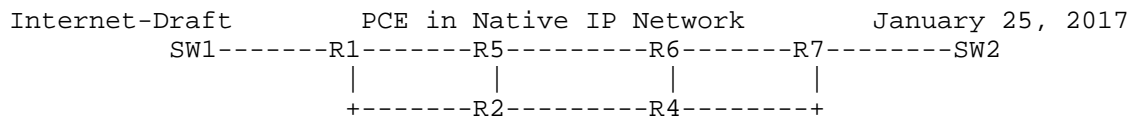


Fig.2 Dual-BGP Framework for large scale network

#### 4. Multi-BGP for Extended Traffic Differentiation

In general situation, several additional traffic differentiation criteria exist, including:

- o Traffic that requires low latency links and is not sensitive to packet loss
- o Traffic that requires low packet loss but can endure higher latency
- o Traffic that requires lowest jitter path
- o Traffic that requires high bandwidth links

These different traffic requirements can be summarized in the following table:

| Flow No. | Latency | Packet Loss | Jitter      |
|----------|---------|-------------|-------------|
| 1        | Low     | Normal      | Don't care  |
| 2        | Normal  | Low         | Dont't care |
| 3        | Normal  | Normal      | Low         |

Table 1. Traffic Requirement Criteria

For Flow No.1, we can select the shortest distance path to carry the traffic; for Flow No.2, we can select the idle links to form its end to end path; for Flow No.3, we can let all the traffic pass one single path, no ECMP distribution on the parallel links is required.

It is difficult and almost impossible to provide an end-to-end (E2E) path with latency, latency variation, packet loss, and bandwidth utilization constraints to meet the above requirements in large scale IP-based network via the traditional distributed routing protocol, but these requirements can be solved using the CCDR architecture since the PCE has the overall network view, can collect real network topology and network performance information about the underlying

5. CCDR based framework for Multi-BGP strategy deployment.

With the advent of SDN concepts towards pure IP networks, it is possible now to accomplish the central and dynamic control of network traffic according to the application's various requirements.

The procedure to implement the dynamic deployment of Multi-BGP strategy is the following:

- 1) PCE gets topology and link utilization information from the underlying network, calculate the appropriate link path upon application's requirements.
- 2) PCE sends the key parameters to edge/RR routers(R1, R7 and R3 in Fig.3) to build multi-BGP peer relations and advertise different prefixes via them.
- 3) PCE sends the route information to the routers (R1,R2,R4,R7 in Fig.3) on forwarding path via PCEP, to build the path to the BGP next-hop of the advertised prefixes.
- 4) If the assured traffic prefixes were changed but the total volume of assured traffic does not exceed the physical capacity of the previous end-to-end path, then PCE needs only change the related information on edge routers (R1,R7 in Fig.3).
- 5) If volume of the assured traffic exceeds the capacity of previous calculated path, PCE must recalculate the appropriate path to accommodate the exceeding traffic via some new end-to-end physical link. After that PCE needs to update on-path routers to build such path hop by hop.

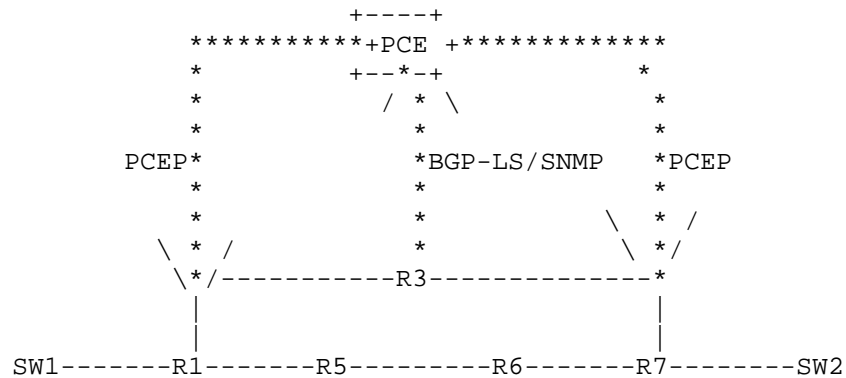




Fig.3 PCE based framework for Multi-BGP deployment

## 6. PCEP extension for key parameters delivery.

The PCEP protocol needs to be extended to transfer the following key parameters:

- 1) BGP peer address and advertised prefixes.
- 2) Explicit route information to BGP next hop of advertised prefixes.

Once the router receives such information, it should establish the BGP session with the peer appointed in the PCEP message, advertise the prefixes that contained in the corresponding PCEP message, and build the end to end dedicated path hop by hop. Details of communications between PCEP and BGP subsystems in router's control plane are out of scope of this draft and will be described in separate draft.[draft-wang-pce-extension for native IP]

The reason why we selected PCEP as the southbound protocol instead of OpenFlow, is that PCEP is suitable for the changes in control plane of the network devices, there OpenFlow dramatically changes the forwarding plane. We also think that the level of centralization that requires by OpenFlow is hardly achievable in many today's SP networks so hybrid BGP+PCEP approach looks much more interesting.

