TEAS Working Group
Internet Draft
Intended status: Informational
Expires: November 2016

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

April 14, 2016

Framework for Abstraction and Control of Traffic Engineered Networks

draft-ceccarelli-teas-actn-framework-02

Status of this Memo

Copyright Notice

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.

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 draft provides a framework for Abstraction and Control of Traffic Engineered Networks (ACTN).

Table of Contents

1. Introduction

Traffic Engineered networks have a variety of mechanisms to facilitate 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.

The term Traffic Engineered Network in this draft refers to any connection-oriented network that has the ability of dynamic provisioning, abstracting and orchestrating network resource to the network's clients. Some examples of networks that are in scope of this definition are optical networks, MPLS Transport Profile (MPLS-TP), MPLS Traffic Engineering (MPLS-TE), and other emerging technologies with connection-oriented behavior.

One of the main drivers for Software Defined Networking (SDN) is a decoupling of the network control plane from the data plane. This separation of the control plane from the data plane has been already achieved with the development of MPLS/GMPLS [GMPLS] and PCE [PCE] for TE-based transport networks. One of the advantages of SDN is its logically centralized control regime that allows a global view of the underlying network under its control. Centralized control in SDN helps improve network resources utilization compared with distributed network control. For TE-based transport network control, PCE is essentially equivalent to a logically centralized control for path computation function.

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

. Network and service abstraction: Detach the network and service control from underlying technology and help customer express the network as desired by business needs.

. Coordination of resources across multiple domains and multiple layers to provide end-to-end services regardless of whether the domains use SDN or not.

As networks evolve, the need to provide resource and service abstraction has emerged as a key requirement for operators; this implies in effect the virtualization of network resources so that the network is "sliced" for different tenants shown as a dedicated portion of the network resources

Particular attention needs to be paid to the multi-domain case, where Abstraction and Control of Traffic Engineered Networks (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) as a single virtualized network.

Network virtualization refers to allowing the customers of network operators (see Section 2.1) to utilize a certain amount of network resources as if they own them and thus control their allocated resources with higher layer or application processes that enables the resources to be used in the most optimal way. More flexible, dynamic customer control capabilities are added to the traditional VPN along with a customer specific virtual network view. Customers control a view of virtual network resources, specifically allocated to each one of them. This view is called an abstracted network topology. Such a view may be specific to a specific service, the set of consumed resources or to a particular customer. Customer controller of the virtual network is envisioned to support a plethora of distinct applications. This means that there may be a further level of virtualization that provides a view of resources in the customer’s virtual network for use by an individual application.

The framework described in this draft is named Abstraction and Control of Traffic Engineered Network (ACTN) and facilitates:

- Abstraction of the underlying network resources to higher-layer applications and customers [TE-INFO].

- Virtualization of particular underlying resources, whose selection criterion is the allocation of those resources to a particular customer, application or service. [ONF-ARCH]

- Slicing infrastructure to connect multiple customers to meet specific customer’s service requirements.

- Creation of a virtualized environment allowing operators to view and control multi-domain networks into a single virtualized network;

- Possibility of providing a customer with virtualized network or services (totally hiding the network).

- A virtualization/mapping network function that adapts customer requests to the virtual resources (allocated to them) to the supporting physical network control and performs the necessary mapping, translation, isolation and security/policy enforcement, etc.; This function is often referred to as orchestration.

- The presentation of the networks as a virtualized topology to the customers via open and programmable interfaces. This allows for the recursion of controllers in a customer-provider relationship.

1.1. Terminology

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

- Node: A node is a topological entity describing the "opaque" forwarding aspect of the topological component which represents the opportunity to enable forwarding between points at the edge of the node. It provides the context for instructing the formation, adjustment and removal of the forwarding. A node, in a VN network, can be represented by single physical entity or by a group of nodes moving from physical to virtual network.

- Link: A link is a topological entity describing the effective adjacency between two or more forwarding entities, such as two or more nodes. In its basic form (i.e., point-to-point Link) it associates an edge point of a node with an equivalent edge point on another node. Links in virtual network is in fact connectivity, realized by bandwidth engineering between any two nodes meeting certain criteria, for example, redundancy, protection, latency, not tied to any technology specific characteristics like timeslots or wavelengths. The link can be dynamic, realized by a service in underlay, or static.

- PNC domain: A PNC domain includes all the resources under the control of a single PNC. It can be composed by different routing domains, administrative domains and different layers. The interconnection between PNC domains can be a link or a node.

```
                    border
           -------   link    -------
          (       )---------(       )
         -         -       -         -
        (   PNC     )+---+(   PNC     )
       (  Domain X   )   (  Domain Y   )
        (           )+---+(           )
         -         - border-         -
          (       )   node  (       )
           -------           -------
```

Figure 1 : PNC domain borders

- Virtual Network: A Virtual Network (VN) is a customer view of the transport network. It is composed by a set of physical resources sliced in the provider network and presented to the customer as a set of abstract resources i.e. virtual nodes and virtual links. Depending on the agreement between customer and provider a VN can be just represented by:

  o How the end points can be connected with given SLA attributes(e.g., re satisfying the customer's objectives)
  o A pre-configured set of physical resources
  o Or as outcome of a dynamic request from customer.

  In the first case the VN can be seen at customer level as an e2e connectivity that can be formed by recursive aggregation of lower layers tunnels within the provider domain.
  When the VN is pre-configured, it is provided after a static negotiation between customer and provider while in the third case VN can be dynamically created, deleted, or modified in response to requests from the customer. This implies dynamic changes of network resources reserved for the customer.
  In the second and third case , once that customer has obtained his VN, can act upon the virtual network resources to perform connection management (set-up/release/modify connections).

- Abstract Topology: Every lower controller in the provider network, when is representing its network topology to an higher layer, it may want to hide details of the actual network topology. In such case, an abstract topology may be used for this purpose. Abstract topology enhances scalability for the MDSC to operate multi-domain networks

- Access link: A link between a customer node and a provider node.

- Inter domain link: A link between domains managed by different PNCs. The MDSC is in charge of managing inter-domain links.

- Border node: A node whose interfaces belong to different domains. It may be managed by different PNCs or by the MDSC.

- Access Point (AP): An access point is defined on an access link. It is used to keep confidentiality between the customer and the provider. It is an identifier shared between the customer and the provider, used to map the end points of the border node in the provider NW. The AP can be used by the customer when requesting connectivity service to the provider. A number of parameters, e.g. available bandwidth, need to be associated to the AP to qualify it.

- VN Access Point (VNAP): A VNAP is defined within an AP as part of a given VN and is used to identify the portion of the AP, and hence of the access link) dedicated to a given VN.

2. Business Model of ACTN

The Virtual Private Network (VPN) [RFC4026] and Overlay Network (ON) models [RFC4208] are built on the premise that one single network provider provides all virtual private or overlay networks to its customers. These models are simple to operate but have some disadvantages in accommodating the increasing need for flexible and dynamic network virtualization capabilities.

The ACTN model is built upon entities that reflect the current landscape of network virtualization environments. There are three key entities in the ACTN model [ACTN-PS]:

- Customers
- Service Providers
- Network Providers

2.1. Customers

Within the ACTN framework, different types of customers may be taken into account depending on the type of their resource needs, on their number and type of access. As example, it is possible to group them into two main categories:

Basic Customer: Basic customers include fixed residential users, mobile users and small enterprises. Usually the number of basic customers is high; they require small amounts of resources and are characterized by steady requests (relatively time invariant). A typical request for a basic customer is for a bundle of voice services and internet access. Moreover basic customers do not modify their services themselves; if a service change is needed, it is performed by the provider as proxy and they generally have very few dedicated resources (subscriber drop), with everything else shared on the basis of some SLA, which is usually best-efforts.

Advanced Customer: Advanced customers typically include enterprises, governments and utilities. Such customers can ask for both point to point and multipoint connectivity with high resource demand significantly varying in time and from customer to customer. This is one of the reasons why a bundled service offering is not enough and it is desirable to provide each of them with a customized virtual network service.

Advanced customers may own dedicated virtual resources, or share resources. They may also have the ability to modify their service parameters within the scope of their virtualized environments.

As customers are geographically spread over multiple network provider domains, they have to interface multiple providers and may have to support multiple virtual network services with different underlying objectives set by the network providers. 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 an abstracted view of the topology that spans all of the network providers.

ACTN’s primary focus is Advanced Customers.

Customers of a given service provider can in turn offer a service to other customers in a recursive way. An example of recursiveness with 2 service providers is shown below.

- Customer (of service B)
- Customer (of service A) & Service Provider (of service B)
- Service Provider (of service A)
- Network Provider

```
   +------------------------------------------------------------+    ---
   |                                                            |     ^
   |                                     Customer (of service B)|     .
   | +--------------------------------------------------------+ |     B
   | |                                                        | |--- .
   | |Customer (of service A) & Service Provider(of service B)| | ^   .
   | | +---------------------------------------------------+  | | .   .
   | | |                                                   |  | | .   .
   | | |                     Service Provider (of service A)|  | | A   .
   | | |+------------------------------------------+       |  | | .   .
   | | ||                                          |       |  | | .   .
   | | ||                           Network provider|       |  | | v   v
   | | |+------------------------------------------+       |  | |------
   | | +---------------------------------------------------+  | |
   | +--------------------------------------------------------+ |
   +------------------------------------------------------------+
```

Figure 2 : Service Recursiveness.

2.2. Service Providers

Service providers are the providers of virtual network services to their customers. Service providers may or may not own physical network resources. When a service provider is the same as the network provider, this is similar to traditional VPN models. This model works well when the customer maintains a single interface with a single provider. When customer location spans across multiple independent network provider domains, then it becomes hard to facilitate the creation of end-to-end virtual network services with this model.

A more interesting case arises when network providers only provide infrastructure while service providers directly interface their customers. In this case, service providers themselves are customers of the network infrastructure providers. One service provider may need to keep multiple independent network providers as its end-users span geographically across multiple network provider domains as shown in Figure 2 where Service Provider A uses resources from Network Provider A and Network Provider B to offer a virtualized network to its customer.

```
Customer           X -----------------------------------X

Service Provider A X -----------------------------------X

Network Provider B                   X------------------X

Network Provider A X------------------X
```

Figure 3 : A service Provider as Customer of Two Network Providers.

The ACTN network model is predicated upon this three tier model and is summarized in Figure 3:

```
                   +----------------------+
                   |       customer       |
                   +----------------------+
                             |
                             |   /\  Service/Customer specific
                             |   ||  Abstract Topology
                             |   ||
                   +----------------------+  E2E abstract
                   |  Service Provider    | topology creation
                   +----------------------+
                   /         |            \
                  /          |             \  Network Topology
                 /           |              \ (raw or abstract)
                /            |               \
+------------------+   +------------------+   +------------------+
|Network Provider 1|   |Network Provider 2|   |Network Provider 3|
+------------------+   +------------------+   +------------------+
```

Figure 4 : Three tier model.

There can be multiple types of service providers.

- Data Center providers: can be viewed as a service provider type as they own and operate data center resources to various WAN customers, they can lease physical network resources from network providers.
- Internet Service Providers (ISP): can be a service provider of internet services to their customers while leasing physical network resources from network providers.
- Mobile Virtual Network Operators (MVNO): provide mobile services to their end-users without owning the physical network infrastructure.

2.3. Network Providers

Network Providers are the infrastructure providers that own the physical network resources and provide network resources to their customers. The layered model proposed by this draft separates the concerns of network providers and customers, with service providers acting as aggregators of customer requests.

3. ACTN architecture

This section provides a high-level control and interface model of ACTN.

The ACTN architecture, while being aligned with the ONF SDN architecture [ONF-ARCH], is presenting a 3-tiers reference model. It allows for hierarchy and recursiveness not only of SDN controllers but also of traditionally controlled domains. It defines three types of controllers depending on the functionalities they implement. The main functionalities that are identified are:

- Multi domain coordination function: With the definition of domain being "everything that is under the control of the same controller",it is needed to have a control entity that oversees the specific aspects of the different domains and to build a single abstracted end-to-end network topology in order to coordinate end-to-end path computation and path/service provisioning.

- Virtualization/Abstraction function: To provide an abstracted view of the underlying network resources towards customer, being it the client or a higher level controller entity. It includes computation of customer resource requests into virtual network paths based on the global network-wide abstracted topology and the creation of an abstracted view of network slices allocated to each customer, according to customer-specific virtual network objective functions, and to the customer traffic profile.

- Customer mapping function: In charge of mapping customer VN setup commands into network provisioning requests to the Physical Network Controller (PNC) according to business OSS/NMS provisioned static or dynamic policy. Moreover it provides mapping and translation of customer virtual network slices into physical network resources

- Virtual service coordination: Virtual service coordination function in ACTN incorporates customer service-related knowledge into the virtual network operations in order to seamlessly operate virtual networks while meeting customer's service requirements.

The virtual services that are coordinated under ACTN can be split into two categories:

- Service-aware Connectivity Services: This category includes all the network service operations used to provide connectivity between customer end-points while meeting policies and service related constraints. The data model for this category would include topology entities such as virtual nodes, virtual links, adaptation and termination points and service-related entities such as policies and service related constraints. (See Section 4.2.2)

- Network Function Virtualization Services: These kinds of services are usually setup in NFV (e.g. cloud) providers and require connectivity between a customer site and the NFV provider site (e.g. a data center). These VNF services may include a security function like firewall, a traffic optimizer, the provisioning of storage or computation capacity. In these cases the customer does not care whether the service is implemented in a given data center or another. This allows the network provider divert customer requests where most suitable. This is also known as "end points mobility" case. (See Section 4.2.3)

The types of controller defined are shown in Figure 4 below and are the following:

- CNC - Customer Network Controller
- MDSC - Multi Domain Service Coordinator

. PNC - Physical Network Controller

```
   VPN customer         NW Mobile Customer     ISP NW service Customer
       |                         |                           |
   +-------+                 +-------+                   +-------+
   | CNC-A |                 | CNC-B |                   | CNC-C |
   +-------+                 +-------+                   +-------+
        \                        |                           /
          -----------            |CMI I/F      --------------
                     \           |            /
                      +------------------------+
                      |         MDSC           |
                      +------------------------+
                      /          |            \
         -------------           |MPI I/F      -------------
        /                        |                          \
   +-------+                 +-------+                   +-------+
   |  PNC  |                 |  PNC  |                   |  PNC  |
   +-------+                 +-------+                   +-------+
        | GMPLS             /      |                      /    \
        | trigger          /       |                     /      \
       --------         ----      +-----+            +-----+     \
      (        )       (    )     | PNC |            | PCE |      \
      -        -      ( Phys )    +-----+            +-----+    -----
     (  GMPLS   )      (Netw)        |                /        (     )
    (  Physical  )      ----         |               /        ( Phys. )
     (  Network )                 -----        -----           ( Net )
      -        -                 (     )      (     )           -----
      (        )                ( Phys. )    ( Phys  )
       --------                  ( Net )      ( Net )
                                  -----        -----
```

Figure 5 : ACTN Control Hierarchy

3.1. Customer Network Controller

A Virtual Network Service is instantiated by the Customer Network Controller via the CMI (CNC-MDSC Interface). As the Customer Network Controller directly interfaces the applications, it understands multiple application requirements and their service needs. It is assumed that the Customer Network Controller and the MDSC have a common knowledge on the end-point interfaces based on their business negotiation prior to service instantiation. End-point interfaces refer to customer-network physical interfaces that connect customer premise equipment to network provider equipment.

In addition to abstract networks, ACTN allows to provide the CNC with services. Example of services include connectivity between one of the customer’s end points with a given set of resources in a data center from the service provider.

3.2. Multi Domain Service Coordinator

The MDSC (Multi Domain Service Coordinator) sits between the CNC (the one issuing connectivity requests) and the PNCs (Physical Network Controllersr - the ones managing the physical network resources). The MDSC can be collocated with the PNC, especially in those cases where the service provider and the network provider are the same entity.

The internal system architecture and building blocks of the MDSC are out of the scope of ACTN. Some examples can be found in the Application Based Network Operations (ABNO) architecture [ABNO] and the ONF SDN architecture [ONF-ARCH].

The MDSC is the only building block of the architecture that is able to implement all the four ACTN main functionalities, i.e. multi domain coordination function, virtualization/abstraction function, customer mapping function and virtual service coordination. The key point of the MDSC and the whole ACTN framework is detaching the network and service control from underlying technology and help customer express the network as desired by business needs. The MDSC envelopes the instantiation of right technology and network control to meet business criteria. In essence it controls and manages the primitives to achieve functionalities as desired by CNC
A hierarchy of MDSCs can be foreseen for scalability and administrative choices.

```
   +-------+                 +-------+                 +-------+
   | CNC-A |                 | CNC-B |                 | CNC-C |
   +-------+                 +-------+                 +-------+
       \                       |                         /
          ----------             |             ----------
                     \           |            /
                      +-----------------------+
                      |         MDSC          |
                      +-----------------------+
                      /          |            \
            ----------           |             -----------
           /                     |                        \
   +----------+              +----------+              +---------+
   |   MDSC   |              |   MDSC   |              |  MDSC   |
   +----------+              +----------+              +---------+
        |                     /     |                    /    \
        |                    /      |                   /      \
     +-----+               +-----+  +-----+            +-----+  +-----+
     | PNC |               | PNC |  | PNC |            | PNC |  | PNC |
     +-----+               +-----+  +-----+            +-----+  +-----+
```

Figure 6 : Controller recursiveness

A key requirement for allowing recursion of MDSCs is that a single interface needs to be defined both for the north and the south bounds.

In order to allow for multi-domain coordination a 1:N relationship must be allowed between MDSCs and between MDSCs and PNCs (i.e. 1 parent MDSC and N child MDSC or 1 MDSC and N PNCs). In addition to that it could be possible to have also a M:1 relationship between MDSC and PNC to allow for network resource partitioning/sharing among different customers not necessarily connected to the same MDSC (e.g. different service providers).

3.3. Physical Network Controller

The Physical Network Controller is the one in charge of configuring the network elements, monitoring the physical topology of the network and passing it, either raw or abstracted, to the MDSC.

The internal architecture of the PNC, his building blocks and the way it controls its domain, are out of the scope of ACTN. Some examples can be found in the Application Based Network Operations (ABNO) architecture [ABNO] and the ONF SDN architecture [ONF-ARCH]

The PNC, in addition to being in charge of controlling the physical network, is able to implement two of the four ACTN main functionalities: multi domain coordination function and virtualization/abstraction function
A hierarchy of PNCs can be foreseen for scalability and administrative choices.

3.4. ACTN interfaces

To allow virtualization and multi domain coordination, the network has to provide open, programmable interfaces, in which customer applications can create, replace and modify virtual network resources and services in an interactive, flexible and dynamic fashion while having no impact on other customers. Direct customer control of transport network elements and virtualized services is not perceived as a viable proposition for transport network providers due to security and policy concerns among other reasons. In addition, as discussed in the previous section, the network control plane for transport networks has been separated from data plane and as such it is not viable for the customer to directly interface with transport network elements.

Figure 5 depicts a high-level control and interface architecture for ACTN. A number of key ACTN interfaces exist for deployment and operation of ACTN-based networks. These are highlighted in Figure 5 (ACTN Interfaces) below:

```
                .---------------
               -------------    |
              | Application |--
               -------------
                     ^
                     | I/F A                 --------
                     v                      (        )
                --------------             -          -
               | Customer     |           (  Customer  )
               |  Network     |--------->(    Network   )
               |   Controller |           (            )
                --------------             -          -
                     ^                      (        )
                     | I/F B                 --------
                     v
                --------------
               | MultiDomain  |
               |  Service     |
               |   Coordinator|            --------
                --------------            (        )
                     ^                   -          -
                     | I/F C            (  Physical  )
                     v                 (    Network   )
                  ---------------       (            )      --------
                 |               |<----> -          -      (        )
                --------------   |        (        )      -          -
               | Physical     |--          --------      (  Physical  )
               |  Network     |<---------------------->(    Network   )
               |   Controller |         I/F D            (            )
                --------------                            -          -
                                                           (        )
                                                            --------
```

Figure 7 : ACTN Interfaces

   The interfaces and functions are described below:

      . Interface A: A north-bound interface (NBI) that will
        communicate the service request or application demand. A
        request will include specific service properties, including:
        services, topology, bandwidth and constraint information.

      . Interface B: The CNC-MDSC Interface (CMI) is an interface
        between a Customer Network Controller and a Multi Service
        Domain Controller. It requests the creation of the network
        resources, topology or services for the applications. The
        Virtual Network Controller may also report potential network

topology availability if queried for current capability from the Customer Network Controller.

. Interface C: The MDSC-PNC Interface (MPI) is an interface between a Multi Domain Service Coordinator and a Physical Network Controller. It communicates the creation request, if required, of new connectivity of bandwidth changes in the physical network, via the PNC. In multi-domain environments, the MDSC needs to establish multiple MPIs, one for each PNC, as there are multiple PNCs responsible for its domain control.

. Interface D: The provisioning interface for creating forwarding state in the physical network, requested via the Physical Network Controller.

The interfaces within the ACTN scope are B and C.

4. VN creation process

The provider can present to the customer different level of network abstraction, spanning from one extreme (say "black") where nothing is shown, just the APs, to the other extreme (say "white") where a slice of the network is shown to the customer. There are shades of gray in between where a number of abstract links and nodes can be shown.

The VN creation is composed by two phases: Negotiation and Implementation.

Negotiation: In the case of grey/white topology abstraction, there is an a priori phase in which the customer agrees with the provider on the type of topology to be shown, e.q. 10 virtual links and 5 virtual nodes with a given interconnectivity. This is something that is assumed to be preconfigured by the operator off-line, what is online is the capability of modifying/deleting something (e.g. a virtual link). In the case of "black" abstraction this negotiation phase does not happen, in the sense that the customer can only see the APs of the network.

Implementation: In the case of black topology abstraction, the customers can ask for connectivity with given constraints/SLA between the APs and LSPs/tunnels are created by the provider to satisfy the request. What the customer sees is only that his CEs are connected with a given SLA. In the case of grey/white topology the customer creates his own LSPs accordingly to the topology that was presented to him.