## 7. CCDR Deployment Consideration

CCDR framework requires the parallel work of 2 subsystems in router's control plane: PCE (PCEP) and BGP as well as coordination between them, so it might require additional planning work before deployment.

### 8.1 Scalability

In CCDR framework, PCE needs only to influence the edge routers for the prefixes differentiation via the multi-BGP deployment. The route information for these prefixes within the on-path routers were distributed via the traditional BGP protocol. Unlike the solution from BGP Flowspec, the on-path router need only keep the specific policy routes to the BGP next-hop of the differentiate prefixes, not

Internet-Draft                      PCE in Native IP Network                      January 25, 2017  
the specific routes to the prefixes themselves. This can lessen the burden from the table size of policy based routes for the on-path routers, and has more scalability when comparing with the solution from BGP flowspec or Openflow.

## 8.2 High Availability

CCDR framework is based on the traditional distributed IP protocol. If the PCE failed, the forwarding plane will not be impacted, as the BGP session between all devices will not flap, and the forwarding table will remain the same. If one node on the optimal path is failed, the assurance traffic will fall over to the best-effort forwarding path. One can even design several assurance paths to load balance/hot standby the assurance traffic to meet the path failure situation, as done in MPLS FRR.

From PCE/SDN-controller HA side we will rely on existing HA solutions of SDN controllers such as clustering.

## 8.3 Incremental deployment

Not every router within the network support will support the PCEP extension that defined in [draft-wang-pce-extension-native-IP] simultaneously. For such situations, router on the edge of sub domain can be upgraded first, and then the traffic can be assured between different sub domains. Within each sub domain, the traffic will be forwarded along the best-effort path. Service provider can selectively upgrade the routers on each sub-domain in sequence.

## 8. Security Considerations

TBD

## 9. IANA Considerations

TBD

## 10. Conclusions

TBD

#### 11.1. Normative References

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006, <<http://www.rfc-editor.org/info/rfc4655>>.

[RFC5440] Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009, <<http://www.rfc-editor.org/info/rfc5440>>.

[RFC8283] A.Farrel, Q.Zhao et al., "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", [RFC8283], December 2017

#### 11.2. Informative References

[I-D.draft-wang-teas-ccdr]

A.Wang, X.Huang et al. "CCDR Scenario, Simulation and Suggestion" <https://datatracker.ietf.org/doc/draft-wang-teas-ccdr/>

[I-D. draft-ietf-teas-pcecc-use-cases]

Quintin Zhao, Robin Li, Boris Khasanov et al. "The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs" <https://tools.ietf.org/html/draft-ietf-teas-pcecc-use-cases-00>  
March, 2017

[draft-wang-pcep-extension for native IP]

## 12. Acknowledgments

The authors would like to thank George Swallow, Xia Chen, Jeff Tantsura, Scharf Michael, Daniele Ceccarelli and Dhruv Dhody for their valuable comments and suggestions.

The authors would also like to thank Lou Berger, Adrian Farrel, Vishnu Pavan Beeram, Deborah Brungard and King Daniel for their suggestions to put forward this draft.

## Authors' Addresses

Aijun Wang  
China Telecom  
Beiqijia Town, Changping District  
Beijing, China

Email: wangaj.bri@chinatelecom.cn



Internet-Draft                      PCE in Native IP Network

January 25, 2017

Quintin Zhao  
Huawei Technologies  
125 Nagog Technology Park  
Acton, MA 01719  
USA

Email: quintin.zhao@huawei.com

Boris Khasanov  
Huawei Technologies  
Moskovskiy Prospekt 97A  
St.Petersburg 196084  
Russia

Email: khasanov.boris@huawei.com

Huaimo Chen  
Huawei Technologies  
Boston, MA,  
USA

Email: huaimo.chen@huawei.com

Penghui Mi  
Tencent  
Tencent Building, Kejizhongyi Avenue,  
Hi-techPark, Nanshan District, Shenzhen 518057, P.R.China

Email kevinmi@tencent.com

Raghavendra Mallya  
Juniper Networks  
1133 Innovation Way  
Sunnyvale, California 94089 USA

Email: rmallya@juniper.net

Shaofu Peng  
ZTE Corporation  
No.68 Zijinghua Road, Yuhuatai District  
Nanjing 210012  
China

Email: peng.shaofu@zte.com.cn



TEAS WG

Internet Draft

Intended status: Informational

Expires: December 31, 2017

Young Lee  
Haomian Zheng  
Huawei

Daniel Ceccarrelli  
Ericsson

Bin Yeong Yoon  
ETRI

Oscar Gonzalez de Dios  
Telefonica

Jong Yoon Shin  
SKT

Sergio Belotti  
Nokia

June 30, 2017

Applicability of YANG models for Abstraction and Control of Traffic  
Engineered Networks

draft-zhang-teas-actn-yang-05

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 December 30, 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

Abstraction and Control of TE Networks (ACTN) refers to the set of virtual network operations needed to orchestrate, control and manage large-scale multi-domain TE networks, so as to facilitate network programmability, automation, efficient resource sharing, and end-to-end virtual service aware connectivity and network function virtualization services.