5. Access Points and Virtual Network Access Points

In order not to share unwanted topological information between the customer domain and provider domain, a new entity is defined and associated to an access link, the Access Point (AP). See the definition of AP in Section 1.1.

A customer node will use APs as the end points for the request of VNs.

A number of parameters need to be associated to the APs. Such parameters need to include at least: the maximum reservable bandwidth on the link, the available bandwidth and the link characteristics (e.g. switching capability, type of mapping).

Editor note: it is not appropriate to define link characteristics like bandwidth against a point (AP). A solution needs to be found.

```
                      -------------
                     (             )
                    -               -
     +---+ X       (                 )      Z +---+
     |CE1|---+----(                   )---+---|CE2|
     +---+   |     (                 )    |   +---+
            AP1     -               -    AP2
                     (             )
                      -------------
```

Figure 8 : APs definition customer view

Let’s take as example a scenario in which CE1 is connected to the network via a 10Gb link and CE2 via a 40Gb link. Before the creation of any VN between AP1 and AP2 the customer view can be summarized as follows:

AP id	MaxResBw	AvailableBw	CE,port
AP1	10Gb	10Gb	CE1,portX
AP2	40Gb	40Gb	CE2,portZ

Table 1: AP - customer view

On the other side what the provider sees is:

```
          -------            -------
         (       )          (       )
        -         -        -         -
   W  (+---+       )      (       +---+)  Y
-+---( |PE1| Dom.X  )----(  Dom.Y |PE2| )---+-
 |    (+---+       )      (       +---+)    |
 AP1    -         -        -         -     AP2
         (       )          (       )
          -------            -------
```

Figure 9 : Provider view of the AP

Which in the example above ends up in a summarization as follows:

AP id	MaxResBw	AvailableBw	PE,port
AP1	10Gb	10Gb	PE1,portW
AP2	40Gb	40Gb	PE2,portY

Table 2: AP - provider view

The second entity that needs to be defined is a structure within the AP that is linked to a VN and that is used to allow for different VN to be provided starting from the same AP. It also allows reserving the bandwidth for the VN on the access link. Such entity is called Virtual Network Access Point. For each virtual network is defined on an AP, a different VNAP is created.

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

The customer view would evolve as follows:

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

Table 3: AP and VNAP - provider view after VN creation

5.1. Dual homing scenario

Often there is a dual homing relationship between a CE and a pair of PE. This case needs to be supported also by the definition of VN, AP and VNAP. Suppose to have CE1 connected to two different PE in the operator domain via AP1 and AP2 and the customer needing 5Gbps of bandwidth between CE1 and CE2.

```
         AP1    --------------    AP3
        -------(PE1)      (PE3) -------
     W /      -                -        \X
+---+ /      (                  )        \ +---+
|CE1|       (                    )         |CE2|
+---+ \      (                  )        / +---+
     Y \      -                -        /Z
        -------(PE2)      (PE4) -------
         AP2    --------------   AP4
```

Figure 10 : 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 be flowing 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 as follows:

AP/VNAPid	MaxResBw	AvailableBw	CE,port	Dual Homing
AP1 -VNAP1.9	10Gbps 5Gbps	5Gbps N.A.	CE1,portW	 VNAP2.9
AP2 -VNAP2.9	40Gbps 5Gbps	35Gbps N.A.	CE1,portY	 VNAP1.9
AP3 -VNAP3.9	40Gbps 5Gbps	35Gbps N.A.	CE2,portZ	 NONE

Table 4: Dual homing - customer view after VN creation

6. End point selection & mobility

Virtual networks could be used as the infrastructure to connect a number of sites of a customer among them or to provide connectivity between customer sites and virtualized network functions (VNF) like for example virtualized firewall, vBNG, storage, computational functions.

6.1. End point selection & mobility

A VNF could be deployed in different places (e.g. data centers A, B or C in figure below) but the VNF provider (=ACTN customer) doesn’t know which is the best site where to install the VNF from a network point of view (e.g. latency). For example it is possible to compute the path minimizing the delay between AP1 and AP2, but the customer doesn’t know a priori if the path with minimum delay is towards A, B or C.

```
                    -------            -------
                   (       )          (       )
                  -         -        -         -
   +---+         (           )      (           )        +-----+
   |CE1|---+----(  Domain X   )----(  Domain Y   )---+---|DC-A|
   +---+   |     (           )      (           )    |   +-----+
           AP1    -         -        -         -    AP2
                   (       )          (       )
                    ---+---            ---+---
                   AP3 |              AP4 |
                    +-----+             +-----+
                    |DC-B|              |DC-C|
                    +-----+             +-----+
```

Figure 11 : End point selection

In this case the VNF provider (=ACTN customer) should be allowed to ask for a VN between AP1 and a set of end points. The list of end points is provided by the VNF provider. When the end point is identified the connectivity can be instantiated and a notification can be sent to the VNF provider for the instantiation of the VNF.

6.2. Preplanned end point migration

A premium SLA for VNF service provisioning consists on the offering of a protected VNF instantiated on two or more sites and with a hot stand-by protection mechanism. In this case the VN should be provided so to switch from one end point to another upon a trigger from the VNF provider or an automatic failure detection mechanism. An example is provided in figure below where the request from the VNF provider is for connectivity with given constraint and resiliency between CE1 and a VNF with primary installation in DC-A and a protection in DC-C.

```
                    -------            -------
                   (       )          (       )
                  -         -    __  -         -
   +---+         (           )      (           )        +-----+
   |CE1|---+----(  Domain X   )----(  Domain Y   )---+---|DC-A|
   +---+   |     (           )      (           )    |   +-----+
           AP1    -         -        -         -    AP2     |
                   (       )          (       )             |
                    ---+---            ---+---              |
                   AP3 |              AP4 |         HOT STANDBY
                                       +-----+              |
                                       |DC-C|<-------------
                                       +-----+
```

Figure 12 : Preplanned endpoint migration

6.3. On the fly end point migration

The one the fly end point migration concept is very similar to the end point selection one. The idea is to give the provider not only the list of sites where the VNF can be installed, but also a mechanism to notify changes in the network that have impacts on the SLA. After an handshake with the customer controller/applications, the bandwidth in network would be moved accordingly with the moving of the VNFs.

7. Security

TBD

8. References

8.1. Informative References

[PCE]   Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", IETF RFC 4655, August 2006.

[RFC4026] L. Andersson, T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.

[RFC4208] G. Swallow, J. Drake, H.Ishimatsu, Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, October 2005.

[PCE-S] Crabbe, E, et. al., "PCEP extension for stateful PCE",draft-ietf-pce-stateful-pce, work in progress.

[GMPLS] Manning, E., et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[NFV-AF] "Network Functions Virtualization (NFV); Architectural Framework", ETSI GS NFV 002 v1.1.1, October 2013.

[ACTN-PS] Y. Lee, D. King, M. Boucadair, R. Jing, L. Contreras Murillo, "Problem Statement for Abstraction and Control of Transport Networks", draft-leeking-actn-problem-statement, work in progress.

[ONF] Open Networking Foundation, "OpenFlow Switch Specification Version 1.4.0 (Wire Protocol 0x05)", October 2013.

[TE-INFO] A. Farrel, Editor, "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", draft-ietf-teas-interconnected-te-info-exchange, work in progress.

[ABNO] King, D., and Farrel, A., "A PCE-based Architecture for Application-based Network Operations", draft-farrkingel-pce-abno-architecture, work in progress.

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

[Cheng] W. Cheng, et. al., "ACTN Use-cases for Packet Transport Networks in Mobile Backhaul Networks", draft-cheng-actn-ptn-requirements, work in progress.

[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, work in progress.

[Fang] L. Fang, "ACTN Use Case for Multi-domain Data Center Interconnect", draft-fang-actn-multidomain-dci, work in progress.

[Klee] K. Lee, H. Lee, R. Vilata, V. Lopez, "ACTN Use-case for On-demand E2E Connectivity Services in Multiple Vendor Domain Transport Networks", draft-klee-actn-connectivity-multi-vendor-domains, work in progress.

[Kumaki] K. Kumaki, T. Miyasaka, "ACTN : Use case for Multi Tenant VNO ", draft-kumaki-actn-multitenant-vno, work in progress.

[Lopez] D. Lopez (Ed), "ACTN Use-case for Virtual Network Operation for Multiple Domains in a Single Operator Network", draft-lopez-actn-vno-multidomains, work in progress.

[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, work in progress.

[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, work in progress.

9. Contributors

Authors' Addresses

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

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

Luyuan Fang
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@alcatel-lucent.com

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

Dhruv Dhoddy
Huawei Technologies
dhruv.ietf@gmail.com

Gert Grammel
Juniper Networks
ggrammel@juniper.net

Internet Engineering Task Force H. Chen
Internet-Draft Huawei Technologies
Intended status: Standards Track M. Toy
Expires: September 22, 2016 Comcast
L. Liu
Fujitsu
K. Pithewan
Infinera
March 21, 2016

Framework for Temporal Tunnel Services
draft-chen-teas-frmwk-tts-01.txt

Abstract

For existing MPLS LSP tunnel services, it is hard for LSP tunnels to be booked in advance. In addition, once an LSP tunnel is set up, it is assumed to consume a certain amount of resources such as link bandwidth forever.

Temporal LSP tunnel services (TTS) provides an easy way for us to book temporal LSP tunnels in advance. More importantly, a temporal LSP is an LSP with one or more time intervals and it is assumed to consume the resources and carry traffic only in these time intervals.

This document specifies a framework for temporal LSP tunnel services and provides a few of reference models along with logical components required to design a solution for TTS.

Table of Contents

1. Introduction

Once an existing multiprotocol label switching (MPLS) traffic engineering (TE) label switched path (LSP) is set up, it is assumed to carry traffic forever until it is down. When an MPLS TE LSP tunnel is up, it is assumed to consume the reserved network resources for it forever even though the LSP may only use the network resources during some period of time. As a result, the entire network resources are not used efficiently. Moreover, a tunnel service can not be reserved or booked in advance for a period of time or a sequence of time periods/intervals.

Temporal LSP tunnel services (TTS) provides an easy way for us to book temporal LSP tunnels in advance. More importantly, a temporal LSP is an LSP with one or more time intervals/periods and it is assumed to consume the resources and carry traffic only in these time intervals. Thus the entire network resources are efficiently used. Moreover, some new services can be provided easily. For example, a user can book a tunnel service in advance for a given time interval or a sequence of given time intervals. Tunnel services may be easily scheduled.

This document specifies a framework for temporal LSP tunnel services and provides a few of reference models along with logical components required to design a solution for TTS.

2. Terminology

A Time Interval: a time period from time Ta to time Tb.

LSP: Label Switched Path. An LSP is a P2P (point-to-point) LSP or a P2MP (point-to-multipoint) LSP.

LSP with a time interval: LSP that carries traffic in the time interval.

LSP with a sequence of time intervals: LSP that carries traffic in each of the time intervals.

Temporal LSP: LSP with a time interval or LSP with a sequence of time intervals.

TED: Traffic Engineering Database.

CSPF: Constrained Shortest Path First.

LER: Label Edge Router.

PCE: Path Computation Element.

PCEP: Path Computation Element Communication Protocol.

3. Operations Overview

This section briefly describes some operations on a temporal LSP.

3.1. Simple Time Interval

For a temporal LSP, a user configures it with a time interval or a sequence of time intervals. A simple time interval is a time period from time Ta to time Tb, which may be represented as [Ta, Tb].

When an LSP is configured with time interval [Ta, Tb], a path satisfying the constraints for the LSP in the time interval is computed and the LSP along the path is set up to carry traffic from time Ta to time Tb.

In addition to simple time intervals, there are recurrent time intervals and elastic time intervals. Sometimes a simple time interval is called a time interval.

3.2. Recurrent Time Interval

A recurrent time interval represents a series of repeated simple time intervals. It has a simple time interval such as [Ta, Tb], a number of repeats such as 10 (repeats 10 times), and a repeat cycle/time such as a week (repeats every week). The recurrent time interval: "[Ta, Tb] repeats n times with repeat cycle C" represents n+1 simple time intervals as follows:

  [Ta, Tb], [Ta+C, Tb+C], [Ta+2C, Tb+2C], ..., [Ta+nC, Tb+nC]

When an LSP is configured with a recurrent time interval such as "[Ta, Tb] repeats 10 times with a repeat cycle a week" (representing 11 simple time intervals), a path satisfying the constraints for the LSP in each of the simple time intervals represented by the recurrent time interval is computed and the LSP along the path is set up to carry traffic in each of the simple time intervals.

3.3. Changes to Time Interval

After a temporal LSP is configured, a user may change its parameters including some of the time intervals configured. A new time interval may be added, an existing time interval may be removed or changed.

When a new time interval is added to an existing LSP, a path satisfying the constraints for the LSP in the time interval is computed and the LSP along the path is set up to carry traffic in the time interval.

When an existing time interval is removed from an existing LSP, the time interval is deleted from the lifetime of the LSP. If the lifetime is over, the LSP is deleted.

A change to an existing time interval may generate some of four possible results:

1. The existing time interval is extended for a time period EA after the existing time period;

2. The existing time interval is extended for a time period EB before the existing time period;

3. The existing time interval is shrunk for a time period SA from the end of the existing time period; and

4. The existing time interval is shrunk for a time period SB from the beginning of the existing time period.

When an existing time interval for an LSP is extended, a path satisfying the constraints for the LSP in the extended time interval is computed and the LSP along the path is set up to carry traffic in the extended time interval. If the LSP is already up to carry traffic in the existing time interval, the lifetime of the LSP is extended for time period EA following the existing time interval.

When an existing time interval for an LSP is shrunk, the shrunk time periods are removed from the lifetime of the LSP.

4. Problem Statement

Assume that a set of temporal LSPs have been set up in a network. In other words, the network resources have been reserved in advance for the LSPs in the set. For every LSP configured with a number of time intervals, the network resources have been reserved in advance for each of these time intervals. Initially, there is no LSP set up in the network.

For the given state of the network, how to handle/satisfy a set of service requests containing set up a number of temporal LSPs, delete a number of temporal LSPs and update a number of temporal LSPs?

More specifically, based on the current network state, how to compute paths for the temporal LSPs to be set up? how to reserve the network resources in advance along the paths computed for the time intervals configured for the LSPs? how to release the network resources reserved in advance for the LSPs to be deleted and update the network state accordingly? how to change the parameters for the LSPs configured with time intervals and update the network state accordingly?

The reference models described in the following section can provide solutions for this.

5. Reference Models

This section presents a few of reference models for providing temporal tunnel services (TTS) after introducing some of building blocks. For each of the models, its advantages and disadvantages are listed.

5.1. Building Blocks

This section briefly describes some of the components that may be used to build a solution for creating and maintaining temporal LSP tunnels.

5.1.1. Temporal TED

The Traffic Engineering (TE) information in a normal TE Database (TED) represents a unreserved bandwidth Bi at each of eight priority levels for a link at one point of time, i.e., at the current time.

```
      Bandwidth
        ^
        |
      Bi|______________________________________________________
        |
        |
       -+-------------------------------------------------------> Time
        |
```

This means that the link has bandwidth Bi at a priority level from now to forever until there is a change to it. Thus, a TE Label Switching Path (LSP) tunnel for a given time interval cannot be set up in advance using the information in the TED and the bandwidth cannot be reserved in advance for the LSP in the time interval given.

The TED needs to be enhanced for supporting temporal LSPs. The enhanced TED is called Temporal TED (T-TED for short). It maintains the TE information such as bandwidth for every link with a series of time intervals.

For example, suppose that the amount of the unreserved bandwidth at a priority level for a link is Bj in a time interval from time Tj to Tk (k = j+1), where j = 0, 1, 2, ... The unreserved bandwidth for the link can be represented as

[T0, B0], [T1, B1], [T2, B2], [T3, B3], ...

```
     Bandwidth
       ^
       |                                    B3
       |          B1                        ___________
       |          __________
       |B0                                             B4
       |__________          B2                         _________
       |                    ________________
       |
      -+-------------------------------------------------------> Time
       |T0        T1        T2              T3         T4
```

The unreserved bandwidth information above for the link will be stored in the T-TED.

If an LSP is completely deleted at time T and uses bandwidth B, then for every time interval/period (after time T) during which bandwidth B is reserved for the LSP on a link, B is added to the link for that time interval/period.

If an LSP is to be up at time T and uses bandwidth B, then for every time interval/period (after time T) during which bandwidth B is reserved for the LSP on a link, B is subtracted from the link for that time interval/period.

5.1.2. Temporal CSPF

An existing constrained shortest path first (CSPF) (or say a normal CSPF) computes a path for a normal LSP that satisfies a set of given constraints using a traffic engineering database (TED).

A temporal CSPF (T-CSPF for short) computes a path for a temporal LSP (i.e., an LSP with a number of time intervals) that satisfies a set of given constraints in each of the time intervals through using a temporal TED (T-TED).

5.1.3. Temporal Label Database

In a centralized controller, a normal label database (LDB) records and maintains the status of every label for every node (and/or interface) in a network, which the controller controls. The status of a label indicates whether the label is assigned to an LSP.

A temporal label database (T-LDB) in a centralized controller records and maintains the status of every label in a series of time intervals for every node (and/or interface) in a network, on which the controller controls. The status of a label in a time interval indicates whether the label is assigned to an LSP in the time interval.

If there are enough labels anytime, we do not need any temporal label database and we can just use a normal label database. For example, if we can make sure that at any time the number of LSPs going through any node in the network is less than the number of labels on the node, then there are enough labels anytime. Thus, we can just use a normal label database.

5.1.4. Temporal LSP Tunnel Manager

An existing LSP tunnel manager (or say a normal LSP tunnel manager) receives a request for an operation on an MPLS TE LSP from a user or an application. The operation may be a creation of a new MPLS TE LSP tunnel, a deletion of an existing MPLS TE LSP tunnel, or a change to an existing LSP tunnel.

A temporal LSP tunnel manager (T-LSP Manager for short) receives a request for an operation on a temporal LSP from a user or an application. The operation may be a creation of a new temporal LSP tunnel, a deletion of an existing temporal LSP tunnel, or a change to an existing temporal LSP tunnel.

When receiving a request for creating a new temporal LSP (i.e., an LSP with a sequence of time intervals), the T-LSP manager asks the T-CSPF to compute a path for the LSP that satisfies the constraints given for the LSP in each of the time intervals.

After obtaining the path for the LSP from the T-CSPF, the T-LSP manager requests the T-LDB to assign labels along the path for the LSP and asks the T-TED to reserve the resources such as link bandwidth along the path for the LSP in each of the time intervals if it is used in a centralized controller.

The T-LSP manager in a centralized controller really sets up the LSP along the path in the network by writing a forwarding entry on every node along the path through the API to the network in each of the time intervals. The T-LSP manager in a distributed environment initiates the RSVP signaling to set up the LSP along the path.

The T-LSP manager records the related information for the LSP into the temporal LSP database (T-LSPDB). The information includes the time intervals for the LSP, the path computed for the LSP, the resources such as bandwidth reserved along the path in each of the time intervals for the LSP (for centralized controller), the labels assigned along the path for the LSP (for centralized controller), and the status of the LSP.

5.1.5. Temporal LSP Database

A temporal LSP database (T-LSPDB for short) in a centralized controller stores the related information for every temporal LSP. For each LSP, the following information will be stored in the T-LSPDB:

- o the time intervals for the LSP,
- o the paths computed for the LSP,
- o the resources such as bandwidth reserved along the path in each of the time intervals for the LSP,
- o the labels assigned along the path for the LSP, and
- o the status of the LSP.

In a distributed environment, a T-LSPDB on a label edge router (LER) stores the following information for every temporal LSP originating from the LER (i.e., the LER is the ingress of the LSP):

- o the time intervals for the LSP,
- o the paths computed for the LSP, and
- o the status of the LSP.

5.1.6. Temporal PCE

A temporal PCE (T-PCE for short) is an enhanced version of the existing PCE. It receives a request for computing a path for a temporal LSP crossing multiple domains, computes the path for the LSP and replies to the request with the path computed. For the LSP with a number of time intervals and some constraints, the path computed satisfies the constraints in each of the time intervals.

5.2. Centralized Model

This section presents two centralized reference models. one model is for a single domain, the other for multiple domains.

5.2.1. Centralized Model for Single Domain

Figure below illustrates a centralized SDN controller reference model for temporal LSP tunnel services for a network (i.e., a single domain).

```
+--------------------------------------------+
| TS-SDN Controller                          |
|                     +---------------+      |
|        /------------| T-LSP Manager |      |
|       /     Ia      +---------------+      |
| +--------+           | |  |       \        |
| | T-CSPF |   ________| |  |        \Id     |
| +--------+  /   Ib    /Ic |    +---------+ |
|       \Ie  /         /    |    | T-LSPDB | |
|    +---------+ +-------+  |    +---------+ |
|    |  T-TED  | | T-LDB |  |                |
|    +---------+ +-------+  |                |
|          \         \      |In              |
+---------------API to Network(PCEP+)--------+
                      /       \
                     /         \____
                    /           \   \____
                   /\  .---. .---+       \
                  |  \(     '    |'.---. |
                  |---\  Network |      '+.
                 (o    \         |       | )
                  (     |        |       o)
                   (    |        |       )
                    (   o        o    .-'
                     '               )
                      '---._.-.     )
                               '---'
```

The temporal SDN (TS-SDN) controller in the reference model controls a network through an API to the network such as PCEP+ (extensions to PCEP for central controller). The TS-SDN controller is responsible for creating and maintaining every temporal LSP in the network.