This document explains how the different types of YANG models defined in the Operations and Management Area and in the Routing Area are applicable to the ACTN framework. This document also shows how the ACTN architecture can be satisfied using classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

#### Table of Contents

|                                                                    |   |
|--------------------------------------------------------------------|---|
| 1. Introduction.....                                               | 3 |
| 2. Abstraction and Control of TE Networks (ACTN) Architecture..... | 3 |
| 3. Service Models.....                                             | 5 |
| 4. Service Model Mapping to ACTN.....                              | 6 |

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 4.1. Customer Service Models in the ACTN Architecture (CMI).... | 7  |
| 4.2. Service Delivery Models in ACTN Architecture.....          | 8  |
| 4.3. Network Configuration Models in ACTN Architecture (MPI)... | 8  |
| 4.4. Device Models in ACTN Architecture (SBI).....              | 9  |
| 5. Examples of Using Different Types of YANG Models.....        | 10 |
| 5.1. Simple Connectivity Examples.....                          | 10 |
| 5.2. VN service example.....                                    | 10 |
| 5.3. Data Center-Interconnection Example.....                   | 11 |
| 5.3.1. CMI (CNC-MDSC Interface).....                            | 13 |
| 5.3.2. MPI (MDSC-PNC Interface).....                            | 13 |
| 5.3.3. PDI (PNC-Device interface).....                          | 13 |
| 6. Security.....                                                | 14 |
| 7. Acknowledgements.....                                        | 14 |
| 8. References.....                                              | 14 |
| 8.1. Informative References.....                                | 14 |
| 9. Contributors.....                                            | 16 |
| Authors' Addresses.....                                         | 17 |

## 1. Introduction

Abstraction and Control of TE Networks (ACTN) describes a method for operating a Traffic Engineered (TE) network (such as an MPLS-TE network or a layer 1 transport network) to provide connectivity and virtual network services for customers of the TE network. The services provided can be tuned to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers. More details about ACTN can be found in Section 2.

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modelling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [Netmod-Yang-Model-Classification] and [Service-YANG].

This document shows how the ACTN architecture can be satisfied using classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

## 2. Abstraction and Control of TE Networks (ACTN) Architecture

[ACTN-Requirements] describes the high-level ACTN requirements.  
[ACTN-Frame] describes the architecture model for ACTN including the

entities (Customer Network Controller (CNC), Multi-domain Service Coordinator (MDSC), and Physical Network Controller (PNC)) and their interfaces.

Figure 1 depicts a high-level control and interface architecture for ACTN and is a reproduction of Figure 3 from [ACTN-Frame]. A number of key ACTN interfaces exist for deployment and operation of ACTN-based networks. These are highlighted in Figure 1 (ACTN Interfaces) below:

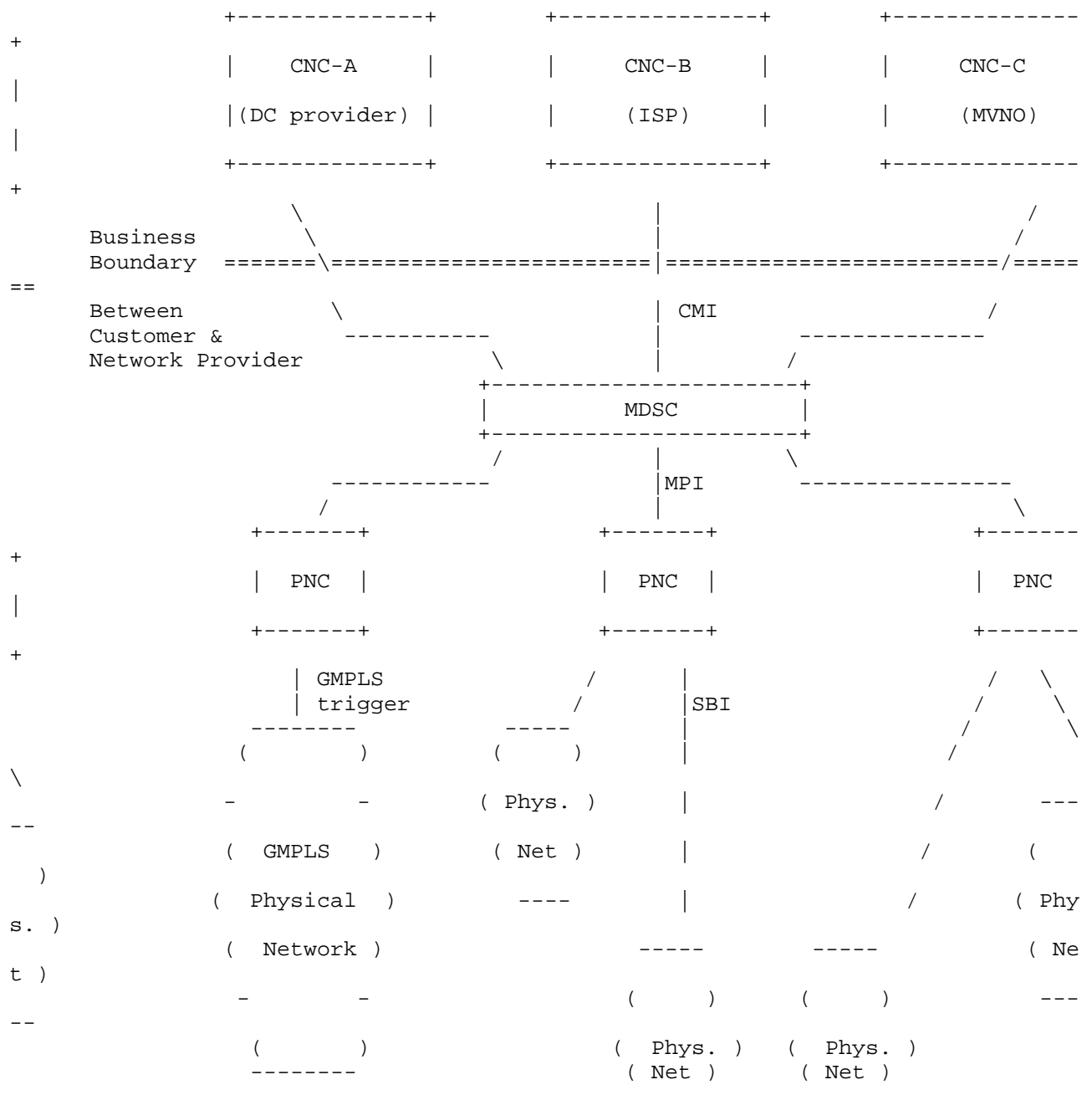


Figure 1 : ACTN Interfaces

The interfaces and functions are described below (without modifying the definitions) in [ACTN-Frame]:



- . The CNC-MDSC Interface (CMI) is an interface between a Customer Network Controller and a Multi Domain Service Controller. The interface will communicate the service request or application demand. A request will include specific service properties, for example, services type, bandwidth and constraint information. These constraints SHOULD be measurable by MDSC and therefore visible to CNC via CMI. The CNC can also request the creation of the virtual network based on underlying physical resources to provide network services for the applications. The CNC can provide the end-point information/characteristics, traffic matrix specifying specific customer constraints. The MDSC may also report potential network topology availability if queried for current capability from the Customer Network Controller.
- . The MDSC-PNC Interface (MPI) is an interface between a Multi Domain Service Coordinator and a Physical Network Controller. It allows the MDSC to communicate requests to create/delete connectivity or to modify bandwidth reservations in the physical network. In multi-domain environments, each PNC is responsible for a separate domain. The MDSC needs to establish multiple MPIs, one for each PNC and perform coordination between them to provide cross-domain connectivity.
- . The South-Bound Interface (SBI) is the provisioning interface for creating forwarding state in the physical network, requested via the Physical Network Controller. The SBI is not in the scope of ACTN, however, it is included in this document so that it can be compared to models in [Service-Yang].

### 3. Service Models

[Service-YANG] introduces a reference architecture to explain the nature and usage of service YANG models in the context of service orchestration. Figure 2 below depicts this relationship and is a reproduction of Figure 2 from [Service-YANG]. Four models depicted in Figure 2 are defined as follows:

- . Customer Service Model: A customer service model is used to describe a service as offer or delivered to a customer by a network operator.
- . Service Delivery Model: A service delivery model is used by a network operator to define and configure how a service is provided by the network.
- . Network Configuration Model: A network configuration model is used by a network orchestrator to provide network-level configuration model to a controller.



- . Device Configuration Model: A device configuration model is used by a controller to configure physical network elements.