The TS-SDN controller comprises a number of modules, including a T-LSP manager, a T-CSPF, a T-TED, a T-LDB and a T-LSPDB. The interfaces among these modules are listed as follows:

o Interface Ia between the T-LSP manager and the T-CSPF. Through this interface, the T-LSP manager requests the T-CSPF to compute a path for a temporal LSP with a number of time intervals and a set of constraints, and the T-CSPF responses the T-LSP manager with the path computed that satisfies the constraints in each of the time intervals.

o Interface Ib between the T-LSP manager and the T-TED. When a temporal LSP with a number of time intervals is to be created, through this interface, the T-LSP manager reserves in the T-TED the TE resources such as link bandwidths on every link in each of the time intervals along the path computed for the LSP. When a temporal LSP with a number of time intervals is deleted, the T-LSP manager releases the TE resources such as link bandwidths on every link in each of the time intervals along the path for the LSP.

o Interface Ic between the T-LSP manager and the T-LDB. When a temporal LSP with a number of time intervals is to be created, through this interface, the T-LSP manager reserves in the T-LDB a label for every link in each of the time intervals along the path computed for the LSP. When a temporal LSP with a number of time intervals is deleted, the T-LSP manager releases the label for every link in each of the time intervals along the path for the LSP.

o Interface Id between the T-LSP manager and the T-LSPDB. the T-LSP manager updates the information for every LSP in the T-LSPDB through this interface.

o Interface Ie between the T-CSPF and the T-TED. Through this interface, the T-CSPF accesses the traffic engineering information such as link bandwidths when it computes a path for a temporal LSP with a number of time intervals.

There is an interface In between the TS-SDN controller and the network. In fact, there is a control channel (or interface) between the TS-SDN controller and every node in the network.

Initially, the T-TED obtains the original traffic engineering (TE) information such as link bandwidths from the network through the interface In (i.e., API to network) for every link in the network. The T-LDB gets the original label resources from the network through the interface In for every node and link in the network.

Then the TE information in the T-TED is updated mostly by the following events.

- o When a temporal LSP with a number of time intervals is to be created, the T-LSP manager reserves in the T-TED bandwidths on every link in each of the time intervals along the path for the LSP.

- o When a temporal LSP with a number of time intervals is deleted, the T-LSP manager releases bandwidths on every link in each of the time intervals along the path for the LSP.

- o When a link in the network is down, the TE information about the link is removed from the T-TED.

- o When a link in the network is up, the TE information about the link is added into the T-TED.

The label resources in the T-LDB may be updated as follows:

- o When a temporal LSP with a number of time intervals is to be created, the T-LSP manager reserves in the T-LDB a label for every link in each of the time intervals along the path for the LSP. For a node specific label space, a label on the downstream node is assigned for the link. For a link specific label space, a label on the link is assigned for the link.

- o When a temporal LSP with a number of time intervals is deleted, the T-LSP manager releases the label for every link in each of the time intervals along the path for the LSP.

- o When a node in the network is down, the label resources on the node is removed from the T-LDB if a node specific label space is used. When a link in the network is down, the label resources on the link is removed from the T-LDB if a link specific label space is used.

- o When a node in the network is up, the label resources on the node is added into the T-LDB if a node specific label space is used. When a link in the network is up, the label resources on the link is added into the T-LDB if a link specific label space is used.

There are a couple of ways in which the TS-SDN controller sets up a temporal LSP with a number of time intervals in the network.

One way is to set up the LSP in the network along the path computed for the LSP at the start of each time interval and to delete the LSP at the end of each time interval.

Another way is to set up the LSP in the network along the path computed for the LSP before or at the start of the first time interval, to update the parameters such as bandwidth for each time interval, and to delete the LSP at the end of last time interval.

5.2.2. Centralized Model for Multiple Domains

The centralized model described in the previous section is for temporal LSPs within a single domain, which is called single-domain centralized model. It can be easily extended to support temporal LSPs crossing multiple domains. The extended model is called multi-domain centralized model. Basically, through replacing the T-CSPF module with a T-PCE module in the single-domain centralized model, we obtain a multi-domain centralized model.

Figure below illustrates a centralized SDN controller reference model for temporal LSPs crossing multiple domains.

```
         +--------------------------------------------+
         | TS-SDN Controller                          |
         |                     +---------------+      |
         |        /------------| T-LSP Manager |      |
         |       /     Ia      +---------------+      |
      Im | +--------+           | |  |       \        |
     ----+-+ T-PCE  |   ________| |  |        \Id     |
         | +--------+  /   Ib    /Ic |    +---------+ |
         |       \Ie  /         /    |    | T-LSPDB | |
         |    +---------+ +-------+  |    +---------+ |
         |    |  T-TED  | | T-LDB |  |                |
         |    +---------+ +-------+  |                |
         |          \         \      |In              |
         +---------------API to Network(PCEP+)--------+
                               /       \
                              /         \____
                             /           \   \____
                            /\  .---. .---+       \
                           |  \(     '    |'.---. |
                           |---\  Network |      '+.
                          (o    \         |       | )
                           (     |        |       o)
                            (    |        |       )
                             (   o        o    .-'
                              ,               )
                               '---._.-.     )
                                        ,---'
```

The T-PCE may be outside of the TS-SDN controller. When receiving a request for creating a new temporal LSP with a number of time intervals and some constraints, the TS-SDN controller (or say the T-LSP manager) asks the T-PCE to compute a path for the LSP.

For computing a path for a temporal LSP crossing multiple domains, the T-PCE communicates with other T-PCEs through interface Im to get an end to end path for the LSP crossing domains. For computing a path for a temporal LSP within the network (one domain), the T-PCE uses a T-CSPF inside it to obtain a path for the LSP.

5.2.3. Analysis on Centralized Model

In a centralized model, all the network resources are managed and maintained by a central controller. Thus it has the following advantages:

- o Efficiently use network resources for providing TTS (i.e., finding paths for LSPs with time intervals, reserving the network resources in these intervals and setting up LSPs in the network).
- o Optimal paths for the LSPs with time intervals.

A centralized model may have the following disadvantages or issues:

- o Scalability issues since all the work is done in the controller, which include computing all the paths for the LSPs with time intervals, managing all the network resources, controlling the network and so on.
- o Reliability issues since the failure of the controller will lead to the failure of the whole system.

5.3. Hybrid Model

This section presents a couple of hybrid reference models. one model is for a single domain, the other for multiple domains.

5.3.1. Hybrid Model for Single Domain

Figure below illustrates a hybrid SDN controller reference model for temporal LSP tunnel services within a single domain.

```
+--------------------------------------------+
| TS-SDN Controller                           |
|                     +----------------+      |
|       /-------------| T-LSP Manager  |      |
|      /     Ia       +----------------+      |
| +--------+           |    |       \         |
| | T-CSPF |   ________|    |        \Id      |
| +--------+  /   Ib        |    +----------+ |
|      \Ie  /               |    | T-LSPDB  | |
|   +---------+             |    +----------+ |
|   |  T-TED  |             |                 |
|   +---------+             |                 |
|         \                 |In               |
+----------------API to Network(PCEP)--------+
                      /       \
                     /         \____
                    /               \____
                   /   .---. .---,       \
                  |   (      '     '.---. |
                  |---'  Network        '+.
                 (o                       | )
                  (                       o)
                   (                      )
                    (   o        o    .-'
                     '                )
                      '---._.-.      )
                               '---'
```

The temporal SDN (TS-SDN) controller in this hybrid reference model manages some parts of the resources in the network it controls. For example, it may just manage the link bandwidth for every link in the network. The label resources in the network is not managed by the TS-SDN controller. It may still be managed by each node in the network.

The TS-SDN controller controls the network through an API to the network such as PCEP. There is a control channel between the TS-SDN controller and each of the LERs in the network. The TS-SDN controller is responsible for creating and maintaining every temporal LSP in the network through the control channel to the ingress node of the LSP.

The TS-SDN controller comprises a T-LSP manager, a T-CSPF, a T-TED and a T-LSPDB. The interfaces among these modules are listed as follows:

o Interface Ia between the T-LSP manager and the T-CSPF. This interface is the same as the one between the T-LSP manager and the T-CSPF in the centralized model.

o Interface Ib between the T-LSP manager and the T-TED. This interface is the same as the one between the T-LSP manager and the T-TED in the centralized model.

o Interface Id between the T-LSP manager and the T-LSPDB. This interface is similar to the one between the T-LSP manager and the T-LSPDB in the centralized model. Most of the information for a temporal LSP stored in the T-LSPDB in the hybrid model is the same as that stored in the T-LSPDB in the centralized model. For example, the time intervals associated with the LSP and the link bandwidths reserved for the LSP in each of the time intervals are the same in both models. The labels assigned to the LSP is stored in the T-LSPDB in the centralized model, but there is not any label information for the LSP stored in the T-LSPDB in the hybrid model.

o Interface Ie between the T-CSPF and the T-TED. This interface is the same as the one between the T-CSPF and the T-TED in the centralized model.

The TE information in the T-TED in the hybrid model is updated in the same way as that in the T-TED in the centralized model. But the way in which the T-TED in one model obtains the original TE information from the network may be different from the one in another model.

For example, the T-TED in the centralized model may obtain the original TE information from the network through polling every node in the network. The T-TED in the hybrid model may get the original TE information from the network through an OSPF or ISIS adjacency between the TS-SDN controller and one of the nodes in the network.

There are a few of ways in which the TS-SDN controller sets up a temporal LSP with a number of time intervals in the network.

One way is that the TS-SDN controller asks the ingress of the LSP to signal the LSP in the network along the path computed for the LSP at the start of each time interval and to tear down the LSP at the end of each time interval.

Another way is that the TS-SDN controller asks the ingress of the LSP to signal the LSP in the network along the path computed for the LSP before or at the start of the first time interval, to update the parameters such as bandwidth for each time interval, and to tear down the LSP at the end of the last time interval.

The third way is that the TS-SDN controller asks the ingress of the LSP to signal the LSP in the network along the path computed for the LSP before or at the start of the first time interval, to reserve bandwidth for each time interval, and to tear down the LSP at the end of the last time interval.

5.3.2. Hybrid Model for Multiple Domains

The hybrid model described in the previous section is for temporal LSPs within a single domain, which is called single-domain hybrid model. It can be easily extended to support temporal LSPs crossing multiple domains. The extended model is called multi-domain hybrid model. Basically, through replacing the T-CSPF module with a T-PCE module in the single-domain hybrid model, we obtain a multi-domain hybrid model.

Figure below illustrates a hybrid SDN controller reference model for temporal LSPs crossing multiple domains.

```
            +--------------------------------------------+
            | TS-SDN Controller                          |
            |                     +---------------+      |
            |        /------------| T-LSP Manager |      |
            |       /     Ia      +---------------+      |
         Im | +--------+           |    |       \        |
        ----+-+ T-PCE  |   ________|    |        \Id     |
            | +--------+  /   Ib        |    +---------+ |
            |       \Ie  /              |    | T-LSPDB | |
            |    +---------+            |    +---------+ |
            |    |  T-TED  |            |                |
            |    +---------+            |                |
            |          \                |In              |
            +----------------API to Network(PCEP)--------+
                               /       \
                              /         \_____
                             /               \_____
                            /   .---. .---,       \
                           |   (     '     '.---. |
                           |---'  Network        '+.
                          (o                      | )
                           (                      o)
                            (                     )
                             (   o        o    .-'
                              '               )
                               '---._.-.     )
                                        '---'
```

The T-PCE may be outside of the TS-SDN controller. When receiving a request for creating a new temporal LSP with a number of time intervals and some constraints, the TS-SDN controller (or say the T-LSP manager) asks the T-PCE to compute a path for the LSP.

For computing a path for a temporal LSP crossing multiple domains, the T-PCE communicates with other T-PCEs through interface Im to get an end to end path for the LSP crossing domains. For computing a path for a temporal LSP within the network (one domain), the T-PCE uses a T-CSPF inside it to obtain a path for the LSP.

5.3.3. Temporal Stateful PCE

From the multi-domain hybrid model described in the previous section, we can get a temporal stateful PCE (controller) if we uses the stateful PCEP as the interface between the temporal stateful PCE (TS-Stateful-PCE for short) controller and the network on which the PCE controls.

Figure below illustrates a temporal stateful PCE controller reference model.

```
            +---------------------------------------------+
            | TS-Stateful-PCE (Controller)                |
            |                     +---------------+       |
            |        /------------| T-LSP Manager |       |
            |       /     Ia      +---------------+       |
         Im | +--------+           |           \          |
        ----+-+ T-PCE  |   ________|            \Id       |
            | +--------+  /   Ib             +---------+ |
            |       \Ie  /                   | T-LSPDB | |
            |    +---------+                 +---------+ |
            |    |  T-TED  |                             |
            |    +---------+                             |
            |          \                                 |
            +---------------- Stateful PCEP -------------+
                               /       \
                              /         \_____
                             /               \_____
                            /   .---. .---,       \
                           |   (     '     '.---. |
                           |---'  Network        '+.
                          (o                      | )
                           (                      o)
                            (                     )
                             (   o        o    .-'
                              ,               )
                               '---._.-.     )
```

'---'

The T-PCE may be outside of the TS-Stateful PCE controller. When receiving a request for creating a new temporal LSP with a number of time intervals and some constraints, the TS-Stateful PCE (controller) asks the T-PCE to compute a path for the LSP.

For computing a path for a temporal LSP crossing multiple domains, the T-PCE communicates with other T-PCEs through interface Im to get an end to end path for the LSP crossing domains. For computing a path for a temporal LSP within the network (one domain), the T-PCE uses a T-CSPF inside it to obtain a path for the LSP.

After obtaining the path for the LSP, the TS-Statefule PCE (controller) reserves in the T-TED the TE resources such as link bandwidths for the LSP along the path in each of the time intervals, updates in the T-LSPDB the information about the LSP, initiates the creation of the LSP at the start of each time interval through sending a Path Computation LSP Initiate Request (PCInitiate) message to the ingress of the LSP, and deletes the LSP at the end of each time interval through sending another PCInitiate message with R (remove) flag set to 1.

The TS-Statefule PCE (controller) updates the information about the LSP in the T-LSPDB accordingly after receiving a Path Computation LSP State Report (PCRpt) message from the ingress of the LSP.

5.3.4. Analysis on Hybrid Model

In a hybrid model, some of the network resources are managed and maintained by a central controller. Thus it has the following advantages:

o Efficiently use network resources for providing TTS (i.e., finding paths for LSPs with time intervals, reserving the network resources in these intervals and setting up LSPs in the network).

o Optimal paths for the LSPs with time intervals.

A hybrid model may have the following disadvantages or issues:

o Reliability issues since the failure of the controller will lead to the failure of the whole system.

5.4. Distributed Model

5.4.1. Router Distributed Model

Figure below illustrates a traditional/router distributed reference model for temporal LSP tunnel services.

```
      +----------------------------------------------------+
      | Router                                             |
      |  +-----------+  +-------------------------------+  |
      |  | T-OSPF    |  | TS-MPLS                       |  |
   In |  |(ISIS)     |  |          Ia +---------------+ |  |
   ---+--+           |  |        /----+ T-LSP Manager | |  |
      |  |           |  |       /     +----+-----+----+ |  |
      |  |           |  | +----+---+       |     |      |  |
      |  +-----+-----+  | | T-CSPF |     __|Ir   |Id    |  |
      |        |        | +----+---+    /    +---+----+ |  |
      |        |Ig      |      |       /     |T-LSPDB | |  |
      |        |_       |   Ie/       /      +--------+ |  |
      |          \      |    /  +----+-----+            |  | In
      |           \     |   /   |T-RSVP-TE +------------+--+----
      |            \    |  /    +----------+            |  |
      |             \   +-+-----------------------------+  |
      |              \   /                                 |
      |           +---+-+---+                              |
      |           |  T-TED  |                              |
      |           +---------+                              |
      +----------------------------------------------------+
```

In an existing distributed network, the existing MPLS and OSPF/ISIS running on every node in the network need to be enhanced to support temporal LSP tunnel services.

The enhanced OSPF is represented by T-OSPF in the distributed model. The T-OSPF running on a router distributes the traffic engineering (TE) information such as bandwidth about every link connected to the router in a series of time intervals. It also receives the TE information about a link in a number of time intervals from a router in the network. It updates the TE information about every link in the network in the T-TED. The T-OSPF distributes and receives the TE information about a link through interface In connecting to another router.

The T-TED stores the TE information about every link in the network. It is updated by the T-OSPF through interface Ig when the T-OSPF receives the TE information about a link that is changed or the TE information about a link connected to the router is changed.

The T-CSPF in the distributed model has the same function as the one in the other two models. It computes a path for a temporal LSP using the T-TED. It accesses the T-TED through interface Ie.

The enhanced RSVP-TE for supporting temporal LSP tunnel services is represented by T-RSVP-TE. The T-RSVP-TE running on every node along the path for a temporal LSP signals and maintains the LSP with time intervals. The signaling messages for the LSP is sent and received through interface In connecting to another router.

The T-LSP manager on an LER receives a request to create, delete or update a temporal LSP with a number of time intervals.

When receiving a request for creating a new temporal LSP with a sequence of time intervals and constraints, the T-LSP manager asks the T-CSPF to compute a path for the LSP that satisfies the constraints for the LSP in each of the time intervals.

After obtaining the path for the LSP from the T-CSPF, the T-LSP manager requests T-RSVP-TE to signal the LSP along the path with the time intervals.

The T-LSP manager records the related information for the LSP into the temporal LSP database (T-LSPDB). The information includes the time intervals for the LSP, the path computed for the LSP, and the status of the LSP.

5.4.2. Analysis on Distributed Model

In a distributed model, the network resources are managed and maintained by multiple routers. Thus it has the following advantages:

- o More reliable. When one router fails, the system continues working.

- o More scalable for path computations since the path computation load is distributed among the multiple routers.

A distributed model may have the following disadvantages or issues:

- o Sub-optimal paths for the LSPs with time intervals since a controller or router may not have the latest accurate information about the network resources.

- o Network resources may not be used efficiently.
- o Scalability issues for the distribution of link bandwidth with time intervals by IGP in the network, which may consume lots of network resources such as memory and link bandwidth.

6. Security Considerations

The mechanism described in this document does not raise any new security issues.

7. Acknowledgement

The authors would like to thank every one who gives his valuable comments on this draft.

8. References

8.1. Normative References

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

[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>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <http://www.rfc-editor.org/info/rfc5440>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The

OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <http://www.rfc-editor.org/info/rfc5250>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <http://www.rfc-editor.org/info/rfc3630>.

[I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-11 (work in progress) , April 2015.

[I-D.ietf-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-04 (work in progress) , April 2015.

8.2. Informative References

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC5862] Yasukawa, S. and A. Farrel, "Path Computation Clients (PCC) - Path Computation Element (PCE) Requirements for Point-to-Multipoint MPLS-TE", RFC 5862, DOI 10.17487/RFC5862, June 2010, <http://www.rfc-editor.org/info/rfc5862>.

[RFC6006] Zhao, Q., Ed., King, D., Ed., Verhaeghe, F., Takeda, T., Ali, Z., and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 6006, DOI 10.17487/RFC6006, September 2010, <http://www.rfc-editor.org/info/rfc6006>.

[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, <http://www.rfc-editor.org/info/rfc4875>.

Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
US

Email: huaimo.chen@huawei.com

Mehmet Toy
Comcast
1800 Bishops Gate Blvd.
Mount Laurel, NJ 08054
USA

Email: mehmet_toy@cable.comcast.com

Lei Liu
Fujitsu
USA

Email: lliu@us.fujitsu.com

Khuzema Pithewan
Infinera

Email: kpithewan@infinera.com

Internet Engineering Task Force M. Chen
Internet-Draft Huawei
Intended status: Standards Track L. Ginsberg
Expires: April 16, 2016 S. Previdi
Cisco Systems
October 14, 2015

ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering
draft-chen-teas-rfc5316bis-00

Abstract

This document describes extensions to the ISIS (ISIS) protocol to support Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE) for multiple Autonomous Systems (ASes). It defines ISIS-TE extensions for the flooding of TE information about inter-AS links, which can be used to perform inter-AS TE path computation.

No support for flooding information from within one AS to another AS is proposed or defined in this document.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC5305] defines extensions to the ISIS protocol [RFC1195] to support intra-area Traffic Engineering (TE). The extensions provide a way of encoding the TE information for TE-enabled links within the network (TE links) and flooding this information within an area. The extended IS reachability TLV and traffic engineering router ID TLV, which are defined in [RFC5305], are used to carry such TE information. The extended IS reachability TLV has several nested sub-TLVs that describe the TE attributes for a TE link.

[RFC6119] and [RFC5307] define similar extensions to ISIS in support of IPv6 and Generalized Multiprotocol Label Switching (GMPLS) TE respectively.

Requirements for establishing Multiprotocol Label Switching (MPLS) TE Label Switched Paths (LSPs) that cross multiple Autonomous Systems (ASes) are described in [RFC4216]. As described in [RFC4216], a method SHOULD provide the ability to compute a path spanning multiple ASes. So a path computation entity that may be the head-end Label Switching Router (LSR), an AS Border Router (ASBR), or a Path Computation Element (PCE) [RFC4655] needs to know the TE information not only of the links within an AS, but also of the links that connect to other ASes.

In this document, a new TLV, which is referred to as the inter-AS reachability TLV, is defined to advertise inter-AS TE information, three new sub-TLVs are defined for inclusion in the inter-AS reachability TLV to carry the information about the remote AS number and remote ASBR ID. The sub-TLVs defined in [RFC5305][RFC6119] and other documents for inclusion in the extended IS reachability TLV for describing the TE properties of a TE link are applicable to be included in the Inter-AS Reachability TLV for describing the TE properties of an inter-AS TE link as well. Also, two more new sub-TLVs are defined for inclusion in the IS-IS router capability TLV to carry the TE Router ID when the TE Router ID needs to reach all routers within an entire ISIS routing domain. The extensions are equally applicable to IPv4 and IPv6 as identical extensions to [RFC5305] and [RFC6119]. Detailed definitions and procedures are discussed in the following sections.

This document does not propose or define any mechanisms to advertise any other extra-AS TE information within ISIS. See Section 2.1 for a full list of non-objectives for this work.

2. Problem Statement

As described in [RFC4216], in the case of establishing an inter-AS TE LSP that traverses multiple ASes, the Path message [RFC3209] may include the following elements in the Explicit Route Object (ERO) in order to describe the path of the LSP:

o a set of AS numbers as loose hops; and/or

o a set of LSRs including ASBRs as loose hops.

Two methods for determining inter-AS paths are currently being discussed. The per-domain method [RFC5152] determines the path one domain at a time. The backward recursive method [RFC5441] uses cooperation between PCEs to determine an optimum inter-domain path. The sections that follow examine how inter-AS TE link information could be useful in both cases.

2.1. A Note on Non-Objectives

It is important to note that this document does not make any change to the confidentiality and scaling assumptions surrounding the use of ASes in the Internet. In particular, this document is conformant to the requirements set out in [RFC4216].

The following features are explicitly excluded:

o There is no attempt to distribute TE information from within one AS to another AS.

o There is no mechanism proposed to distribute any form of TE reachability information for destinations outside the AS.

o There is no proposed change to the PCE architecture or usage.

o TE aggregation is not supported or recommended.

o There is no exchange of private information between ASes.

o No ISIS adjacencies are formed on the inter-AS link.

2.2. Per-Domain Path Determination

In the per-domain method of determining an inter-AS path for an MPLS-TE LSP, when an LSR that is an entry-point to an AS receives a Path message from an upstream AS with an ERO containing a next hop that is an AS number, it needs to find which LSRs (ASBRs) within the local AS are connected to the downstream AS. That way, it can compute a TE

LSP segment across the local AS to one of those LSRs and forward the Path message to that LSR and hence into the next AS. See Figure 1 for an example.

```
R1------R3----R5-----R7------R9-----R11
        |     | \    |      / |
        |     |  \   |  ----  |
        |     |   \  | /      |
R2------R4----R6   --R8------R10----R12
           :              :
<-- AS1 -->:<---- AS2 --->:<--- AS3 --->
```

Figure 1: Inter-AS Reference Model

The figure shows three ASes (AS1, AS2, and AS3) and twelve LSRs (R1 through R12). R3 and R4 are ASBRs in AS1. R5, R6, R7, and R8 are ASBRs in AS2. R9 and R10 are ASBRs in AS3.

If an inter-AS TE LSP is planned to be established from R1 to R12, the AS sequence will be: AS1, AS2, AS3.

Suppose that the Path message enters AS2 from R3. The next hop in the ERO shows AS3, and R5 must determine a path segment across AS2 to reach AS3. It has a choice of three exit points from AS2 (R6, R7, and R8), and it needs to know which of these provide TE connectivity to AS3, and whether the TE connectivity (for example, available bandwidth) is adequate for the requested LSP.

Alternatively, if the next hop in the ERO is the entry ASBR for AS3 (say R9), R5 needs to know which of its exit ASBRs has a TE link that connects to R9. Since there may be multiple ASBRs that are connected to R9 (both R7 and R8 in this example), R5 also needs to know the TE properties of the inter-AS TE links so that it can select the correct exit ASBR.

Once the Path message reaches the exit ASBR, any choice of inter-AS TE link can be made by the ASBR if not already made by the entry ASBR that computed the segment.

More details can be found in Section 4 of [RFC5152], which clearly points out why advertising of inter-AS links is desired.

To enable R5 to make the correct choice of exit ASBR, the following information is needed:

o List of all inter-AS TE links for the local AS.

o TE properties of each inter-AS TE link.

- o AS number of the neighboring AS connected to by each inter-AS TE link.

- o Identity (TE Router ID) of the neighboring ASBR connected to by each inter-AS TE link.

In GMPLS networks, further information may also be required to select the correct TE links as defined in [RFC5307].

The example above shows how this information is needed at the entry-point ASBRs for each AS (or the PCEs that provide computation services for the ASBRs). However, this information is also needed throughout the local AS if path computation functionality is fully distributed among LSRs in the local AS, for example to support LSPs that have start points (ingress nodes) within the AS.

2.3. Backward Recursive Path Computation

Another scenario using PCE techniques has the same problem. [RFC5441] defines a PCE-based TE LSP computation method (called Backward Recursive Path Computation) to compute optimal inter-domain constrained MPLS-TE or GMPLS LSPs. In this path computation method, a specific set of traversed domains (ASes) are assumed to be selected before computation starts. Each downstream PCE in domain(i) returns to its upstream neighbor PCE in domain(i-1) a multipoint-to-point tree of potential paths. Each tree consists of the set of paths from all boundary nodes located in domain(i) to the destination where each path satisfies the set of required constraints for the TE LSP (bandwidth, affinities, etc.).

So a PCE needs to select boundary nodes (that is, ASBRs) that provide connectivity from the upstream AS. In order for the tree of paths provided by one PCE to its neighbor to be correlated, the identities of the ASBRs for each path need to be referenced. Thus, the PCE must know the identities of the ASBRs in the remote AS that are reached by any inter-AS TE link, and, in order to provide only suitable paths in the tree, the PCE must know the TE properties of the inter-AS TE links. See the following figure as an example.

```
                  PCE1<------>PCE2<-------->PCE3
                  /       :             :
                 /        :             :
               R1------R3----R5-----R7------R9-----R11
                        |     | \    |      / |
                        |     |  \   |  ----  |
                        |     |   \  | /      |
               R2------R4----R6   --R8------R10----R12
                          :              :
               <-- AS1 -->:<---- AS2 --->:<--- AS3 --->
```

Figure 2: BRPC for Inter-AS Reference Model

The figure shows three ASes (AS1, AS2, and AS3), three PCEs (PCE1, PCE2, and PCE3), and twelve LSRs (R1 through R12). R3 and R4 are ASBRs in AS1. R5, R6, R7, and R8 are ASBRs in AS2. R9 and R10 are ASBRs in AS3. PCE1, PCE2, and PCE3 cooperate to perform inter-AS path computation and are responsible for path segment computation within their own domain(s).

If an inter-AS TE LSP is planned to be established from R1 to R12, the traversed domains are assumed to be selected: AS1->AS2->AS3, and the PCE chain is: PCE1->PCE2->PCE3. First, the path computation request originated from the PCC (R1) is relayed by PCE1 and PCE2 along the PCE chain to PCE3. Then, PCE3 begins to compute the path segments from the entry boundary nodes that provide connection from AS2 to the destination (R12). But, to provide suitable path segments, PCE3 must determine which entry boundary nodes provide connectivity to its upstream neighbor AS (identified by its AS number), and must know the TE properties of the inter-AS TE links. In the same way, PCE2 also needs to determine the entry boundary nodes according to its upstream neighbor AS and the inter-AS TE link capabilities.

Thus, to support Backward Recursive Path Computation, the same information listed in Section 2.2 is required. The AS number of the neighboring AS connected to by each inter-AS TE link is particularly important.

3. Extensions to ISIS-TE

Note that this document does not define mechanisms for distribution of TE information from one AS to another, does not distribute any form of TE reachability information for destinations outside the AS, does not change the PCE architecture or usage, does not suggest or recommend any form of TE aggregation, and does not feed private information between ASes. See Section 2.1.

In this document, for the advertisement of inter-AS TE links, a new TLV, which is referred to as the inter-AS reachability TLV, is defined. Three new sub-TLVs are also defined for inclusion in the inter-AS reachability TLV to carry the information about the neighboring AS number and the remote ASBR ID of an inter-AS link. The sub-TLVs defined in [RFC5305], [RFC6119], and other documents for inclusion in the extended IS reachability TLV are applicable to be included in the inter-AS reachability TLV for inter-AS TE links advertisement. Also, two other new sub-TLVs are defined for inclusion in the IS-IS router capability TLV to carry the TE Router ID when the TE Router ID is needed to reach all routers within an entire ISIS routing domain.

While some of the TE information of an inter-AS TE link may be available within the AS from other protocols, in order to avoid any dependency on where such protocols are processed, this mechanism carries all the information needed for the required TE operations.

3.1. Inter-AS Reachability TLV

The inter-AS reachability TLV has type 141 (see Section 6.1) and contains a data structure consisting of:

```
4 octets of Router ID
3 octets of default metric
1 octet of control information, consisting of:
   1 bit of flooding-scope information (S bit)
   1 bit of up/down information (D bit)
   6 bits reserved
1 octet of length of sub-TLVs
0-246 octets of sub-TLVs, where each sub-TLV consists of a sequence of:
      1 octet of sub-type
      1 octet of length of the value field of the sub-TLV
      0-244 octets of value
```

Compared to the extended reachability TLV which is defined in [RFC5305], the inter-AS reachability TLV replaces the "7 octets of System ID and Pseudonode Number" field with a "4 octets of Router ID" field and introduces an extra "control information" field, which consists of a flooding-scope bit (S bit), an up/down bit (D bit), and 6 reserved bits.

The Router ID field of the inter-AS reachability TLV is 4 octets in length, which contains the IPv4 Router ID of the router who generates the inter-AS reachability TLV. The Router ID SHOULD be identical to the value advertised in the Traffic Engineering Router ID TLV [RFC5305]. If no Traffic Engineering Router ID is assigned, the Router ID SHOULD be identical to an IP Interface Address [RFC1195] advertised by the originating IS. If the originating node does not support IPv4, then the reserved value 0.0.0.0 MUST be used in the Router ID field and the IPv6 Router ID sub-TLV MUST be present in the inter-AS reachability TLV.

The flooding procedures for inter-AS reachability TLV are identical to the flooding procedures for the GENINFO TLV, which are defined in Section 4 of [RFC6823]. These procedures have been previously discussed in [RFC4971]. The flooding-scope bit (S bit) SHOULD be set to 0 if the flooding scope is to be limited to within the single IGP area to which the ASBR belongs. It MAY be set to 1 if the information is intended to reach all routers (including area border routers, ASBRs, and PCEs) in the entire ISIS routing domain. The choice between the use of 0 or 1 is an AS-wide policy choice, and configuration control SHOULD be provided in ASBR implementations that support the advertisement of inter-AS TE links.

The sub-TLVs defined in [RFC5305], [RFC6119], and other documents for describing the TE properties of a TE link are also applicable to the inter-AS reachability TLV for describing the TE properties of an Inter-AS TE link. Apart from these sub-TLVs, four new sub-TLVs are defined for inclusion in the inter-AS reachability TLV defined in this document:

Sub-TLV type	Length	Name
24	4	remote AS number
25	4	IPv4 remote ASBR identifier
26	16	IPv6 remote ASBR identifier
TBD1	16	IPv6 Router ID

Detailed definitions of the three new sub-TLVs are described in Section 3.3.1, 3.3.2, 3.3.3, and 3.3.4.

3.2. TE Router ID

The IPv4 TE Router ID TLV and IPv6 TE Router ID TLV, which are defined in [RFC5305] and [RFC6119] respectively, only have area flooding-scope. When performing inter-AS TE, the TE Router ID MAY be needed to reach all routers within an entire ISIS routing domain and it MUST have the same flooding scope as the Inter-AS Reachability TLV does.

[RFC4971] defines a generic advertisement mechanism for ISIS which allows a router to advertise its capabilities within an ISIS area or an entire ISIS routing domain. [RFC4971] also points out that the TE Router ID is a candidate to be carried in the IS-IS router capability TLV when performing inter-area TE.

This document uses such mechanism for TE Router ID advertisement when the TE Router ID is needed to reach all routers within an entire ISIS Routing domain. Two new sub-TLVs are defined for inclusion in the IS-IS Router Capability TLV to carry the TE Router IDs.

Sub-TLV type	Length	Name
11	4	IPv4 TE Router ID
12	16	IPv6 TE Router ID

Detailed definitions of the new sub-TLV are described in Section 3.4.1 and 3.4.2.

3.3. Sub-TLVs for Inter-AS Reachability TLV

3.3.1. Remote AS Number Sub-TLV

A new sub-TLV, the remote AS number sub-TLV, is defined for inclusion in the inter-AS reachability TLV when advertising inter-AS links. The remote AS number sub-TLV specifies the AS number of the neighboring AS to which the advertised link connects.

The remote AS number sub-TLV is TLV type 24 (see Section 6.2) and is 4 octets in length. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Remote AS Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The remote AS number field has 4 octets. When only 2 octets are used for the AS number, as in current deployments, the left (high-order) 2 octets MUST be set to 0. The remote AS number sub-TLV MUST be included when a router advertises an inter-AS TE link.

3.3.2. IPv4 Remote ASBR ID Sub-TLV

A new sub-TLV, which is referred to as the IPv4 remote ASBR ID sub-TLV, is defined for inclusion in the inter-AS reachability TLV when advertising inter-AS links. The IPv4 remote ASBR ID sub-TLV specifies the IPv4 identifier of the remote ASBR to which the advertised inter-AS link connects. This could be any stable and routable IPv4 address of the remote ASBR. Use of the TE Router ID as specified in the Traffic Engineering router ID TLV [RFC5305] is RECOMMENDED.

The IPv4 remote ASBR ID sub-TLV is TLV type 25 (see Section 6.2) and is 4 octets in length. The format of the IPv4 remote ASBR ID sub-TLV is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote ASBR ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv4 remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv4 address. If the neighboring ASBR does not have an IPv4 address (not even an IPv4 TE Router ID), the IPv6 remote ASBR ID sub-TLV MUST be included instead. An IPv4 remote ASBR ID sub-TLV and IPv6 remote ASBR ID sub-TLV MAY both be present in an extended IS reachability TLV.

3.3.3. IPv6 Remote ASBR ID Sub-TLV

A new sub-TLV, which is referred to as the IPv6 remote ASBR ID sub-TLV, is defined for inclusion in the inter-AS reachability TLV when advertising inter-AS links. The IPv6 remote ASBR ID sub-TLV specifies the IPv6 identifier of the remote ASBR to which the advertised inter-AS link connects. This could be any stable and routable IPv6 address of the remote ASBR. Use of the TE Router ID as specified in the IPv6 Traffic Engineering router ID TLV [RFC6119] is RECOMMENDED.

The IPv6 remote ASBR ID sub-TLV is TLV type 26 (see Section 6.2) and is 16 octets in length. The format of the IPv6 remote ASBR ID sub-TLV is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote ASBR ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote ASBR ID (continued)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote ASBR ID (continued)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Remote ASBR ID (continued)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv6 remote ASBR ID sub-TLV MUST be included if the neighboring ASBR has an IPv6 address. If the neighboring ASBR does not have an IPv6 address, the IPv4 remote ASBR ID sub-TLV MUST be included instead. An IPv4 remote ASBR ID sub-TLV and IPv6 remote ASBR ID sub-TLV MAY both be present in an extended IS reachability TLV.

3.3.4. IPv6 Router ID sub-TLV

The IPv6 Router ID sub-TLV is TLV type TBD1 (see Section 6.3) and is 16 octets in length. The format of the IPv6 Router ID sub-TLV is as follows:

```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |              Type               |              Length         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       IPv6 Router ID                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       IPv6 Router ID   (continued)              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       IPv6 Router ID   (continued)              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       IPv6 Router ID   (continued)              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv6 TE Router ID SHOULD be identical to the value advertised in the IPv6 Traffic Engineering Router ID TLV [RFC6119].

If the originating node does not support IPv4, the IPv6 Router ID sub-TLV MUST be present in the inter-AS reachability TLV. Inter-AS reachability TLVs which have a Router ID of 0.0.0.0 and do NOT have the IPv6 Router ID sub-TLV present MUST be ignored.

3.4. Sub-TLVs for IS-IS Router Capability TLV

3.4.1. IPv4 TE Router ID sub-TLV

The IPv4 TE Router ID sub-TLV is TLV type 11 (see Section 6.3) and is 4 octets in length. The format of the IPv4 TE Router ID sub-TLV is as follows:

```
    0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        TE Router ID                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv4 TE Router ID SHOULD be identical to the value advertised in the IPv4 Traffic Engineering Router ID TLV [RFC5305].

When the TE Router ID is needed to reach all routers within an entire ISIS routing domain, the IS-IS Router capability TLV MUST be included in its LSP. If an ASBR supports Traffic Engineering for IPv4 and if the ASBR has an IPv4 TE Router ID, the IPv4 TE Router ID sub-TLV MUST be included. If the ASBR does not have an IPv4 TE Router ID, the IPv6 TE Router sub-TLV MUST be included instead. An IPv4 TE Router ID sub-TLV and IPv6 TE Router ID sub-TLV MAY both be present in an IS-IS router capability TLV.

3.4.2. IPv6 TE Router ID sub-TLV

The IPv6 TE Router ID sub-TLV is TLV type 12 (see Section 6.3) and is 16 octets in length. The format of the IPv6 TE Router ID sub-TLV is as follows:

```
    0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |              Type             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        TE Router ID                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        TE Router ID   (continued)             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        TE Router ID   (continued)             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                        TE Router ID   (continued)             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv6 TE Router ID SHOULD be identical to the value advertised in the IPv6 Traffic Engineering Router ID TLV [RFC6119].

When the TE Router ID is needed to reach all routers within an entire ISIS routing domain, the IS-IS router capability TLV MUST be included in its LSP. If an ASBR supports Traffic Engineering for IPv6 and if the ASBR has an IPv6 TE Router ID, the IPv6 TE Router ID sub-TLV MUST be included. If the ASBR does not have an IPv6 TE Router ID, the

IPv4 TE Router sub-TLV MUST be included instead. An IPv4 TE Router ID sub-TLV and IPv6 TE Router ID sub-TLV MAY both be present in an IS-IS router capability TLV.

4. Procedure for Inter-AS TE Links

When TE is enabled on an inter-AS link and the link is up, the ASBR SHOULD advertise this link using the normal procedures for [RFC5305]. When either the link is down or TE is disabled on the link, the ASBR SHOULD withdraw the advertisement. When there are changes to the TE parameters for the link (for example, when the available bandwidth changes), the ASBR SHOULD re-advertise the link but MUST take precautions against excessive re-advertisements.

Hellos MUST NOT be exchanged over the inter-AS link, and consequently, an ISIS adjacency MUST NOT be formed.

The information advertised comes from the ASBR's knowledge of the TE capabilities of the link, the ASBR's knowledge of the current status and usage of the link, and configuration at the ASBR of the remote AS number and remote ASBR TE Router ID.

Legacy routers receiving an advertisement for an inter-AS TE link are able to ignore it because they do not know the new TLV and sub-TLVs that are defined in Section 3 of this document. They will continue to flood the LSP, but will not attempt to use the information received.

In the current operation of ISIS TE, the LSRs at each end of a TE link emit LSPs describing the link. The databases in the LSRs then have two entries (one locally generated, the other from the peer) that describe the different 'directions' of the link. This enables Constrained Shortest Path First (CSPF) to do a two-way check on the link when performing path computation and eliminate it from consideration unless both directions of the link satisfy the required constraints.

In the case we are considering here (i.e., of a TE link to another AS), there is, by definition, no IGP peering and hence no bidirectional TE link information. In order for the CSPF route computation entity to include the link as a candidate path, we have to find a way to get LSPs describing its (bidirectional) TE properties into the TE database.

This is achieved by the ASBR advertising, internally to its AS, information about both directions of the TE link to the next AS. The ASBR will normally generate a LSP describing its own side of a link; here we have it 'proxy' for the ASBR at the edge of the other AS and generate an additional LSP that describes that device's 'view' of the link.

Only some essential TE information for the link needs to be advertised; i.e., the Interface Address, the remote AS number, and the remote ASBR ID of an inter-AS TE link.

Routers or PCEs that are capable of processing advertisements of inter-AS TE links SHOULD NOT use such links to compute paths that exit an AS to a remote ASBR and then immediately re-enter the AS through another TE link. Such paths would constitute extremely rare occurrences and SHOULD NOT be allowed except as the result of specific policy configurations at the router or PCE computing the path.

4.1. Origin of Proxied TE Information

Section 4 describes how an ASBR advertises TE link information as a proxy for its neighbor ASBR, but does not describe where this information comes from.

Although the source of this information is outside the scope of this document, it is possible that it will be a configuration requirement at the ASBR, as are other local properties of the TE link. Further, where BGP is used to exchange IP routing information between the ASBRs, a certain amount of additional local configuration about the link and the remote ASBR is likely to be available.

We note further that it is possible, and may be operationally advantageous, to obtain some of the required configuration information from BGP. Whether and how to utilize these possibilities is an implementation matter.

5. Security Considerations

The protocol extensions defined in this document are relatively minor and can be secured within the AS in which they are used by the existing ISIS security mechanisms (e.g., using the cleartext passwords or Hashed Message Authentication Codes - Message Digest 5 (HMAC-MD5) algorithm, which are defined in [RFC1195] and [RFC3567] separately).

There is no exchange of information between ASes, and no change to the ISIS security relationship between the ASes. In particular, since no ISIS adjacency is formed on the inter-AS links, there is no requirement for ISIS security between the ASes.

Some of the information included in these new advertisements (e.g., the remote AS number and the remote ASBR ID) is obtained manually from a neighboring administration as part of a commercial relationship. The source and content of this information should be carefully checked before it is entered as configuration information at the ASBR responsible for advertising the inter-AS TE links.

It is worth noting that in the scenario we are considering, a Border Gateway Protocol (BGP) peering may exist between the two ASBRs and that this could be used to detect inconsistencies in configuration (e.g., the administration that originally supplied the information may be lying, or some manual mis-configurations or mistakes may be made by the operators). For example, if a different remote AS number is received in a BGP OPEN [RFC4271] from that locally configured to ISIS-TE, as we describe here, then local policy SHOULD be applied to determine whether to alert the operator to a potential mis-configuration or to suppress the ISIS advertisement of the inter-AS TE link. Note further that if BGP is used to exchange TE information as described in Section 4.1, the inter-AS BGP session SHOULD be secured using mechanisms as described in [RFC4271] to provide authentication and integrity checks.

For a discussion of general security considerations for IS-IS, see [RFC5304].

6. IANA Considerations

IANA is requested to make the following allocations from registries under its control.

6.1. Inter-AS Reachability TLV

This document defines the following new ISIS TLV type, described in Section 3.1, which has been registered in the ISIS TLV codepoint registry:

Type	Description	IIH	LSP	SNP
141	inter-AS reachability information	n	y	n

6.2. Sub-TLVs for the Inter-AS Reachability TLV

This document defines the following new sub-TLV types (described in Sections 3.3.1, 3.3.2, 3.3.3, and, 3.3.4) of top-level TLV 141 (see Section 6.1 above), which have been registered in the ISIS sub-TLV registry for TLV 141. Note that these four new sub-TLVs SHOULD NOT appear in TLV 22 (or TLV 23, TLV 222, TLV223) and MUST be ignored in TLV 22 (or TLV 23, TLV 222, TLV223):

Type	Description
24	remote AS number
25	IPv4 remote ASBR identifier
26	IPv6 remote ASBR identifier
TBD1	IPv6 Router ID

As described above in Section 3.1, the sub-TLVs which are defined in [RFC5305], [RFC6119] and other documents for describing the TE properties of an TE link are applicable to describe an inter-AS TE link and MAY be included in the inter-AS reachability TLV when adverting inter-AS TE links.

IANA has created the following sub-TLVs registries in "Sub-TLVs for TLVs 22, 23, 141, 222, and 223" registry.

Type	Description	TLV 22	TLV 23	TLV 141	TLV 222	TLV 223	Reference
24	remote AS number	n	n	y	n	n	[This.I-D]
25	IPv4 remote ASBR identifier	n	n	y	n	n	[This.I-D]
26	IPv6 remote ASBR identifier	n	n	y	n	n	[This.I-D]

IANA is requested to create a new sub-TLV registry in "Sub-TLVs for TLVs 22, 23, 141, 222, and 223" registry.

Type	Description	TLV 22	TLV 23	TLV 141	TLV 222	TLV 223	Reference
TBD1	IPv6 Router ID	n	n	y	n	n	[This.I-D]

6.3. Sub-TLVs for the IS-IS Router Capability TLV

This document defines the following new sub-TLV types, described in Sections 3.4.1 and 3.4.2, of top-level TLV 242 (which is defined in [RFC4971]) that have been registered in the ISIS sub-TLV registry for TLV 242:

Type	Description	Length
11	IPv4 TE Router ID	4
12	IPv6 TE Router ID	16

7. Acknowledgements

For the original version of [RFC5316] the authors would like to thank Adrian Farrel, Jean-Louis Le Roux, Christian Hopps, Les Ginsberg, and Hannes Gredler for their review and comments on this document.

8. References

8.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, DOI 10.17487/RFC1195, December 1990, <http://www.rfc-editor.org/info/rfc1195>.

[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>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <http://www.rfc-editor.org/info/rfc5305>.

[RFC6119] Harrison, J., Berger, J., and M. Bartlett, "IPv6 Traffic Engineering in IS-IS", RFC 6119, DOI 10.17487/RFC6119, February 2011, <http://www.rfc-editor.org/info/rfc6119>.

8.2. 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, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3567] Li, T. and R. Atkinson, "Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication", RFC 3567, DOI 10.17487/RFC3567, July 2003, <http://www.rfc-editor.org/info/rfc3567>.

[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, <http://www.rfc-editor.org/info/rfc4206>.

[RFC4216] Zhang, R., Ed. and J. Vasseur, Ed., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, DOI 10.17487/RFC4216, November 2005, <http://www.rfc-editor.org/info/rfc4216>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC4971] Vasseur, JP., Ed., Shen, N., Ed., and R. Aggarwal, Ed., "Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information", RFC 4971, DOI 10.17487/RFC4971, July 2007, <http://www.rfc-editor.org/info/rfc4971>.

[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, <http://www.rfc-editor.org/info/rfc5152>.

[RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", RFC 5304, DOI 10.17487/RFC5304, October 2008, <http://www.rfc-editor.org/info/rfc5304>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <http://www.rfc-editor.org/info/rfc5307>.

[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, <http://www.rfc-editor.org/info/rfc5316>.

[RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <http://www.rfc-editor.org/info/rfc5441>.

[RFC6823] Ginsberg, L., Previdi, S., and M. Shand, "Advertising Generic Information in IS-IS", RFC 6823, DOI 10.17487/RFC6823, December 2012, <http://www.rfc-editor.org/info/rfc6823>.

Authors' Addresses

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Les Ginsberg
Cisco Systems

Email: ginsberg@cisco.com

Stefano Previdi
Cisco Systems

Email: sprevidi@cisco.com

TEAS Working Group Young Lee (Editor)
Internet Draft Huawei
Intended status: Informational

Daniele Ceccarelli
Expires September 1, 2018. 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

Table of Contents

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, confidentially 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, confidentially 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

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

H. Chen, Ed.
Huawei Technologies
R. Torvi, Ed.
Juniper Networks

October 19, 2015

Extensions to RSVP-TE for LSP Ingress Local Protection
draft-ietf-teas-rsvp-ingress-protection-04.txt

Abstract

This document describes extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for locally protecting the ingress node of a Traffic Engineered (TE) Label Switched Path (LSP), which is a Point-to-Point (P2P) LSP or a Point-to-Multipoint (P2MP) LSP.

Status of this Memo

Copyright Notice

Table of Contents

1. Co-authors

Ning So, Autumn Liu, Alia Atlas, Yimin Shen, Tarek Saad, Fengman Xu, Mehmet Toy, Lei Liu

2. Introduction

For a MPLS LSP it is important to have a fast-reroute method for protecting its ingress node and transit nodes. Protecting an ingress is not covered either in the fast-reroute method defined in [RFC4090] or in the P2MP fast-reroute extensions to fast-reroute in [RFC4875].

An alternate approach to local protection (fast-reroute) is to use global protection and set up a secondary backup LSP (whether P2MP or P2P) from a backup ingress to the egresses. The main disadvantage of this is that the backup LSP may reserve additional network bandwidth.

This specification defines a simple extension to RSVP-TE for local protection of the ingress node of a P2MP or P2P LSP.

2.1. An Example of Ingress Local Protection

Figure 1 shows an example of using a backup P2MP LSP to locally protect the ingress of a primary P2MP LSP, which is from ingress R1 to three egresses: L1, L2 and L3. The backup LSP is from backup ingress Ra to the next hops R2 and R4 of ingress R1.