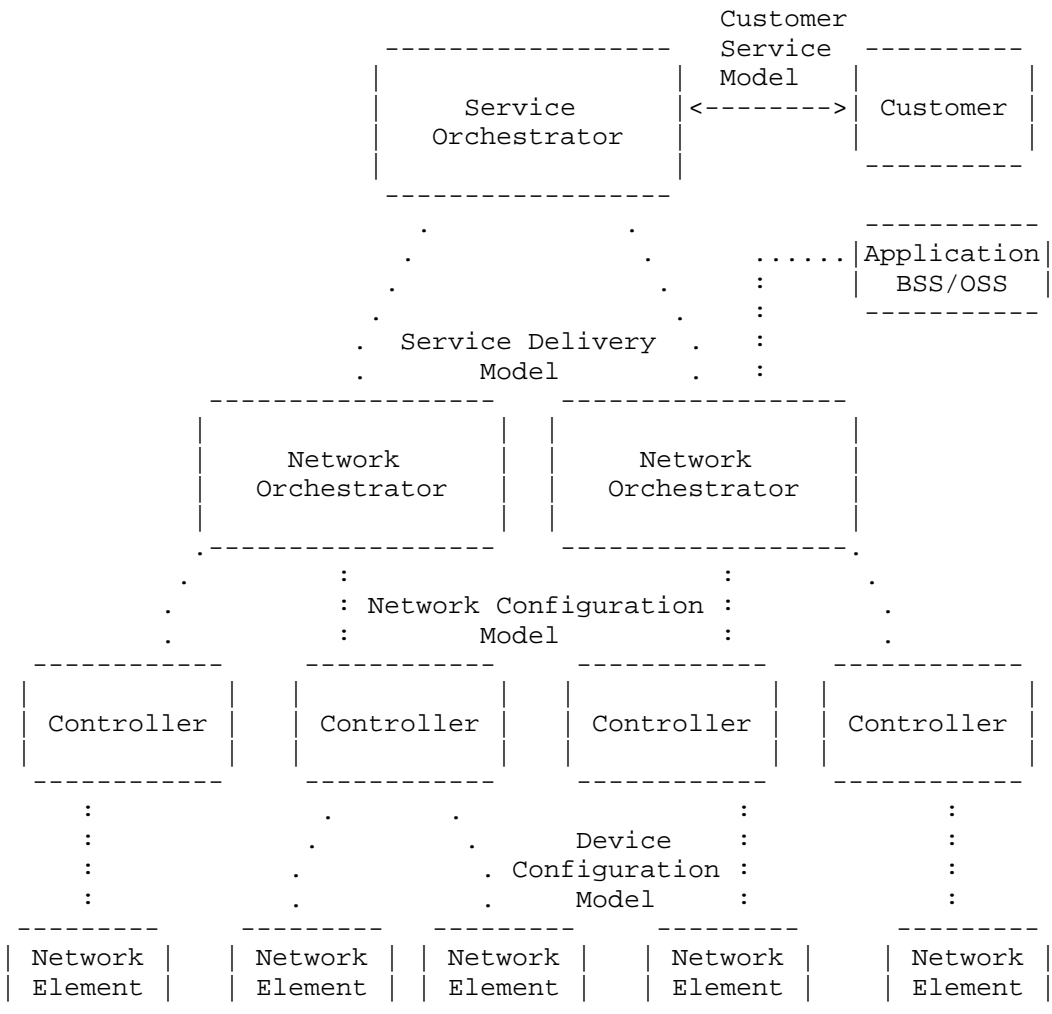


Figure 2: An SDN Architecture with a Service Orchestrator

#### 4. Service Model Mapping to ACTN

YANG models coupled with the RESTCONF/NETCONF protocol [Netconf][Restconf] provides solutions for the ACTN framework. This section explains which types of YANG models apply to each of the ACTN interfaces.

Refer to Figure 5 of [ACTN-Frame] for details of the mapping between ACTN functions and service models. In summary, the following mappings are held between Service Yang Models and the ACTN interfaces.

- o Customer Service Model <-> CMI
- o Network Configuration Model <-> MPI
- o Device Configuration Model <-> SBI

#### 4.1. Customer Service Models in the ACTN Architecture (CMI)

Customer Service Models, which are used between a customer and a service orchestrator as in [Service-YANG], should be used between the CNC and MDSC (e.g., CMI) serving as providing a simple intent-like model/interface.

Among the key functions of Customer Service Models on the CMI is the service request. A request will include specific service properties, including: service type and its characteristics, bandwidth, constraint information, and end-point characteristics.

The following table provides a list of functions needed to build the CMI. They are mapped with Customer Service Models.

| Function                            | Yang Model                |
|-------------------------------------|---------------------------|
| Transport Service Request           | [Transport-Service-Model] |
| VN Service Request & Instantiation  | [ACTN-VN-YANG]            |
| VN Path Computation Request         | [ACTN-VN-YANG]*           |
| VN Performance Monitoring Telemetry | [ACTN-PM-Telemetry]**     |
| Topology Abstraction                | [TE-topology]             |

\*VN Path computation request in the CMI context means network path computation request based on customer service connectivity request constraints prior to the instantiation of a VN creation.

\*\*ietf-actn-te-kpi-telemetry model describes performance telemetry for ACTN VN model. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters. Moreover, this module also provides mechanism to define

aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters.

#### 4.2. Service Delivery Models in ACTN Architecture

The Service Delivery Models where the service orchestration and the network orchestration could be implemented as separate components as seen in [Service-YANG]. This is also known as Network Service Models. On the other hand, from an ACTN architecture point of view, the service delivery model between the service orchestrator and the network orchestrator is an internal interface between sub-components of the MDSC in a single MDSC model.

In the MDSC hierarchical model where there are multiple MDSCs, the interface between the top MDSC and the bottom MDSC can be mapped to service delivery models.

#### 4.3. Network Configuration Models in ACTN Architecture (MPI)

The Network Configuration Models is used between the network orchestrator and the controller in [Service-YANG]. In ACTN, this model is used primarily between a MDSC and a PNC. The Network Configuration Model can be also used for the foundation of more advanced models, like hierarchical MDSCs (see Section 4.5)

The Network Configuration Model captures the parameters which are network wide information.

The following table provides a list of functions needed to build the MPI. They are mapped with Network Configuration Yang Models. Note that various Yang models are work in progress.

| Function                        | Yang Model              |
|---------------------------------|-------------------------|
| -----                           | -----                   |
| Configuration Scheduling        | [Schedule]              |
| Path computation                | [PATH_COMPUTATION-API]* |
| Path Provisioning               | [TE-Tunnel]**           |
| Topology Abstraction            | [TE-topology]           |
| Tunnel PM Telemetry             | [ACTN-PM-Telemetry]***  |
| Service Provisioning            | TBD****                 |
| OTN Topology Abstraction        | [OTN-YANG]              |
| WSON Topology Abstraction       | [WSON-YANG]             |
| Flexi-grid Topology Abstraction | [Flexi-YANG]            |
| ODU Tunnel Model                | [ODU-Tunnel]            |

|                         |                    |
|-------------------------|--------------------|
| WSON TE Tunnel Model    | [WSON-Tunnel]      |
| Flexi-grid Tunnel Model | [Flexigrid-Tunnel] |

\* Related draft is presenting use cases for path computation API, and Yang related model is foreseen to be added.

\*\* Note that path provisioning function is provided by ietf-te module in [TE-Tunnel].

\*\* ietf-actn-te-kpi-telemetry model describes performance telemetry for TE-tunnel model. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data.

\*\*\*\* This function needs to be investigated further. This can be a part of [TE-Tunnel] which is to be determined. Service provisioning is an optional function that builds on top the path provisioning one.

Path provisioning and Topology abstraction functions are mandatory in any case, while Path Computation may be mandatory or optional depending on the type of topology abstraction used. Details of this topic are discussed in [ACTN-Abstraction].

Telemetry may also be an optional function.

#### 4.4. Device Models in ACTN Architecture (SBI)

For the device YANG models are used for per-device configuration purpose, they can be used between the PNC and the physical network/devices. Note that SBI is not in the scope of ACTN. This section is provided to give some examples of YANG-based Device Models. An example of Device Models is ietf-te-device yang module defined in [TE-tunnel].

## 5. Examples of Using Different Types of YANG Models

### 5.1. Simple Connectivity Examples

The data model in [Transport-Service-Model] provides an intent-like connectivity service model which can be used in connection-oriented networks.

It would be used as follows in the ACTN architecture:

- . A CNC uses this service model to specify the two client nodes that are to be connected, and also indicates the amount of traffic (i.e., the bandwidth required) and payload type. What may be additionally specified is the SLA that describes the required quality and resilience of the service.
- . The MDSC uses the information in the request to pick the right network (domain) and also to select the provider edge nodes corresponding to the customer edge nodes.

If there are multiple domains, then the MDSC needs to coordinate across domains to set up network tunnels to deliver a service. Thus coordination includes, but is not limited to, picking the right domain sequence to deliver a service. Before it can perform such functions, it needs to get the topology information from each PNC, using topology YANG models such as [te-topology]. The topology reported from PNC to MDSC can either be abstract or non-abstract.

Additionally, an MDSC can initiate the creation of a tunnel (or tunnel segment) in order to fulfill the service request from CNC based on path computation upon the overall topology information it synthesized from different PNCs. The based model that can cater this purpose is the te-tunnel model specified in [te-tunnel].

- . Then, the PNC needs to decide the explicit route of such a tunnel or tunnel segment (in case of multiple domains), and create such a tunnel using protocols such as PCEP and RSVP-TE or using per-hop configuration.

### 5.2. VN service example

The service model defined in [ACTN-VN-YANG] describes a virtual network (VN) as a service which is a set of multiple connectivity services:

- . A CNC will request VN to the MDSC by specifying a list of VN members. Each VN member specifies either a single connectivity service, or a source with multiple potential destination points in the case that the precise destination sites are to be determined by MDSC.
  - o In the first case, the procedure is the same as the connectivity service, except that in this case, there is a list of connections requested.
  - o In the second case, where the CNC requests the MDSC to select the right destination out of a list of candidates, the MDSC needs to choose the best candidate and reply with the chosen destination for a given VN member. After this is selected, the connectivity request setup procedure is the same as in the connectivity-as-a-service example.

After the VN is set up, a successful reply message is sent from MDSC to CNC, indicating the VN is ready. This message can also be achieved by using the model defined in [ACTN-VN-YANG].

### 5.3. Data Center-Interconnection Example

This section describes more concretely how existing YANG models described in Section 4 map to an ACTN data center interconnection use case. Figure 3 shows a use-case which shows service policy-driven Data Center selection and is a reproduction of Figure A.1 from [ACTN-Info].