```
                     [R2]******[R3]*****[L1]
                    *  |                               **** Primary LSP
                   *   |                               ---- Backup LSP
                  *    /                               .... BFD Session
                 *    /                                  $  Link
         ....[R1]*******[R4]****[R5]*****[L2]           $
         :  $  $    /     /        *                   $
         : $   $   /     /          *
        [S]    $  /     /            *
           $   $ /     /              *
            $  $/     /                *
             [Ra]----[Rb]               [L3]
```

Figure 1: Backup P2MP LSP for Locally Protecting Ingress

In normal operations, source S sends the traffic to primary ingress R1. R1 imports the traffic into the primary LSP.

When source S detects the failure of R1, it switches the traffic to backup ingress Ra, which imports the traffic from S into the backup LSP to R1's next hops R2 and R4, where the traffic is merged into the primary LSP, and then sent to egresses L1, L2 and L3. Source S detects the failure of R1 and switches the traffic within 10s of ms.

Note that the backup ingress is one logical hop away from the ingress. A logical hop is a direct link or a tunnel such as a GRE tunnel, over which RSVP-TE messages may be exchanged.

2.2. Ingress Local Protection with FRR

Through using the ingress local protection and the FRR, we can locally protect the ingress, all the links and the transit nodes of an LSP. The traffic switchover time is within 10s of ms whenever the ingress, any of the links and the transit nodes of the LSP fails.

The ingress node of the LSP can be locally protected through using the ingress local protection. All the links and all the transit nodes of the LSP can be locally protected through using the FRR.

3. Ingress Failure Detection

Exactly how to detect the failure of the ingress is out of scope. However, it is necessary to discuss different modes for detecting the failure because they determine what is the required behavior for the source and backup ingress.

3.1. Source Detects Failure

Source Detects Failure or Source-Detect for short means that the source is responsible for fast detecting the failure of the primary ingress of an LSP. The backup ingress is ready to import the traffic from the source into the backup LSP after the backup LSP is up.

In normal operations, the source sends the traffic to the primary ingress. When the source detects the failure of the primary ingress, it switches the traffic to the backup ingress, which delivers the traffic to the next hops of the primary ingress through the backup LSP, where the traffic is merged into the primary LSP.

For a P2P LSP, after the primary ingress fails, the backup ingress MUST use a method to reliably detect the failure of the primary ingress before the PATH message for the LSP expires at the next hop of the primary ingress. After reliably detecting the failure, the backup ingress sends/refreshes the PATH message to the next hop through the backup LSP as needed.

After the primary ingress fails, it will not be reachable after routing convergence. Thus checking whether the primary ingress (address) is reachable is a possible method.

3.2. Backup and Source Detect Failure

Backup and Source Detect Failure or Backup-Source-Detect for short means that both the backup ingress and the source are concurrently responsible for fast detecting the failure of the primary ingress.

In normal operations, the source sends the traffic to the primary ingress. It switches the traffic to the backup ingress when it detects the failure of the primary ingress.

The backup ingress does not import any traffic from the source into the backup LSP in normal operations. When it detects the failure of the primary ingress, it imports the traffic from the source into the backup LSP to the next hops of the primary ingress, where the traffic is merged into the primary LSP.

The source-detect is preferred. It is simpler than the backup-source-detect, which needs both the source and the backup ingress detect the ingress failure quickly.

4. Backup Forwarding State

Before the primary ingress fails, the backup ingress is responsible for creating the necessary backup LSPs. These LSPs might be multiple bypass P2P LSPs that avoid the ingress. Alternately, the backup ingress could choose to use a single backup P2MP LSP as a bypass or detour to protect the primary ingress of a primary P2MP LSP.

The backup ingress may be off-path or on-path of an LSP. If a backup ingress is not any node of the LSP, we call it is off-path. If a backup ingress is a next-hop of the primary ingress of the LSP, we call it is on-path. If it is on-path, the primary forwarding state associated with the primary LSP SHOULD be clearly separated from the backup LSP(s) state.

4.1. Forwarding State for Backup LSP

A forwarding entry for a backup LSP is created on the backup ingress after the LSP is set up. Depending on the failure-detection mode (e.g., source-detect), it may be used to forward received traffic or simply be inactive (e.g., backup-source-detect) until required. In either case, when the primary ingress fails, this entry is used to import the traffic into the backup LSP to the next hops of the primary ingress, where the traffic is merged into the primary LSP.

The forwarding entry for a backup LSP is a local implementation issue. In one device, it may have an inactive flag. This inactive forwarding entry is not used to forward any traffic normally. When the primary ingress fails, it is changed to active, and thus the traffic from the source is imported into the backup LSP.

5. Protocol Extensions

A new object INGRESS_PROTECTION is defined for signaling ingress local protection. It is backward compatible.

5.1. INGRESS_PROTECTION Object

The INGRESS_PROTECTION object with the FAST_REROUTE object in a PATH message is used to control the backup for protecting the primary ingress of a primary LSP. The primary ingress MUST insert this object into the PATH message to be sent to the backup ingress for protecting the primary ingress. It has the following format:

```
  Class-Num = TBD      C-Type = 1 for INGRESS_PROTECTION_IPv4
                       C-Type = 2 for INGRESS_PROTECTION_IPv6
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Length (bytes)        |    Class-Num  |    C-Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Reserved (zero)        |      Flags    |    Options    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          (Subobjects)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   Flags
    0x01    Ingress local protection available
    0x02    Ingress local protection in use
    0x04    Bandwidth protection

   Options
    0x01    Revert to Ingress
    0x02    P2MP Backup
```

The flags are used to communicate status information from the backup ingress to the primary ingress.

- o Ingress local protection available: The backup ingress sets this flag after backup LSPs are up and ready for locally protecting the primary ingress. The backup ingress sends this to the primary ingress to indicate that the primary ingress is locally protected.

- o Ingress local protection in use: The backup ingress sets this flag when it detects a failure in the primary ingress. The backup ingress keeps it and does not send it to the primary ingress since the primary ingress is down.

- o Bandwidth protection: The backup ingress sets this flag if the backup LSPs guarantee to provide desired bandwidth for the protected LSP against the primary ingress failure.

The options are used by the primary ingress to specify the desired behavior to the backup ingress.

- o Revert to Ingress: The primary ingress sets this option indicating that the traffic for the primary LSP successfully re-signaled will be switched back to the primary ingress from the backup ingress when the primary ingress is restored.

- o P2MP Backup: This option is set to ask for the backup ingress to use P2MP backup LSP to protect the primary ingress. Note that one spare bit of the flags in the FAST-REROUTE object can be used to indicate whether P2MP or P2P backup LSP is desired for protecting an ingress and transit node.

The INGRESS_PROTECTION object may contain some sub objects below.

5.1.1. Subobject: Backup Ingress IPv4 Address

When the primary ingress of a protected LSP sends a PATH message with an INGRESS_PROTECTION object to the backup ingress, the object may have a Backup Ingress IPv4 Address sub object containing an IPv4 address belonging to the backup ingress. The Type of the sub object is TBD1 (the exact number to be assigned by IANA), and the body of the sub object is given below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Backup ingress IPv4 address (4 bytes)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Backup ingress IPv4 address: An IPv4 host address of backup ingress

5.1.2. Subobject: Backup Ingress IPv6 Address

When the primary ingress of a protected LSP sends a PATH message with an INGRESS_PROTECTION object to the backup ingress, the object may have a Backup Ingress IPv6 Address sub object containing an IPv6 address belonging to the backup ingress. The Type of the sub object is TBD2, the body of the sub object is given below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Backup ingress IPv6 address (16 bytes)            |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Backup ingress IPv6 address: An IPv6 host address of backup ingress

5.1.3. Subobject: Ingress IPv4 Address

The INGRESS_PROTECTION object may have an Ingress IPv4 Address sub object containing an IPv4 address belonging to the primary ingress. The Type of the sub object is TBD3. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Ingress IPv4 address (4 bytes)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ingress IPv4 address: An IPv4 host address of ingress

5.1.4. Subobject: Ingress IPv6 Address

The INGRESS_PROTECTION object may have an Ingress IPv6 Address sub object containing an IPv6 address belonging to the primary ingress. The Type of the sub object is TBD4. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Ingress IPv6 address (16 bytes)                 |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ingress IPv6 address: An IPv6 host address of ingress

5.1.5. Subobject: Traffic Descriptor

The INGRESS_PROTECTION object may have a Traffic Descriptor sub object describing the traffic to be mapped to the backup LSP on the backup ingress for locally protecting the primary ingress. The Type of the sub object is TBD5, TBD6, TBD7 or TBD8 for Interface, IPv4 Prefix, IPv6 Prefix or Application Identifier respectively. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Traffic Element 1                      |
~                                                               ~
|                        Traffic Element n                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Traffic Descriptor sub object may contain multiple Traffic Elements of same type as follows:

- o Interface Traffic (Type TBD5): Each of the Traffic Elements is a 32 bit index of an interface, from which the traffic is imported into the backup LSP.

- o IPv4 Prefix Traffic (Type TBD6): Each of the Traffic Elements is an IPv4 prefix, containing an 8-bit prefix length followed by an IPv4 address prefix, whose length, in bits, is specified by the prefix length, padded to a byte boundary.

- o IPv6 Prefix Traffic (Type TBD7): Each of the Traffic Elements is an IPv6 prefix, containing an 8-bit prefix length followed by an IPv6 address prefix, whose length, in bits, is specified by the prefix length, padded to a byte boundary.

- o Application Traffic (Type TBD8): Each of the Traffic Elements is a 32 bit identifier of an application, from which the traffic is imported into the backup LSP.

5.1.6. Subobject: Label-Routes

The INGRESS_PROTECTION object in a PATH message from the primary ingress to the backup ingress will have a Label-Routes sub object containing the labels and routes that the next hops of the ingress use. The Type of the sub object is TBD9. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                           Subobjects                          ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Subobjects in the Label-Routes are copied from those in the RECORD_ROUTE objects in the RESV messages that the primary ingress receives from its next hops for the primary LSP. They MUST contain the first hops of the LSP, each of which is paired with its label.

6. Behavior of Ingress Protection

There are four parts of ingress protection: 1) setting up the necessary backup LSP forwarding state; 2) identifying the failure and providing the fast repair (as discussed in Sections 3 and 4); 3) maintaining the RSVP-TE control plane state until a global repair is done; and 4) performing the global repair(see Section 6.3).

6.1. Ingress Behavior

The primary ingress MUST be configured with a couple of pieces of information for ingress protection.

- o Backup Ingress Address: The primary ingress MUST know an IP address for it to be included in the INGRESS_PROTECTION object.

- o Application Traffic Identifier: The primary ingress and backup ingress MUST both know what application traffic should be directed into the LSP. If a list of prefixes in the Traffic Descriptor sub-object will not suffice, then a commonly understood Application Traffic Identifier can be sent between the primary ingress and backup ingress. The exact meaning of the identifier should be configured similarly at both the primary ingress and backup ingress. The Application Traffic Identifier is understood within the unique context of the primary ingress and backup ingress.

With this additional information, the primary ingress can create and signal the necessary RSVP extensions to support ingress protection.

The primary ingress relays the information for ingress protection of an LSP to the backup ingress via PATH messages. Once the LSP is created, the ingress of the LSP sends the backup ingress a PATH message with an INGRESS_PROTECTION object with Label-Routes subobject, which is populated with the next-hops and labels. This provides sufficient information for the backup ingress to create the appropriate forwarding state and backup LSP(s).

The ingress also sends the backup ingress all the other PATH messages for the LSP with an empty INGRESS_PROTECTION object. Thus, the backup ingress has access to all the PATH messages needed for modification to refresh control-plane state after a failure.

To protect the ingress of an LSP, the ingress MUST do the following after the LSP is up.

1. Select a PATH message.

2. If the backup ingress is off-path, then send it a PATH message with the content from the selected PATH message and an INGRESS_PROTECTION object; else (the backup ingress is a next hop, i.e., on-path case) add an INGRESS_PROTECTION object into the existing PATH message to the backup ingress (i.e., the next hop). The object contains the Traffic-Descriptor sub-object, the Backup Ingress Address sub-object and the Label-Routes sub-object. The options is set to indicate whether a Backup P2MP LSP is desired. The Label-Routes sub-object contains the next-hops of the ingress and their labels.

3. For each of the other PATH messages, send the backup ingress a PATH message with the content copied from the message and an empty INGRESS_PROTECTION object, which is an object without any Traffic-Descriptor sub-object.

6.2. Backup Ingress Behavior

An LER determines that the ingress local protection is requested for an LSP if the INGRESS_PROTECTION object is included in the PATH message it receives for the LSP. The LER can further determine that it is the backup ingress if one of its addresses is in the Backup Ingress Address sub-object of the INGRESS_PROTECTION object. The LER as the backup ingress will assume full responsibility of the ingress after the primary ingress fails. In addition, the LER determines that it is off-path if it is not any node of the LSP.

6.2.1. Backup Ingress Behavior in Off-path Case

The backup ingress considers itself as a PLR and the primary ingress as its next hop and provides a local protection for the primary ingress. It behaves very similarly to a PLR providing fast-reroute where the primary ingress is considered as the failure-point to protect. Where not otherwise specified, the behavior given in [RFC4090] for a PLR applies.

The backup ingress MUST follow the control-options specified in the INGRESS_PROTECTION object and the flags and specifications in the FAST-REROUTE object. This applies to providing a P2MP backup if the "P2MP backup" is set, a one-to-one backup if "one-to-one desired" is set, facility backup if the "facility backup desired" is set, and backup paths that support the desired bandwidth, and administrative-colors that are requested.

If multiple non empty INGRESS_PROTECTION objects have been received via multiple PATH messages for the same LSP, then the most recent one MUST be the one used.

The backup ingress creates the appropriate forwarding state for the backup LSP tunnel(s) to the merge point(s).

When the backup ingress sends a RESV message to the primary ingress, it MUST add an INGRESS_PROTECTION object into the message. It MUST set or clear the flags in the object to report "Ingress local protection available", "Ingress local protection in use", and "bandwidth protection".

If the backup ingress doesn’t have a backup LSP tunnel to each of the merge points, it SHOULD clear "Ingress local protection available". [Editor Note: It is possible to indicate the number or which are unprotected via a sub-object if desired.]

When the primary ingress fails, the backup ingress redirects the traffic from a source into the backup P2P LSPs or the backup P2MP LSP transmitting the traffic to the next hops of the primary ingress, where the traffic is merged into the protected LSP.

In this case, the backup ingress MUST keep the PATH message with the INGRESS_PROTECTION object received from the primary ingress and the RESV message with the INGRESS_PROTECTION object to be sent to the primary ingress. The backup ingress MUST set the "local protection in use" flag in the RESV message, indicating that the backup ingress is actively redirecting the traffic into the backup P2P LSPs or the backup P2MP LSP for locally protecting the primary ingress failure.

Note that the RESV message with this piece of information will not be sent to the primary ingress because the primary ingress has failed.

If the backup ingress has not received any PATH message from the primary ingress for an extended period of time (e.g., a cleanup timeout interval) and a confirmed primary ingress failure did not occur, then the standard RSVP soft-state removal SHOULD occur. The backup ingress SHALL remove the state for the PATH message from the primary ingress, and tear down the one-to-one backup LSPs for protecting the primary ingress if one-to-one backup is used or unbind the facility backup LSPs if facility backup is used.

When the backup ingress receives a PATH message from the primary ingress for locally protecting the primary ingress of a protected LSP, it MUST check to see if any critical information has been changed. If the next hops of the primary ingress are changed, the backup ingress SHALL update its backup LSP(s) accordingly.

When the backup ingress receives a PATH message with an non empty INGRESS_PROTECTION object, it examines the object to learn what traffic associated with the LSP. It determines the next-hops to be merged to by examining the Label-Routes sub-object in the object.

The backup ingress MUST store the PATH message received from the primary ingress, but NOT forward it.

The backup ingress responds with a RESV to the PATH message received from the primary ingress. If the INGRESS_PROTECTION object is not "empty", the backup ingress SHALL send the RESV message with the state indicating protection is available after the backup LSP(s) are successfully established.

6.2.2. Backup Ingress Behavior in On-path Case

An LER as the backup ingress determines that it is on-path if one of its addresses is a next hop of the primary ingress. The LER on-path MUST send the corresponding PATH messages without any INGRESS_PROTECTION object to its next hops. It creates a number of backup P2P LSPs or a backup P2MP LSP from itself to the other next hops (i.e., the next hops other than the backup ingress) of the primary ingress. The other next hops are from the Label-Routes sub object.

It also creates a forwarding entry, which sends/multicasts the traffic from the source to the next hops of the backup ingress along the protected LSP when the primary ingress fails. The traffic is described by the Traffic-Descriptor.

After the forwarding entry is created, all the backup P2P LSPs or the backup P2MP LSP is up and associated with the protected LSP, the backup ingress MUST send the primary ingress the RESV message with the INGRESS_PROTECTION object containing the state of the local protection such as "local protection available" flag set to one, which indicates that the primary ingress is locally protected.

When the primary ingress fails, the backup ingress sends/multicasts the traffic from the source to its next hops along the protected LSP and imports the traffic into each of the backup P2P LSPs or the backup P2MP LSP transmitting the traffic to the other next hops of the primary ingress, where the traffic is merged into protected LSP.

During the local repair, the backup ingress MUST continue to send the PATH messages to its next hops as before, keep the PATH message with the INGRESS_PROTECTION object received from the primary ingress and the RESV message with the INGRESS_PROTECTION object to be sent to the primary ingress. It MUST set the "local protection in use" flag in the RESV message.

6.2.3. Failure Detection and Refresh PATH Messages

As described in [RFC4090], it is necessary to refresh the PATH messages via the backup LSP(s). The Backup Ingress MUST wait to refresh the PATH messages until it can accurately detect that the ingress node has failed. An example of such an accurate detection would be that the IGP has no bi-directional links to the ingress node and the last change was long enough in the past that changes should have been received (i.e., an IGP network convergence time or approximately 2-3 seconds) or a BFD session to the primary ingress' loopback address has failed and stayed failed after the network has reconverged.

As described in [RFC4090 Section 6.4.3], the backup ingress, acting as PLR, MUST modify and send any saved PATH messages associated with the primary LSP to the corresponding next hops through backup LSP(s). Any PATH message sent will not contain any INGRESS_PROTECTION object. The RSVP_HOP object in the message contains an IP source address belonging to the backup ingress. The sender template object has the backup ingress address as its tunnel sender address.

6.3. Revertive Behavior

Upon a failure event in the (primary) ingress of a protected LSP, the protected LSP is locally repaired by the backup ingress. There are a couple of basic strategies for restoring the LSP to a full working path.

- Revert to Primary Ingress: When the primary ingress is restored, it re-signals each of the LSPs that start from the primary ingress. The traffic for every LSP successfully re-signaled is switched back to the primary ingress from the backup ingress.

- Global Repair by Backup Ingress: After determining that the primary ingress of an LSP has failed, the backup ingress computes a new optimal path, signals a new LSP along the new path, and switches the traffic to the new LSP.

6.3.1. Revert to Primary Ingress

If "Revert to Primary Ingress" is desired for a protected LSP, the (primary) ingress of the LSP SHOULD re-signal the LSP that starts from the primary ingress after the primary ingress restores. After the LSP is re-signaled successfully, the traffic SHOULD be switched back to the primary ingress from the backup ingress on the source node and redirected into the LSP starting from the primary ingress.

The primary ingress can specify the "Revert to Ingress" control-option in the INGRESS_PROTECTION object in the PATH messages to the backup ingress. After receiving the "Revert to Ingress" control-option, the backup ingress MUST stop sending/refreshing PATH messages for the protected LSP.

6.3.2. Global Repair by Backup Ingress

When the backup ingress has determined that the primary ingress of the protected LSP has failed (e.g., via the IGP), it can compute a new path and signal a new LSP along the new path so that it no longer relies upon local repair. To do this, the backup ingress MUST use the same tunnel sender address in the Sender Template Object and allocate a LSP ID different from the one of the old LSP as the LSP-ID of the new LSP. This allows the new LSP to share resources with the old LSP. In addition, if the Ingress recovers, the Backup Ingress SHOULD send it RESVs with the INGRESS_PROTECTION object where the "Revert to Ingress" is specified. The Ingress can learn from the RESVs what to signal. The Backup Ingress can reoptimize the new LSP as necessary until the Ingress recovers. Alternately, the Backup Ingress can create a new LSP with no bandwidth reservation that duplicates the path(s) of the protected LSP, move traffic to the new LSP, delete the protected LSP, and then resignal the new LSP with bandwidth.

7. Security Considerations

In principle this document does not introduce new security issues. The security considerations pertaining to RFC 4090, RFC 4875 and other RSVP protocols remain relevant.

8. IANA Considerations

IANA is requested to administer the assignment of new values defined in this document and summarized in this section.

8.1. A New Class Number

IANA maintains a registry called "Class Names, Class Numbers, and Class Types" under "Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters". IANA is requested to assign a new Class Number for new object INGRESS_PROTECTION as follows:

Class Names	Class Numbers	Class Types
INGRESS_PROTECTION	TBD (>192)	1: INGRESS_PROTECTION_IPv4
		2: INGRESS_PROTECTION_IPv6

IANA is requested to assign Types for new TLVs in the new objects as follows:

Type	Name	Allowed in
1	BACKUP_INGRESS_IPv4_ADDRESS	INGRESS_PROTECTION_IPv4
2	BACKUP_INGRESS_IPv6_ADDRESS	INGRESS_PROTECTION_IPv6
3	INGRESS_IPv4_ADDRESS	INGRESS_PROTECTION_IPv4
4	INGRESS_IPv6_ADDRESS	INGRESS_PROTECTION_IPv6
5	TRAFFIC_DESCRIPTOR_INTERFACE	INGRESS_PROTECTION
6	TRAFFIC_DESCRIPTOR_IPv4_PREFIX	INGRESS_PROTECTION_IPv4
7	TRAFFIC_DESCRIPTOR_IPv6_PREFIX	INGRESS_PROTECTION_IPv6
8	TRAFFIC_DESCRIPTOR_APPLICATION	INGRESS_PROTECTION
9	LabeL_Routes	INGRESS_PROTECTION

9. Contributors

Renwei Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA
Email: renwei.li@huawei.com

Quintin Zhao
Huawei Technologies
Boston, MA
USA
Email: quintin.zhao@huawei.com

Zhenbin Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA
Email: zhenbin.li@huawei.com

Boris Zhang
Telus Communications
200 Consilium Pl Floor 15
Toronto, ON M1H 3J3
Canada
Email: Boris.Zhang@telus.com

Markus Jork
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: mjork@juniper.net

10. Acknowledgement

The authors would like to thank Nobo Akiya, Rahul Aggarwal, Eric Osborne, Ross Callon, Loa Andersson, Daniel King, Michael Yue, Olufemi Komolafe, Rob Rennison, Neil Harrison, Kannan Sampath, Gregory Mirsky, and Ronhazli Adam for their valuable comments and suggestions on this draft.

11. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997,
<http://www.rfc-editor.org/info/rfc2119>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <http://www.rfc-editor.org/info/rfc4090>.

[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, <http://www.rfc-editor.org/info/rfc4875>.

Appendix A. Authors’ Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
USA
Email: huaimo.chen@huawei.com

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

Ning So
Tata Communications
2613 Fairbourne Cir.
Plano, TX 75082
USA
Email: ningso01@gmail.com

Autumn Liu
Ericsson
300 Holger Way
San Jose, CA 95134
USA
Email: autumn.liu@ericsson.com

Alia Atlas
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: akatlas@juniper.net

Yimin Shen
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: yshen@juniper.net

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Fengman Xu
Verizon
2400 N. Glenville Dr
Richardson, TX 75082
USA
Email: fengman.xu@verizon.com

Mehmet Toy
Comcast
1800 Bishops Gate Blvd.
Mount Laurel, NJ 08054
USA
Email: mehmet_toy@cable.comcast.com

Lei Liu
UC Davis
USA
Email: liulei.kddi@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2020

V. Beeram
T. Saad
Juniper Networks
R. Gandhi
Cisco Systems, Inc.
X. Liu
Jabil
I. Bryskin
Huawei Technologies
July 04, 2019

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-11

Abstract

This document defines a YANG data model for the configuration and management of RSVP Protocol. The model covers the building blocks of the RSVP protocol that can be augmented and used by other RSVP extension models such as RSVP extensions to Traffic-Engineering (RSVP-TE). The model covers the configuration, operational state, remote procedure calls, and event notifications data.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG data model that can be used to configure and manage the RSVP protocol [RFC2205]. This model covers RSVP protocol building blocks that can be augmented and used by other RSVP extension models- such as for signaling RSVP-TE MPLS (or other technology specific) Label Switched Paths (LSP)s.

1.1. Terminology

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. 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].

1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are prefixed using the standard prefix associated with the corresponding YANG imported modules, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-type	ietf-routing-types	XX
key-chain	ietf-key-chain	XX

Table 1: Prefixes and corresponding YANG modules

2. Model Overview

The RSVP base YANG module augments the "control-plane-protocol" list in ietf-routing [RFC8349] module with specific RSVP parameters in an "rsvp" container. It also defines an extension identity "rsvp" of base "rt:routing-protocol" to identify the RSVP protocol.

The augmentation of the RSVP model by other models (e.g. RSVP-TE for MPLS or other technologies) are outside the scope of this document and are discussed in separate document(s), e.g. [I-D.ietf-teas-yang-rsvp-te].

2.1. Module(s) Relationship

This document divides the RSVP model into two modules: base and extended RSVP modules. Some RSVP features are categorized as core to the function of the protocol and are supported by most vendors claiming the support for RSVP protocol. Such features configuration and state are grouped in the RSVP base module.

Other extended RSVP features are categorized as either optional or providing ability to better tune the basic functionality of the RSVP protocol. The support for extended RSVP features by all vendors is considered optional. Such features are grouped in a separate RSVP extended module.

The relationship between the base and extended RSVP YANG model and the IETF routing YANG model is shown in Figure 1.

```
                 +---------------+
       Routing   | ietf-routing |
                 +---------------+
                       o
                       |
                  +------------+
  RSVP module     | ietf-rsvp |
                  +------------+
                       o
                       |                    o: augment relationship
  RSVP extended        |
    module        +---------------------+
                  | ietf-rsvp-extended |
                  +---------------------+
```

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

2.2. Design Considerations

The RSVP base model does not aim to be feature complete. The primary intent is to cover a set of standard core features that are commonly in use. For example:

- o Authentication ([RFC2747])
- o Refresh Reduction ([RFC2961])
- o Hellos ([RFC3209])
- o Graceful Restart ([RFC3473], [RFC5063])

The extended RSVP YANG model covers the configuration for optional features that are not must for basic RSVP protocol operation.

The defined data model supports configuration inheritance for neighbors, and interfaces. Data elements defined in the main container (e.g. the container that encompasses the list of interfaces, or neighbors) are assumed to apply equally to all elements of the list, unless overridden explicitly for a certain element (e.g. interface). Vendors are expected to augment the above container(s) to provide the list of inheritance command for their implementations.

2.3. Model Notifications

Notifications data modeling is key in any defined 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:

- o Subscribe notifications on a per client basis
- o Specify subtree filters or xpath filters so that only interested contents will be sent.
- o Specify either periodic or on-demand notifications.

2.4. RSVP Base YANG Model

The RSVP base YANG data model defines the container "rsvp" as the top level container in this data model. The presence of this container enables the RSVP protocol functionality.

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 globals
         .
         .
      +--rw interfaces
            .
            +-- ro <<derived state associated with interfaces>>
         .
         .
      +--rw neighbors
            .
            +-- ro <<derived state associated with the tunnel>>
         .
         .
      +--rw sessions
            .
            +-- ro <<derived state associated with the tunnel>>
         .
   rpcs:
      +--x clear-session
      +--x clear-neighbor
```

Figure 2: RSVP high-level tree model view

Configuration and state data are grouped to those applicable on per node (global), per interface, per neighbor, or per session.

Global Data:

The global data cover the configuration and state that is applicable the RSVP protocol behavior.

Interface Data:

The interface data configuration and state model relevant attributes applicable to one or all RSVP interfaces. Any data or state at the "interfaces" container level is equally applicable to all interfaces - unless overridden by explicit configuration or state under a specific interface.

Neighbor Data:

The neighbor data cover configuration and state relevant to RSVP neighbors. Neighbors can be dynamically discovered using RSVP signaling or explicitly configured.

Session Data:

The sessions data branch covers configuration and state relevant to RSVP sessions. This is usually derived state that is result of signaling. This model defines attributes related to IP RSVP sessions as defined in [RFC2205].

2.4.1. Tree Diagram

Figure 3 shows the YANG tree representation for configuration and state data that is augmenting the RSVP basic module:

```
 module: ietf-rsvp
   augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol:
     +--rw rsvp!
        +--rw globals
        |  +--rw sessions
        |  |  +--ro session-ip*
        |  |          [destination protocol-id destination-port]
        |  |     +--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-state?      enumeration
        |  |     +--ro session-type?       identityref
        |  |     +--ro psbs
        |  |     |  +--ro psb* []
        |  |     |     +--ro source-port?   inet:port-number
        |  |     |     +--ro expires-in?    uint32
        |  |     +--ro rsbs
        |  |        +--ro rsb* []
        |  |           +--ro source-port?         inet:port-number
        |  |           +--ro reservation-style?   identityref
        |  |           +--ro expires-in?          uint32
        |  +--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 summary-refresh-sent?           yang:counter64
        |  |  |  +--ro summary-refresh-received?       yang:counter64
        |  |  |  +--ro unknown-messages-received?      yang:counter64
        |  |  +--ro packets
        |  |  |  +--ro sent?       yang:counter64
        |  |  |  +--ro received?   yang:counter64
        |  |  +--ro errors
        |  |     +--ro authenticate?    yang:counter64
        |  |     +--ro checksum?        yang:counter64
        |  |     +--ro packet-length?   yang:counter64
        |  +--rw graceful-restart
        |     +--rw enabled?   boolean
        +--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 summary-refresh-sent?           yang:counter64
     |  |  |  +--ro summary-refresh-received?       yang:counter64
     |  |  |  +--ro unknown-messages-received?      yang:counter64
     |  |  +--ro packets
     |  |  |  +--ro sent?       yang:counter64
     |  |  |  +--ro received?   yang:counter64
     |  |  +--ro errors
     |  |     +--ro authenticate?    yang:counter64
     |  |     +--ro checksum?        yang:counter64
     |  |     +--ro packet-length?   yang:counter64
     |  +--rw interface* [interface]
     |     +--rw interface            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 summary-refresh-sent?           yang:counter64
     |        |  +--ro summary-refresh-received?       yang:counter64
     |        |  +--ro unknown-messages-received?      yang:counter64
     |        +--ro packets
     |        |  +--ro sent?       yang:counter64
     |        |  +--ro received?   yang:counter64
     |        +--ro errors
     |           +--ro authenticate?    yang:counter64
     |           +--ro checksum?        yang:counter64
     |           +--ro packet-length?   yang:counter64
     +--rw neighbors
        +--rw neighbor* [address]
           +--rw address                      inet:ip-address
           +--rw epoch?                       uint32
           +--rw expiry-time?                 uint32
           +--rw graceful-restart
           |  +--rw enabled?                  boolean
           |  +--rw local-restart-time?       uint32
           |  +--rw local-recovery-time?      uint32
           |  +--rw neighbor-restart-time?    uint32
           |  +--rw neighbor-recovery-time?   uint32
           |  +--rw helper-mode
           |     +--rw enabled?                             boolean
           |     +--rw max-helper-restart-time?             uint32
           |     +--rw max-helper-recovery-time?            uint32
           |     +--rw neighbor-restart-time-remaining?     uint32
           |     +--rw neighbor-recovery-time-remaining?    uint32
           +--rw hello-status?                enumeration
           +--rw interface?                   if:interface-ref
           +--rw neighbor-state?              enumeration
           +--rw refresh-reduction-capable?   boolean
```

```
              +--rw restart-count?              yang:counter32
              +--rw restart-time?               yang:date-and-time

   rpcs:
     +---x clear-session
     |  +---w input
     |     +---w routing-protocol-instance-name    leafref
     |     +---w (filter-type)
     |        +--:(match-all)
     |        |  +---w all                         empty
     |        +--:(match-one)
     |           +---w session-info
     |              +---w (session-type)
     |                 +--:(rsvp-session-ip)
     |                    +---w destination         leafref
     |                    +---w protocol-id         uint8
     |                    +---w destination-port    inet:ip-address
     +---x clear-neighbor
        +---w input
           +---w routing-protocol-instance-name    leafref
           +---w (filter-type)
              +--:(match-all)
              |  +---w all                         empty
              +--:(match-one)
                 +---w neighbor-address            leafref
```

Figure 3: RSVP model tree diagram

2.4.2. YANG Module

The ietf-rsvp module imports from the following modules:

- o ietf-interfaces defined in [RFC8343]
- o ietf-yang-types and ietf-inet-types defined in [RFC6991]
- o ietf-routing defined in [RFC8349]
- o ietf-key-chain defined in [RFC8177]

```
  <CODE BEGINS> file "ietf-rsvp@2019-07-04.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";
    }

    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:Igor.Bryskin@huawei.com>

       Editor:   Himanshu Shah
                 <mailto:hshah@ciena.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 "2019-07-04" {
       description
         "A YANG Data Model for Resource Reservation Protocol";
       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-config {
      description
        "Base configuration parameters relating to RSVP
         Graceful-Restart";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Graceful Restart is enabled.
           'false' if RSVP Graceful Restart is disabled.";
      }
    }

    grouping graceful-restart {
      description
        "RSVP graceful restart parameters grouping";
      container graceful-restart {
        description
          "RSVP graceful restart parameters container";
        uses graceful-restart-config;
      }
    }

    grouping refresh-reduction-config {
      description
```

```
        "Configuration parameters relating to RSVP
         refresh reduction";

      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Refresh Reduction is enabled.
           'false' if RSVP Refresh Reduction is disabled.";
      }
    }

    grouping refresh-reduction {
      description
        "Top level grouping for RSVP refresh reduction
         parameters";
      container refresh-reduction {
        description
          "Top level container for RSVP refresh reduction
           parameters";
        uses refresh-reduction-config;
      }
    }

    grouping authentication-config {
      description
        "Configuration parameters relating to RSVP
         authentication";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Authentication is enabled.
           'false' if RSVP Authentication is disabled.";
      }
      leaf authentication-key {
        type string;
        description
          "An authentication key string";
        reference
          "RFC 2747: RSVP Cryptographic Authentication";
      }
      leaf crypto-algorithm {
       type identityref {
         base key-chain:crypto-algorithm;
       }
       mandatory true;
       description
         "Cryptographic algorithm associated with key.";
      }
```

```
    }

    grouping authentication {
      description
        "Top level grouping for RSVP authentication parameters";
      container authentication {
        description
          "Top level container for RSVP authentication
           parameters";
        uses authentication-config;
      }
    }

    grouping hellos-config {
      description
        "Configuration parameters relating to RSVP
        hellos";
      leaf enabled {
        type boolean;
        description
          "'true' if RSVP Hello is enabled.
           'false' if RSVP Hello is disabled.";
      }
    }

    grouping hellos {
      description
        "Top level grouping for RSVP hellos parameters";
      container hellos {
        description
          "Top level container for RSVP hello parameters";
        uses hellos-config;
      }
    }

    grouping signaling-parameters-config {
      description
        "Configuration parameters relating to RSVP
        signaling";
    }

    grouping signaling-parameters {
      description
        "Top level grouping for RSVP signaling  parameters";
      uses signaling-parameters-config;
    }

    grouping session-attributes-state {
```

```
      description
        "Top level grouping for RSVP session properties";
      leaf destination-port {
        type inet:port-number;
        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;
        description
          "The signaled name of this RSVP session.";
      }
      leaf session-state {
        type enumeration {
          enum "up" {
            description
              "RSVP session is up";
          }
          enum "down" {
            description
              "RSVP session is down";
          }
        }
        description
          "Enumeration of RSVP session states";
      }
      leaf session-type {
        type identityref {
          base rsvp-session-type;
        }
        description "RSVP session type";
      }
      container psbs {
```

```
        description "Path State Block container";
        list psb {
          description "List of path state blocks";
          leaf source-port {
            type inet:port-number;
            description "RSVP source port";
            reference "RFC2205";
          }
          leaf expires-in {
            type uint32;
            units seconds;
            description "Time to reservation expiry (in seconds)";
          }
        }
      }
      container rsbs {
        description "Reservation State Block container";
        list rsb {
          description "List of reservation state blocks";
          leaf source-port {
            type inet:port-number;
            description "RSVP source port";
            reference "RFC2205";
          }
          leaf reservation-style {
            type identityref {
              base reservation-style;
            }
            description "RSVP reservation style";
          }
          leaf expires-in {
            type uint32;
            units seconds;
            description "Time to reservation expiry (in seconds)";
          }
        }
      }
    }

    grouping neighbor-attributes {
      description
        "Top level grouping for RSVP neighbor properties";
      leaf address {
        type inet:ip-address;
        description
          "Address of RSVP neighbor";
      }
```

```
      leaf epoch {
        type uint32;
        description
          "Neighbor epoch.";
      }

      leaf expiry-time {
        type uint32;
        units seconds;
        description
          "Neighbor expiry time after which the neighbor state
          is purged if no states associated with it";
      }

      container graceful-restart {
        description
          "Graceful restart information.";

        leaf enabled {
          type boolean;
          description
            "'true' if graceful restart is enabled for the neighbor.";
        }

        leaf local-restart-time {
          type uint32;
          units seconds;
          description
            "Local node restart time";
        }

        leaf local-recovery-time {
          type uint32;
          units seconds;
          description
            "Local node recover time";
        }

        leaf neighbor-restart-time {
          type uint32;
          units seconds;
          description
            "Neighbor restart time";
        }

        leaf neighbor-recovery-time {
          type uint32;
          units seconds;
```

```
          description
            "Neighbor recover time";
        }

        container helper-mode {
          description
            "Helper mode information ";

          leaf enabled {
            type boolean;
            description
              "'true' if helper mode is enabled.";
          }

          leaf max-helper-restart-time {
            type uint32;
            units seconds;
            description
              "The time the router or switch waits after it
               discovers that a neighboring router has gone down
               before it declares the neighbor down";
          }

          leaf max-helper-recovery-time {
            type uint32;
            units seconds;
            description
              "The amount of time the router retains the state of its
               RSVP neighbors while they undergo a graceful restart";
          }

          leaf neighbor-restart-time-remaining {
            type uint32;
            units seconds;
            description
              "Number of seconds remaining for neighbor to send
               Hello message after restart.";
          }

          leaf neighbor-recovery-time-remaining {
            type uint32;
            units seconds;
            description
              "Number of seconds remaining for neighbor to
               refresh.";
          }
        } // helper-mode
      } // graceful-restart
```

```
      leaf hello-status {
        type enumeration {
          enum "enabled" {
            description
              "Enabled";
          }
          enum "disabled" {
            description
              "Disabled";
          }
          enum "restarting" {
            description
              "Restarting";
          }
        }
        description
          "Hello status";
      }

      leaf interface {
        type if:interface-ref;
        description
          "Interface where RSVP neighbor was detected";
      }

      leaf neighbor-state {
        type enumeration {
          enum "up" {
            description
              "up";
          }
          enum "down" {
            description
              "down";
          }
          enum "hello-disable" {
            description
              "hello-disable";
          }
          enum "restarting" {
            description
              "restarting";
          }
        }
        description
          "Neighbor state";
      }
```

```
      leaf refresh-reduction-capable {
        type boolean;
        description
          "enables all RSVP refresh reduction message
           bundling, RSVP message ID, reliable message delivery
           and summary refresh";
        reference
          "RFC 2961 RSVP Refresh Overhead Reduction
           Extensions";
      }

      leaf restart-count {
        type yang:counter32;
        description
          "Number of times this neighbor restart";
      }

      leaf restart-time {
        type yang:date-and-time;
        description
          "Last restart time of the neighbor";
      }
    }

    grouping packets-state {
      description
        "Packet statistics grouping";
      container packets {
        description
          "Packet statistics container";
        leaf sent {
          type yang:counter64;
          description
            "Packet sent count";
        }

        leaf received {
          type yang:counter64;
          description
            "Packet received count";
        }
      }
    }

    grouping protocol-state {
      description
        "RSVP protocol statistics grouping";
      container messages {
```

```
        description
          "RSVP protocol statistics container";
        leaf ack-sent {
          type yang:counter64;
          description
            "Hello sent count";
        }

        leaf ack-received {
          type yang:counter64;
          description
            "Hello received count";
        }

        leaf bundle-sent {
          type yang:counter64;
          description
            "Bundle sent count";
        }

        leaf bundle-received {
          type yang:counter64;
          description
            "Bundle received count";
        }

        leaf hello-sent {
          type yang:counter64;
          description
            "Hello sent count";
        }

        leaf hello-received {
          type yang:counter64;
          description
            "Hello received count";
        }

        leaf integrity-challenge-sent {
          type yang:counter64;
          description
            "Integrity Challenge sent count";
        }

        leaf integrity-challenge-received {
          type yang:counter64;
          description
            "Integrity Challenge received count";
```

```
        }

        leaf integrity-response-sent {
          type yang:counter64;
          description
            "Integrity Response sent count";
        }

        leaf integrity-response-received {
          type yang:counter64;
          description
            "Integrity Response received count";
        }

        leaf notify-sent {
          type yang:counter64;
          description
            "Notify sent count";
        }

        leaf notify-received {
          type yang:counter64;
          description
            "Notify received count";
        }

        leaf path-sent {
          type yang:counter64;
          description
            "Path sent count";
        }

        leaf path-received {
          type yang:counter64;
          description
            "Path received count";
        }

        leaf path-err-sent {
          type yang:counter64;
          description
            "Path error sent count";
        }

        leaf path-err-received {
          type yang:counter64;
          description
            "Path error received count";
```

```
        }

        leaf path-tear-sent {
          type yang:counter64;
          description
            "Path tear sent count";
        }

        leaf path-tear-received {
          type yang:counter64;
          description
            "Path tear received count";
        }

        leaf resv-sent {
          type yang:counter64;
          description
            "Resv sent count";
        }

        leaf resv-received {
          type yang:counter64;
          description
            "Resv received count";
        }

        leaf resv-confirm-sent {
          type yang:counter64;
          description
            "Confirm sent count";
        }

        leaf resv-confirm-received {
          type yang:counter64;
          description
            "Confirm received count";
        }

        leaf resv-err-sent {
          type yang:counter64;
          description
            "Resv error sent count";
        }

        leaf resv-err-received {
          type yang:counter64;
          description
            "Resv error received count";
```

```
        }

        leaf resv-tear-sent {
          type yang:counter64;
          description
            "Resv tear sent count";
        }

        leaf resv-tear-received {
          type yang:counter64;
          description
            "Resv tear received count";
        }

        leaf summary-refresh-sent {
          type yang:counter64;
          description
            "Summary refresh sent count";
        }

        leaf summary-refresh-received {
          type yang:counter64;
          description
            "Summary refresh received count";
        }

        leaf unknown-messages-received {
          type yang:counter64;
          description
            "Unknown packet received count";
        }
      }
    }

    grouping errors-state {
      description
        "Error statistics state grouping";
      container errors {
        description
          "Error statistics state container";
        leaf authenticate {
          type yang:counter64;
          description
            "The total number of packets received with an
            authentication failure.";
        }

        leaf checksum {
```

```
          type yang:counter64;
          description
            "The total number of packets received with an invalid
            checksum value.";
        }

        leaf packet-length {
          type yang:counter64;
          description
            "The total number of packets received with an invalid
            packet length.";
        }
      }
    }

    grouping statistics-state {
      description "RSVP statistic attributes.";
      container statistics {
        config false;
        description
          "statistics state container";
        uses protocol-state;
        uses packets-state;
        uses errors-state;
      }
    }

    grouping neighbor-derived-state {
      description
        "Derived state at neighbor level.";

    }

    grouping global-attributes {
      description
        "Top level grouping for RSVP global properties";
      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-state;
        }
      }
```

```
      uses statistics-state;
    }

    grouping intf-attributes {
      description
        "Top level grouping for RSVP interface properties";
      uses signaling-parameters;
      uses refresh-reduction;
      uses hellos;
      uses authentication;
      uses statistics-state;
    }

    augment "/rt:routing/rt:control-plane-protocols/"
      + "rt:control-plane-protocol" {
      when "rt:type = 'rsvp:rsvp'" {
      description
       "This augment is only valid when routing protocol
        instance type is RSVP.";
      }
      description
        "RSVP protocol augmentation";
      container rsvp {
        presence "Enable RSVP feature";
        description "RSVP feature container";
        container globals {
          description "RSVP global properties.";
          uses global-attributes;
          uses graceful-restart;
        }

        container interfaces {
          description
            "RSVP interfaces container";
          uses intf-attributes;

          list interface {
            key "interface";
            description
              "RSVP interfaces.";
            leaf interface {
              type if:interface-ref;
              description
                "RSVP interface.";
            }
            uses intf-attributes;
          }
        }
```

```
        container neighbors {
          description "RSVP neighbors container";
          list neighbor {
            key "address";
            description "List of RSVP neighbors";
            uses neighbor-attributes;
          }
        }
      }
    }

    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:globals" +
               "/rsvp:sessions/rsvp:session-ip/destination";
        }
        mandatory true;
        description "RSVP session";
      }
      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 {
      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 operation on";
              choice session-type {
                mandatory true;
                description "RSVP session type";
                case rsvp-session-ip {
                  uses session-ref;
                }
              }
            }
          }
        }
      }
    }

    rpc clear-neighbor {
      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 "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 specific RSVP neighbor session";
            }
          }
        }
      }
    }
  }
  <CODE ENDS>
```

2.5. RSVP Extended YANG Model

The RSVP extended YANG model covers non-core RSVP feature(s). It also covers feature(s) that are not necessarily supported by all vendors, and hence, can be guarded with "if-feature" checks.

2.5.1. Tree Diagram

Figure 4 shows the YANG tree representation for configuration and state data that is augmenting the RSVP extended module:

```
module: ietf-rsvp-extended
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
            /rsvp:graceful-restart:
    +--rw restart-time?    uint32
    +--rw recovery-time?   uint32
  augment /rt:routing/rt:control-plane-protocols
            /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
```

```
             /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:globals
             /rsvp:statistics/rsvp:messages:
     augment /rt:routing/rt:control-plane-protocols
             /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
             /rsvp:statistics/rsvp:errors:
     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?                boolean
       +--rw patherr-state-removal?   empty
     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 reliable-ack-hold-time?     uint32
       +--rw reliable-ack-max-size?      uint32
       +--rw reliable-retransmit-time?   uint32
       +--rw reliable-srefresh?          empty
       +--rw summary-max-size?           uint32
     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?                boolean
       +--rw patherr-state-removal?   empty
```

```
  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 reliable-ack-hold-time?     uint32
    +--rw reliable-ack-max-size?      uint32
    +--rw reliable-retransmit-time?   uint32
    +--rw reliable-srefresh?          empty
    +--rw summary-max-size?           uint32
  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 model tree diagram

2.5.2. YANG Module

The ietf-rsvp-extended module imports from the following modules:

- o ietf-rsvp defined in this document
- o ietf-routing defined in [RFC8349]
- o ietf-yang-types and ietf-inet-types defined in [RFC6991]
- o ietf-key-chain defined in [RFC8177]

Figure 5 shows the RSVP extended YANG module:

```
<CODE BEGINS> file "ietf-rsvp-extended@2019-07-04.yang"
module ietf-rsvp-extended {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";

  prefix "rsvp-ext";
```

```
     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:   Himanshu Shah
                  <mailto:hshah@ciena.com>

        Editor:   Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>

        Editor:   Xia Chen
```

```
                <mailto:jescia.chenxia@huawei.com>

      Editor:   Raqib Jones
                <mailto:raqib@Brocade.com>

      Editor:   Bin Wen
                <mailto:Bin_Wen@cable.comcast.com>";

  description
    "This module contains the Extended RSVP YANG data model.
     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 "2019-07-04" {
    description
      "A YANG Data Model for Extended Resource Reservation
       Protocol";
    reference
      "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
       (RSVP)";
  }

  /* RSVP features */
  feature authentication {
    description
      "Indicates support for RSVP authentication";
  }

  feature error-statistics {
    description
```

```
        "Indicates support for error statistics";
    }

    feature global-statistics {
      description
        "Indicates support for global statistics";
    }

    feature graceful-restart {
      description
        "Indicates support for RSVP graceful restart";
    }

    feature hellos {
      description
        "Indicates support for RSVP hellos (RFC3209).";
    }

    feature notify {
      description
        "Indicates support for RSVP notify message (RFC3473).";
    }

    feature refresh-reduction {
      description
        "Indicates support for RSVP refresh reduction (RFC2961).";
    }

    feature refresh-reduction-extended {
      description
        "Indicates support for RSVP refresh reduction (RFC2961).";
    }

    feature per-interface-statistics {
      description
        "Indicates support for per interface statistics";
    }

    grouping graceful-restart-extended-config {
      description
        "Configuration parameters relating to RSVP
         Graceful-Restart";
      leaf restart-time {
        type uint32;
        units seconds;
        description
          "Graceful restart time (seconds).";
        reference
```

```
           "RFC 5495: Description of the Resource
            Reservation Protocol - Traffic-Engineered
            (RSVP-TE) Graceful Restart Procedures";
       }
       leaf recovery-time {
         type uint32;
         units seconds;
         description
           "RSVP state recovery time";
       }
     }

     grouping authentication-extended-config {
       description
         "Configuration parameters relating to RSVP
         authentication";
       leaf lifetime {
         type uint32 {
           range "30..86400";
         }
         units seconds;
         description
           "Life time for each security association";
         reference
           "RFC 2747: RSVP Cryptographic
           Authentication";
       }
       leaf window-size {
         type uint32 {
           range "1..64";
         }
         description
           "Window-size to limit number of out-of-order
            messages.";
         reference
           "RFC 2747: RSVP Cryptographic
            Authentication";
       }
       leaf challenge {
         type empty;
         description
           "Enable challenge messages.";
         reference
           "RFC 2747: RSVP Cryptographic
            Authentication";
       }
       leaf retransmits {
         type uint32 {
```

```
           range "1..10000";
         }
         description
           "Number of retransmits when messages are
            dropped.";
         reference
           "RFC 2747: RSVP Cryptographic
            Authentication";
       }
       leaf key-chain {
         type key-chain:key-chain-ref;
         description
           "Key chain name to authenticate RSVP
            signaling messages.";
         reference
           "RFC 2747: RSVP Cryptographic
            Authentication";
       }
     }

     grouping hellos-extended-config {
       description
         "Configuration parameters relating to RSVP
          hellos";
       leaf interface-based {
         type empty;
         description
           "Enable interface-based Hello adjacency if present.";
       }
       leaf hello-interval {
         type uint32;
         units milliseconds;
         description
           "Configure interval between successive Hello
            messages in milliseconds.";
         reference
           "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
            RFC 5495: Description of the Resource
            Reservation Protocol - Traffic-Engineered
            (RSVP-TE) Graceful Restart Procedures";
       }
       leaf hello-misses {
         type uint32 {
           range "1..10";
         }
         description
           "Configure max number of consecutive missed
            Hello messages.";
```

```
        reference
          "RFC 3209: RSVP-TE: Extensions to RSVP for
           LSP Tunnels RFC 5495: Description of the
           Resource Reservation Protocol - Traffic-
           Engineered (RSVP-TE) Graceful Restart
           Procedures";
      }
    }

    grouping signaling-parameters-extended-config {
      description
        "Configuration parameters relating to RSVP
         signaling";
      leaf refresh-interval {
        type uint32;
        description
          "Set interval between successive refreshes";
      }
      leaf refresh-misses {
        type uint32;
        description
          "Set max number of consecutive missed
          messages for state expiry";
      }
      leaf checksum {
        type boolean;
        description
          "Enable RSVP message checksum computation";
      }
      leaf patherr-state-removal {
        type empty;
        description
          "State-Removal flag in Path Error message
          if present.";
      }
    }

    grouping refresh-reduction-extended-config {
      description
        "Configuration parameters relating to RSVP
        refresh reduction";

      leaf bundle-message-max-size {
        type uint32 {
          range "512..65000";
        }
        description
          "Configure maximum size (bytes) of a
```

```
            single RSVP Bundle message.";
        }
        leaf reliable-ack-hold-time {
          type uint32;
          units milliseconds;
          description
            "Configure hold time in milliseconds for
            sending RSVP ACK message(s).";
        }
        leaf reliable-ack-max-size {
          type uint32;
          description
            "Configure max size of a single RSVP ACK
            message.";
        }
        leaf reliable-retransmit-time {
          type uint32;
          units milliseconds;
          description
            "Configure min delay in milliseconds to
            wait for an ACK before a retransmit.";
        }
        leaf reliable-srefresh {
          type empty;
          description
            "Configure use of reliable messaging for
            summary refresh if present.";
        }
        leaf summary-max-size {
          type uint32 {
            range "20..65000";
          }
          description
            "Configure max size (bytes) of a single
            RSVP summary refresh message.";
        }
      }

      grouping packets-extended-state {
        description
          "Packet statistics.";
        leaf discontinuity-time {
          type yang:date-and-time;
          description
            "The time on the most recent occasion at which any one
            or more of the statistic counters suffered a
            discontinuity. If no such discontinuities have occurred
            since the last re-initialization of the local
```

```
            management subsystem, then this node contains the time
            the local management subsystem re-initialized itself.";
        }
        leaf out-dropped {
          type yang:counter64;
          description
            "Out packet drop count";
        }

        leaf in-dropped {
          type yang:counter64;
          description
            "In packet drop count";
        }

        leaf out-errors {
          type yang:counter64;
          description
            "Out packet errors count";
        }

        leaf in-errors {
          type yang:counter64;
          description
            "In packet rx errors count";
        }
      }

      grouping protocol-extended-state {
        description "RSVP protocol statistics.";
      }

      grouping errors-extended-state {
        description
          "Error statistics.";
      }

      grouping extended-state {
        description "RSVP statistic attributes.";
        uses packets-extended-state;
        uses protocol-extended-state;
        uses errors-extended-state;
      }

      /**
       * RSVP extensions augmentations
       */
```

```
     /* RSVP globals graceful restart*/
     augment "/rt:routing/rt:control-plane-protocols/" +
             "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
             "rsvp:graceful-restart" {
       description
         "RSVP globals configuration extensions";
       uses graceful-restart-extended-config;
     }

     /* RSVP statistics augmentation */
     augment "/rt:routing/rt:control-plane-protocols/" +
             "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
             "rsvp:statistics/rsvp:packets" {
       description
         "RSVP packet stats extensions";
       uses packets-extended-state;
     }
     augment "/rt:routing/rt:control-plane-protocols/" +
             "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
             "rsvp:statistics/rsvp:messages" {
       description
         "RSVP protocol message stats extensions";
       uses protocol-extended-state;
     }
     augment "/rt:routing/rt:control-plane-protocols/" +
             "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
             "rsvp:statistics/rsvp:errors" {
       description
         "RSVP errors stats extensions";
       uses errors-extended-state;
     }

     /**
      * RSVP all interfaces extensions
      */

     /* RSVP interface signaling extensions */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
       description
         "RSVP signaling all interfaces configuration extensions";
       uses signaling-parameters-extended-config;
     }

     /* RSVP refresh reduction extension */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
         + "rsvp:refresh-reduction" {
```

```
       description
         "RSVP refresh-reduction all interface configuration
          extensions";
       uses refresh-reduction-extended-config;
     }

     /* RSVP hellos extension */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
         + "rsvp:hellos" {
       description
         "RSVP hello all interfaces configuration extensions";
       uses hellos-extended-config;
     }

     /* RSVP authentication extension */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
         + "rsvp:authentication" {
       description
         "RSVP authentication all interfaces configuration extensions";
       uses authentication-extended-config;
     }

     /**
      * RSVP interface extensions
      */

     /* RSVP interface signaling extensions */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
         "rsvp:interface" {
       description
         "RSVP signaling interface configuration extensions";
       uses signaling-parameters-extended-config;
     }

     /* RSVP refresh reduction extension */
     augment "/rt:routing/rt:control-plane-protocols/"
         + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
         "rsvp:interface/rsvp:refresh-reduction" {
       description
         "RSVP refresh-reduction interface configuration extensions";
       uses refresh-reduction-extended-config;
     }

     /* RSVP hellos extension */
     augment "/rt:routing/rt:control-plane-protocols/"
```

```
        + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
        "rsvp:interface/rsvp:hellos" {
      description
        "RSVP hello interface configuration extensions";
      uses hellos-extended-config;
    }

    /* RSVP authentication extension */
    augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/" +
        "rsvp:interface/rsvp:authentication" {
      description
        "RSVP authentication interface configuration extensions";
      uses authentication-extended-config;
    }
  }
  <CODE ENDS>
```

Figure 5: RSVP extended 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
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
XML: N/A, the requested URI is an XML namespace.
```

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

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

name:       ietf-rsvp-extended
namespace:  urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
prefix:     ietf-rsvp-extendeed
reference:  RFCXXXX
```

4. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The 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 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:

The presence of this container enables the RSVP protocol functionality on a device. It alsocontrols the configuration settings on data nodes pertaining to RSVP sessions, interfaces and neighbors. 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.

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.

5. Acknowledgement

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

6. 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

7. Normative References

[I-D.ietf-netconf-subscribed-notifications]
Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Event Notifications", draft-ietf-netconf-subscribed-notifications-26 (work in progress), May 2019.

[I-D.ietf-netconf-yang-push]
Clemm, A. and E. Voit, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push-25 (work in progress), May 2019.

[I-D.ietf-teas-yang-rsvp-te]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for RSVP-TE Protocol", draft-ietf-teas-yang-rsvp-te-06 (work in progress), April 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>.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <https://www.rfc-editor.org/info/rfc5063>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <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>.

[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>.

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
Jabil

Email: Xufeng_Liu@jabil.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 5, 2020

T. Saad
Juniper Networks
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
I. Bryskin
Individual
November 02, 2019

A YANG Data Model for Traffic Engineering Tunnels and Interfaces
draft-ietf-teas-yang-te-22

Abstract

This document defines a YANG data model for the configuration and management of Traffic Engineering (TE) interfaces, tunnels and Label Switched Paths (LSPs). 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

Table of Contents

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 TE Tunnels, Label Switched Paths (LSPs) and TE interfaces and 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) will augment the TE generic YANG module.

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]
rt-types	ietf-routing-types	[RFC8294]
te	ietf-te	this document
te-dev	ietf-te-device	this document
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]

Table 1: Prefixes and corresponding YANG modules

1.3. TE Technology Models

This document describes the TE generic 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 TE generic 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 TE generic YANG data model does not cover signaling protocol data. This is expected to be covered by augmentations defined in other document(s).

1.4. State Data Organization

The Network Management Datastore Architecture (NMDA) [RFC8342] addresses modeling state data for ephemeral objects. This draft adopts the NMDA proposal for configuration and state data representation as per IETF guidelines for new IETF YANG models.

2. Model Overview

The data model(s) defined in this document cover core TE features that are commonly supported across different vendor implementations. The support of extended or vendor specific TE feature(s) is expected to be in augmentations to the base model defined in this document.

2.1. Module(s) Relationship

The TE generic YANG data model 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 TE generic 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 TE generic 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].