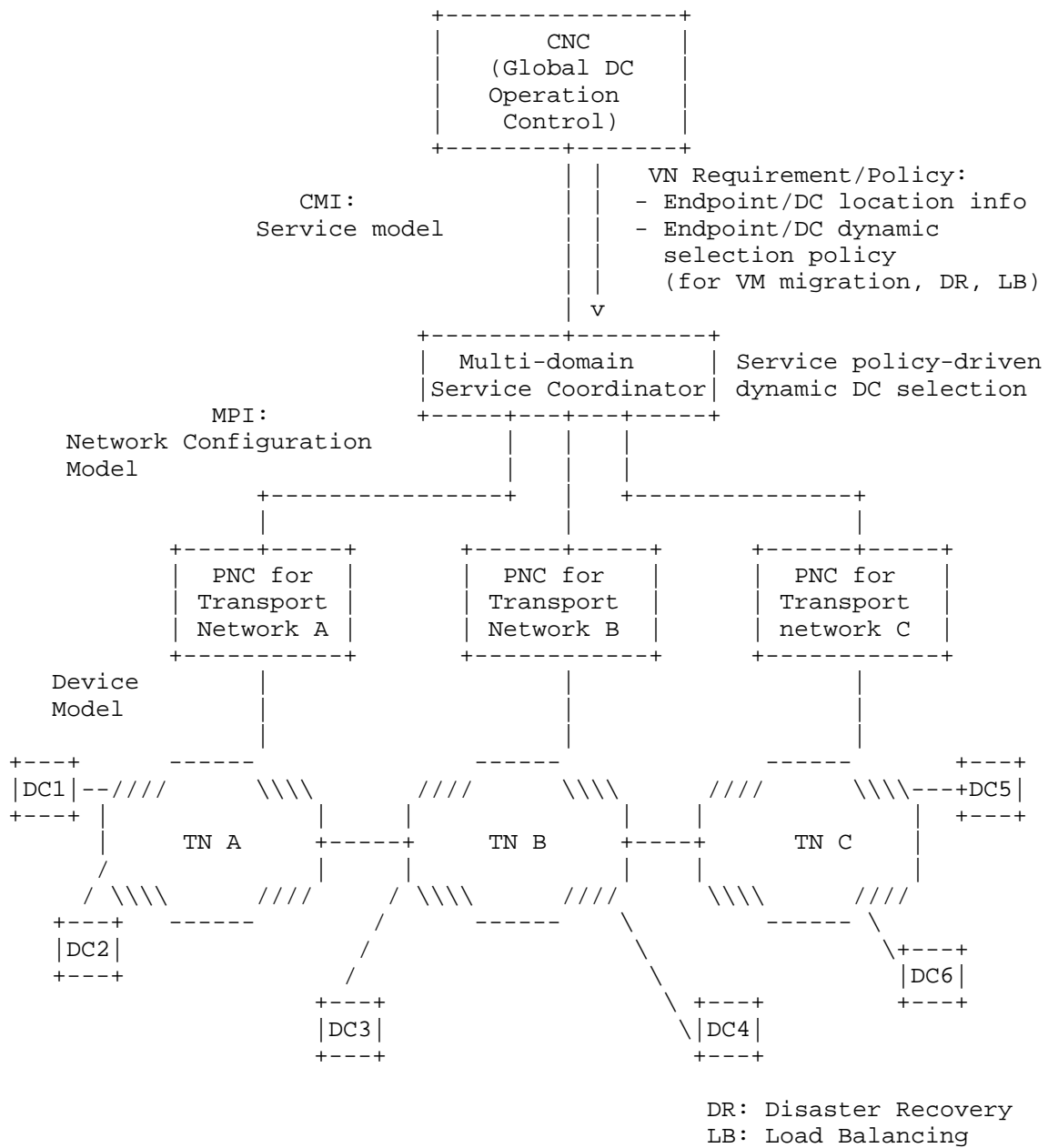


Figure 3: Service Policy-driven Data Center Selection

Figure 3 shows how VN policies from the CNC (Global Data Center Operation) are incorporated by the MDSC to support multi-destination applications. Multi-destination applications refer to applications in which the selection of the destination of a network path for a given source needs to be decided dynamically to support such applications.

Data Center selection problems arise for VM mobility, disaster recovery and load balancing cases. VN's policy plays an important role for virtual network operation. Policy can be static or dynamic. Dynamic policy for data center selection may be placed as a result of utilization of data center resources supporting VMs. The MDSC would then incorporate this information to meet the objective of this application.

#### 5.3.1. CMI (CNC-MDSC Interface)

[ACTN-VN-YANG] is used to express the definition of a VN, its VN creation request, the service objectives (metrics, QoS parameters, etc.), dynamic service policy when VM needs to be moved from one Data Center to another Data Center, etc. This service model is used between the CNC and the MDSC (CMI). The CNC in this use-case is an external entity that wants to create a VN and operates on the VN.

#### 5.3.2. MPI (MDSC-PNC Interface)

The Network Configuration Model is used between the MDSC and the PNCs. Based on the Customer Service Model's request, the MDSC will need to translate the service model into the network configuration model to instantiate a set of multi-domain connections between the prescribed sources and the destinations. The MDSC will also need to dynamically interact with the CNC for dynamic policy changes initiated by the CNC. Upon the determination of the multi-domain connections, the MDSC will need to use the network configuration model such as [TE-Tunnel] to interact with each PNC involved on the path. [TE-Topology] is used to for the purpose of underlying domain network abstraction from the PNC to the MDSC.

#### 5.3.3. PDI (PNC-Device interface)

The Device Model can be used between the PNC and its underlying devices that are controlled by the PNC. The PNC will need to trigger signaling using any mechanisms it employs (e.g. [RSVP-TE-YANG]) to provision its domain path segment. There can be a plethora of choices how to control/manage its domain network. The PNC is responsible to abstract its domain network resources and update it



to the MDSC. Note that this interface is not in the scope of ACTN. This section is provided just for an illustration purpose.

## 6. Security

This document is an informational draft. When the models mentioned in this draft are implemented, detailed security consideration will be given in such work.

How security fits into the whole architecture has the following components:

- the use of Restconf security between components
- the use of authentication and policy to govern which services can be requested by different parties.
- how security may be requested as an element of a service and mapped down to protocol security mechanisms as well as separation (slicing) of physical resources)

## 7. Acknowledgements

We thank Adrian Farrel for providing useful comments and suggestions for this draft.

## 8. References

### 8.1. Informative References

- [Service-YANG] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained, work in progress.
- [Netmod-Yang-Model-Classification] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", draft-ietf-netmod-yang-model-classification, work in progress.
- [Netconf] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.
- [Restconf] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf, work in progress.

- [ACTN-Requirements] Y. Lee, et al., "ACTN Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.
- [ACTN-Frame] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, work in progress.
- [TE-Topology] X. Liu, et. al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.
- [ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.
- [ACTN-Info] Y. Lee & S. Belotti, "Information Model for Abstraction and Control of TE Networks (ACTN)", draft-leebelotti-teas-actn-info, work in progress.
- [Transport-Service-Model] X. Zhang (Editor), "A Service YANG Model for Connection-oriented Transport Networks", draft-zhang-teas-transport-service-model, work in progress.
- [PATH-COMPUTATION-API] I.Busi/S.Belotti et al. "Path Computation API", draft-busibel-ccamp-path-computation-api-00.txt, work in progress
- [RSVP-TE-YANG] T. Saad (Editor), "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp, work in progress.
- [Schedule] X. Liu, et. al., "A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule, work in progress.
- [ACTN-Abstraction] Y. Lee, D. Dhody, and D. Ceccarelli, "ACTN Abstraction Methods", draft-lee-tease-actn-abstraction, work in progress.
- [OTN-YANG] X. Zhang, A. Sharma, and X. Liu, "A YANG Data Model for Optical Transport Network Topology", draft-zhang-ccamp-ll-topo-yang, work in progress.

- [WSON-YANG] Y. Lee, et. al., "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang, work in progress.
- [Flexi-YANG] J.E. Lopez de Vergara, et. al., "YANG data model for Flexi-Grid Optical Networks", draft-vergara-ccamp-flexigrid-yang, work in progress.[ODU-Tunnel] Sharma, R. Rao, and X. Zhang, "OTN Service YANG Model", draft-sharma-ccamp-otn-service-model, work in progress.
- [ACTN-PM-Telemetry] Y. Lee, D. Dhody, S. Karunanithi, R. Vilalta, D. King, and D. Ceccarelli, "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.
- [WSON-Tunnel] Y. Lee, D. Dhody, V. Lopez, D. King, B. Yoon, and R. Vilalta, "A Yang Data Model for WSON Tunnel", draft-lee-ccamp-wson-tunnel-model, work in progress.
- [Flexigrid-Tunnel] J. Vergara, D. Perdices, V. Lopez, O. Gonzalez de Dios, D. King, Y. Lee, and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-vergara-ccamp-flexigrid-media-channel-yang, work in progress.

## 9. Contributors

### Contributor's Addresses

Dhruv Dhody  
Huawei Technologies

Email: dhruv.ietf@gmail.com

Xian Zhang  
Huawei Technologies

Email: zhang.xian@huawei.com

Authors' Addresses

Young Lee  
Huawei Technologies  
5340 Legacy Drive  
Plano, TX 75023, USA  
Phone: (469)277-5838

Email: leeyoung@huawei.com

Haomian Zheng  
Huawei Technologies

Email: zhenghaomian@huawei.com

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Bin Yeong Yoon  
ETRI

Email: byyun@etri.re.kr

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Jong Yoon Shin  
SKT

Email: jongyoon.shin@sk.com

Sergio Belotti  
Nokia

Email: sergio.belotti@nokia.com