```
     TE generic     +---------+         o: augment
     module         | ietf-te |o-------------+
                    +---------+               \
                           o                   \
                           |\                   \
                           | \                   \
                           |  +----------------+  \
                           |  | ietf-te-device | TE device module
                           |  +----------------+    \
                           |       o        o        \
                           |     /           \        \
                           |   /              \        \
                    +--------------+           +---------------+
     RSVP-TE module | ietf-rsvp-te |o .        | ietf-te-mpls^ |
                    +--------------+   \       +---------------+
                       |                \
                       |                 \
                       |                  \
                       |                   \
                       |                    \
                       o                 +-------------------+
                    +-----------+        | ietf-rsvp-otn-te^ |
     RSVP module    | ietf-rsvp |        +-------------------+
                    +-----------+           RSVP-TE with OTN
                                            extensions

                           ^ shown for illustration
                             (not in this document)
```

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

```
     +---------+
     | ietf-te |       ^: import
     +---------+
       ^  ^  ^
       |  |  +--------------------------------------+
       |  +----------------------+                  |
       |                         |                  |
     +----------------+   +---------------------+ +--------------------+
     | ietf-te-types* |   | ietf-te-mpls-types* | | ietf-te-otn-types* |
     +----------------+   +---------------------+ +--------------------+

     *: not in this document, shown for illustration only
```

Figure 2: Relationship between generic and technology specific TE types modules

2.2. Design Considerations

The following design considerations are taken into account with respect data organization:

- o reusable TE data types that are data plane independent are grouped in the TE generic types module "ietf-te-types.yang" defined in [I-D.ietf-teas-yang-te-types]

- o reusable TE data types that are data plane specific are defined in a data plane type module, e.g. "ietf-te-packet-types.yang" as defined in [I-D.ietf-teas-yang-te-types]. Other data plane types are expected to be defined in separate module(s) as shown in Figure 2

- o The TE generic YANG data model "ietf-te" contains device independent data and can be used to model data off a device (e.g. on a controller). The device-specific TE data is defined in module "ietf-te-device" as shown in Figure 1.

- o 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.

- o This model declares a number of TE functions as features that can be optionally supported.

2.3. Model Tree Diagram

Figure 3 shows the tree diagram of the TE YANG model defined in modules: ietf-te.yang, and ietf-te-device.yang.

```
module: ietf-te
  +--rw te!
     +--rw globals
     |  +--rw named-admin-groups
     |  |  +--rw named-admin-group* [name]
     |  |     +--rw name            string
     |  |     +--rw bit-position?   uint32
     |  +--rw named-srlgs
     |  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
     |  |     +--rw name     string
     |  |     +--rw group?   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?
     |  |     |  |              |             rt-types:generalized-label
     |  |     |  |              +--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 shared-resources-tunnels
     |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |     +--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 te-dev:lsp-install-interval?        uint32
     |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  +--rw te-dev:lsp-invalidation-interval?   uint32
     +--rw tunnels
     |  +--rw tunnel* [name]
     |  |  +--ro operational-state?                   identityref
     |  |  +--rw name                                 string
     |  |  +--rw identifier?                          uint16
     |  |  +--rw description?                         string
     |  |  +--rw encoding?                            identityref
     |  |  +--rw switching-type?                      identityref
     |  |  +--rw provisioning-state?                  identityref
     |  |  +--rw preference?                          uint8
     |  |  +--rw reoptimize-timer?                    uint16
     |  |  +--rw source?
     |  |  |       te-types:te-node-id
     |  |  +--rw destination?
     |  |  |       te-types:te-node-id
     |  |  +--rw src-tp-id?                           yang:hex-string
     |  |  +--rw dst-tp-id?                           yang:hex-string
     |  |  +--rw bidirectional?                       boolean
     |  |  +--rw association-objects
     |  |  |  +--rw association-object* [type ID source global-source]
     |  |  |  |  +--rw type             identityref
     |  |  |  |  +--rw ID               uint16
     |  |  |  |  +--rw source           te-types:te-node-id
     |  |  |  |  +--rw global-source    te-types:te-node-id
     |  |  |  +--rw association-object-extended*
     |  |  |          [type ID source global-source extended-ID]
     |  |  |     +--rw type             identityref
     |  |  |     +--rw ID               uint16
     |  |  |     +--rw source           te-types:te-node-id
     |  |  |     +--rw global-source    te-types:te-node-id
     |  |  |     +--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 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
     |  |  +--rw p2p-primary-paths
     |  |  |  +--rw p2p-primary-path* [name]
     |  |  |     +--rw name                             string
     |  |  |     +--rw path-setup-protocol?             identityref
     |  |  |     +--rw path-computation-method?         identityref
     |  |  |     +--rw path-computation-server?
     |  |  |     |       inet:ip-address
     |  |  |     +--rw compute-only?                    empty
     |  |  |     +--rw use-path-computation?            boolean
     |  |  |     +--rw lockdown?                        empty
     |  |  |     +--ro path-scope?                      identityref
     |  |  |     +--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
     |  |  |     +--rw preference?                     uint8
     |  |  |     +--rw k-requested-paths?              uint8
     |  |  |     +--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-label
     |  |  |     |  |              +--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 shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     +--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-computed-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 shared-resources-tunnels
     |  |  |     |           +--ro lsp-shared-resources-tunnel*
     |  |  |     |                   tunnel-ref
     |  |  |     +--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* [lsp-id]
   |  |  |     |     +--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 source?
   |  |  |     |     |       te-types:te-node-id
   |  |  |     |     +--ro destination?
   |  |  |     |     |       te-types:te-node-id
   |  |  |     |     +--ro tunnel-id?
   |  |  |     |     |       uint16
   |  |  |     |     +--ro lsp-id
   |  |  |     |     |       uint16
   |  |  |     |     +--ro extended-tunnel-id?
   |  |  |     |     |       yang:dotted-quad
   |  |  |     |     +--ro operational-state?
   |  |  |     |     |       identityref
   |  |  |     |     +--ro path-setup-protocol?
   |  |  |     |     |       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-shared-resources-tunnel?
   |  |  |     |     |       tunnel-ref
   |  |  |     |     +--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)
     |  |  |     |     |           +--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 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-computed-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 shared-resources-tunnels
     |  |  |     |     |     +--ro lsp-shared-resources-tunnel*
     |  |  |     |     |             tunnel-ref
     |  |  |     |     +--ro te-dev:lsp-timers
     |  |  |     |     |  +--ro te-dev:life-time?         uint32
     |  |  |     |     |  +--ro te-dev:time-to-install?   uint32
     |  |  |     |     |  +--ro te-dev:time-to-destroy?   uint32
```

```
     |  |  |     |     +--ro te-dev:downstream-info
     |  |  |     |     |  +--ro te-dev:nhop?
     |  |  |     |     |  |       inet:ip-address
     |  |  |     |     |  +--ro te-dev:outgoing-interface?
     |  |  |     |     |  |       if:interface-ref
     |  |  |     |     |  +--ro te-dev:neighbor?
     |  |  |     |     |  |       inet:ip-address
     |  |  |     |     |  +--ro te-dev:label?
     |  |  |     |     |          rt-types:generalized-label
     |  |  |     |     +--ro te-dev:upstream-info
     |  |  |     |        +--ro te-dev:phop?       inet:ip-address
     |  |  |     |        +--ro te-dev:neighbor?   inet:ip-address
     |  |  |     |        +--ro te-dev:label?
     |  |  |     |                rt-types:generalized-label
     |  |  |     +--rw p2p-primary-reverse-path
     |  |  |     |  +--rw name?                            string
     |  |  |     |  +--rw path-setup-protocol?             identityref
     |  |  |     |  +--rw path-computation-method?         identityref
     |  |  |     |  +--rw path-computation-server?
     |  |  |     |  |       inet:ip-address
     |  |  |     |  +--rw compute-only?                    empty
     |  |  |     |  +--rw use-path-computation?            boolean
     |  |  |     |  +--rw lockdown?                        empty
     |  |  |     |  +--ro path-scope?                      identityref
     |  |  |     |  +--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
     |  |  |     |  +--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-label
     |  |  |     |  |  |              +--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 shared-resources-tunnels
```

```
     |  |  |     |  |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     |  +--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-computed-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 shared-resources-tunnels
     |  |  |     |  |           +--ro lsp-shared-resources-tunnel*
     |  |  |     |  |                   tunnel-ref
     |  |  |     |  +--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* [lsp-id]
     |  |  |     |  |     +--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 source?
     |  |  |     |  |     |       te-types:te-node-id
     |  |  |     |  |     +--ro destination?
     |  |  |     |  |     |       te-types:te-node-id
     |  |  |     |  |     +--ro tunnel-id?
     |  |  |     |  |     |       uint16
     |  |  |     |  |     +--ro lsp-id
     |  |  |     |  |     |       uint16
     |  |  |     |  |     +--ro extended-tunnel-id?
     |  |  |     |  |     |       yang:dotted-quad
     |  |  |     |  |     +--ro operational-state?
     |  |  |     |  |     |       identityref
     |  |  |     |  |     +--ro path-setup-protocol?
     |  |  |     |  |     |       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-shared-resources-tunnel?
     |  |  |     |  |     |       tunnel-ref
     |  |  |     |  |     +--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)
     |  |  |     |  |     |           +--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 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-computed-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 shared-resources-tunnels
     |  |  |     |  |           +--ro lsp-shared-resources-tunnel*
     |  |  |     |  |                   tunnel-ref
```

```
     |  |  |     |  +--rw p2p-secondary-reverse-path
     |  |  |     |     +--rw secondary-path?        leafref
     |  |  |     |     +--rw path-setup-protocol?   identityref
     |  |  |     +--rw candidate-p2p-secondary-paths
     |  |  |        +--rw candidate-p2p-secondary-path*
     |  |  |                [secondary-path]
     |  |  |           +--rw secondary-path         leafref
     |  |  |           +--rw path-setup-protocol?   identityref
     |  |  |           +--ro active?                boolean
     |  |  +--rw p2p-secondary-paths
     |  |  |  +--rw p2p-secondary-path* [name]
     |  |  |     +--rw name                             string
     |  |  |     +--rw path-setup-protocol?             identityref
     |  |  |     +--rw path-computation-method?         identityref
     |  |  |     +--rw path-computation-server?
     |  |  |     |       inet:ip-address
     |  |  |     +--rw compute-only?                    empty
     |  |  |     +--rw use-path-computation?            boolean
     |  |  |     +--rw lockdown?                        empty
     |  |  |     +--ro path-scope?                      identityref
     |  |  |     +--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
   |  |  |     +--rw preference?                      uint8
   |  |  |     +--rw k-requested-paths?               uint8
   |  |  |     +--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-label
     |  |  |     |  |           +--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 shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   tunnel-ref
     |  |  |     +--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-computed-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 shared-resources-tunnels
     |  |  |     |           +--ro lsp-shared-resources-tunnel*
     |  |  |     |                   tunnel-ref
     |  |  |     +--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* [lsp-id]
     |  |  |           +--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 source?
     |  |  |           |       te-types:te-node-id
     |  |  |           +--ro destination?
     |  |  |           |       te-types:te-node-id
     |  |  |           +--ro tunnel-id?
     |  |  |           |       uint16
     |  |  |           +--ro lsp-id
     |  |  |           |       uint16
     |  |  |           +--ro extended-tunnel-id?
     |  |  |           |       yang:dotted-quad
     |  |  |           +--ro operational-state?
     |  |  |           |       identityref
     |  |  |           +--ro path-setup-protocol?
     |  |  |           |       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-shared-resources-tunnel?
     |  |  |           |       tunnel-ref
     |  |  |           +--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)
     |  |  |           |           +--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 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-computed-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 shared-resources-tunnels
     |  |  |           |     +--ro lsp-shared-resources-tunnel*
     |  |  |           |             tunnel-ref
     |  |  |           +--ro te-dev:lsp-timers
     |  |  |           |  +--ro te-dev:life-time?         uint32
     |  |  |           |  +--ro te-dev:time-to-install?   uint32
     |  |  |           |  +--ro te-dev:time-to-destroy?   uint32
     |  |  |           +--ro te-dev:downstream-info
     |  |  |           |  +--ro te-dev:nhop?
     |  |  |           |  |       inet:ip-address
     |  |  |           |  +--ro te-dev:outgoing-interface?
     |  |  |           |  |       if:interface-ref
     |  |  |           |  +--ro te-dev:neighbor?
     |  |  |           |  |       inet:ip-address
```

```
     |  |  |            |  +--ro te-dev:label?
     |  |  |            |          rt-types:generalized-label
     |  |  |            +--ro te-dev:upstream-info
     |  |  |               +--ro te-dev:phop?       inet:ip-address
     |  |  |               +--ro te-dev:neighbor?   inet:ip-address
     |  |  |               +--ro te-dev:label?
     |  |  |                       rt-types:generalized-label
     |  |  +---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
     |  |  +--rw te-dev:lsp-install-interval?        uint32
     |  |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  |  +--rw te-dev:lsp-invalidation-interval?   uint32
     |  +--rw tunnel-p2mp* [name]
     |     +--rw name                  string
     |     +--rw identifier?           uint16
     |     +--rw description?          string
     |     +--ro operational-state?   identityref
     +--ro lsps-state
     |  +--ro lsp*
     |          [source destination tunnel-id lsp-id extended-tunnel-id]
     |     +--ro source
     |     |       te-types:te-node-id
     |     +--ro destination
     |     |       te-types:te-node-id
     |     +--ro tunnel-id                            uint16
     |     +--ro lsp-id                               uint16
     |     +--ro extended-tunnel-id                   yang:dotted-quad
     |     +--ro operational-state?                   identityref
     |     +--ro path-setup-protocol?                 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)
     |       |           +--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 te-dev:lsp-timers
     |       |  +--ro te-dev:life-time?         uint32
     |       |  +--ro te-dev:time-to-install?   uint32
     |       |  +--ro te-dev:time-to-destroy?   uint32
     |       +--ro te-dev:downstream-info
     |       |  +--ro te-dev:nhop?                  inet:ip-address
     |       |  +--ro te-dev:outgoing-interface?   if:interface-ref
     |       |  +--ro te-dev:neighbor?              inet:ip-address
     |       |  +--ro te-dev:label?
     |       |          rt-types:generalized-label
     |       +--ro te-dev:upstream-info
     |          +--ro te-dev:phop?       inet:ip-address
     |          +--ro te-dev:neighbor?   inet:ip-address
     |          +--ro te-dev:label?      rt-types:generalized-label
     +--rw te-dev:interfaces
     |  +--rw te-dev:threshold-type?            enumeration
     |  +--rw te-dev:delta-percentage?          rt-types:percentage
     |  +--rw te-dev:threshold-specification?   enumeration
```

```
     |  +--rw te-dev:up-thresholds*             rt-types:percentage
     |  +--rw te-dev:down-thresholds*           rt-types:percentage
     |  +--rw te-dev:up-down-thresholds*        rt-types:percentage
     |  +--rw te-dev:interface* [interface]
     |     +--rw te-dev:interface
     |     |       if:interface-ref
     |     +--rw te-dev:te-metric?
     |     |       te-types:te-metric
     |     +--rw (te-dev:admin-group-type)?
     |     |  +--:(te-dev:value-admin-groups)
     |     |  |  +--rw (te-dev:value-admin-group-type)?
     |     |  |     +--:(te-dev:admin-groups)
     |     |  |     |  +--rw te-dev:admin-group?
     |     |  |     |          te-types:admin-group
     |     |  |     +--:(te-dev:extended-admin-groups)
     |     |  |              {te-types:extended-admin-groups}?
     |     |  |        +--rw te-dev:extended-admin-group?
     |     |  |                te-types:extended-admin-group
     |     |  +--:(te-dev:named-admin-groups)
     |     |     +--rw te-dev:named-admin-groups* [named-admin-group]
     |     |        +--rw te-dev:named-admin-group    leafref
     |     +--rw (te-dev:srlg-type)?
     |     |  +--:(te-dev:value-srlgs)
     |     |  |  +--rw te-dev:values* [value]
     |     |  |     +--rw te-dev:value    uint32
     |     |  +--:(te-dev:named-srlgs)
     |     |     +--rw te-dev:named-srlgs* [named-srlg]
     |     |             {te-types:named-srlg-groups}?
     |     |        +--rw te-dev:named-srlg    leafref
     |     +--rw te-dev:threshold-type?
     |     |       enumeration
     |     +--rw te-dev:delta-percentage?
     |     |       rt-types:percentage
     |     +--rw te-dev:threshold-specification?
     |     |       enumeration
     |     +--rw te-dev:up-thresholds*
     |     |       rt-types:percentage
     |     +--rw te-dev:down-thresholds*
     |     |       rt-types:percentage
     |     +--rw te-dev:up-down-thresholds*
     |     |       rt-types:percentage
     |     +--rw te-dev:switching-capabilities* [switching-capability]
     |     |  +--rw te-dev:switching-capability    identityref
     |     |  +--rw te-dev:encoding?               identityref
     |     +--ro te-dev:state
     |        +--ro te-dev:te-advertisements-state
     |           +--ro te-dev:flood-interval?          uint32
     |           +--ro te-dev:last-flooded-time?       uint32
```

```
     |           +--ro te-dev:next-flooded-time?        uint32
     |           +--ro te-dev:last-flooded-trigger?     enumeration
     |           +--ro te-dev:advertized-level-areas* [level-area]
     |              +--ro te-dev:level-area    uint32
     +--rw te-dev:performance-thresholds

  rpcs:
    +---x globals-rpc
    +---x interfaces-rpc
    +---x tunnels-rpc
       +---w input
       |  +---w tunnel-info
       |     +---w (type)?
       |        +--:(tunnel-p2p)
       |        |  +---w p2p-id?    tunnel-ref
       |        +--:(tunnel-p2mp)
       |           +---w p2mp-id?   tunnel-p2mp-ref
       +--ro output
          +--ro result
             +--ro result?   enumeration

  notifications:
    +---n globals-notif
    +---n tunnels-notif
module: ietf-te-device

  rpcs:
    +---x interfaces-rpc

  notifications:
    +---n interfaces-notif
```

Figure 3: TE generic model configuration and state tree

3. Model Organization

The TE generic YANG data module "ietf-te" covers configuration, state, RPC and notifications data pertaining to TE global, tunnels and LSPs parameters that are device independent.

The container "te" is the top level container in the data model. The presence of this container enables TE function system wide.

The model top level organization is shown below in Figure 4:

```
module: ietf-te
   +--rw te!
      +--rw globals
         .
         .

      +--rw tunnels
         .
         .

      +-- lsps-state

rpcs:
   +---x globals-rpc
   +---x tunnels-rpc
notifications:
   +---n globals-notif
   +---n tunnels-notif
```

Figure 4: TE generic highlevel model view

3.1. Global Configuration and State Data

The global TE branch of the data model covers configurations that control TE features behavior system-wide, and its respective state. Examples of such configuration data are:

- o Table of named SRLG mappings
- o Table of named (extended) administrative groups mappings
- o Table of named path-constraints sets
- o System-wide capabilities for LSP reoptimization
  - * Reoptimization timers (periodic interval, LSP installation and cleanup)
  - * Link state flooding thresholds
  - * Periodic flooding interval
- o Global capabilities that affect originating, transiting and terminating LSPs. For example:
  - * Path selection parameters (e.g. metric to optimize, etc.)
  - * Path or segment protection parameters

3.2. Interfaces Configuration and State Data

This branch of the model covers configuration and state data corresponding to TE interfaces that are present on a device. The module "ietf-te-device" is introduced to hold such TE device specific properties.

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)

```
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 5: TE interface state

The derived state associated with interfaces is grouped under the interface "state" sub-container as shown in Figure 5. 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

3.3. Tunnels Configuration and State Data

This branch covers data related to TE tunnels configuration and state. The derived state associated with tunnels is grouped under a state container as shown in Figure 6.

```
module: ietf-te
   +--rw te!
      +--rw tunnels
            <<intended configuration>>
         .
         +-- ro state
            <<derived state associated with the tunnel>>
```

Figure 6: TE interface state tree

Examples of tunnel configuration data for TE tunnels:

- Name and type (e.g. P2P, P2MP) of the TE tunnel
- Administrative and operational state of the TE tunnel
- Set of primary and corresponding secondary paths and corresponding path attributes
- Bidirectional path attribute(s) including forwarding and reverse path properties
- Protection and restoration path parameters

3.3.1. Tunnel Compute-Only Mode

A configured TE tunnel, by default, is provisioned so it can carry traffic as soon as a valid path is computed and an LSP instantiated. In some cases, however, a TE tunnel 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 tunnel is configured in "compute-only" mode to distinguish it from default tunnel behavior.

A "compute-only" TE tunnel is configured as a usual TE tunnel with associated per path constraint(s) and properties on a device or controller. The device or controller computes the feasible path(s) subject to configured constraints and reflects the computed path(s) in the LSP(s) Record-Route Object (RRO) list. At any time, a client may query "on-demand" the "compute-only" TE tunnel computed path(s) properties by querying the state of the tunnel. Alternatively, the client can subscribe on the "compute-only" TE tunnel to be notified of computed path(s) and whenever it changes.

3.3.2. Tunnel Hierarchical Link Endpoint

TE LSPs can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used to form links to carry traffic in 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.

3.4. TE LSPs State Data

TE LSPs are derived state data that are present whenever the LSP(s) are instantiated - for example, when associated signaling completes. TE LSPs exists on routers as ingress (starting point of LSP), transit (mid-point of LSP), or egress (termination point of the LSP). In the model, the nodes holding TE LSP data exist in the read-only lsps-state list as show in Figure 3.

3.5. Global RPC Data

This branch of the model covers system-wide RPC execution data to trigger actions and optionally expect responses. Examples of such TE commands are to:

- Clear global TE statistics of various features

3.6. Interface RPC Data

This collection of data in the model defines TE interface RPC execution commands. Examples of these are to:

- Clear TE statistics for all or for individual TE interfaces

- Trigger immediate flooding for one or all TE interfaces

3.7. Tunnel RPC Data

This branch of the model covers TE tunnel RPC execution data to trigger actions and expect responses. The TE generic YANG data model defines target containers that an external module in [I-D.ietf-teas-yang-path-computation] augments with RPCs that allow the invocation of certain TE functions (e.g. path computations).

4. TE Generic and Helper YANG Modules

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

- o ietf-yang-types and ietf-inet-types defined in [RFC6991]

- o ietf-te-types defined in [I-D.ietf-teas-yang-te-types]

This module references the following documents: [RFC6991], [RFC4875], [RFC7551], [RFC4206], [RFC4427], [RFC4872], [RFC3945], [RFC3209], [RFC4872], [RFC6780], and [RFC7308].

```
<CODE BEGINS> file "ietf-te@2019-11-02.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 "draft-ietf-teas-yang-te-types: A YANG Data Model for
               Common Traffic Engineering Types";
  }

  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, RPC and
     notifications. 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 "2019-11-02" {
    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 error no topology error reason";
  }
  identity path-computation-error-no-server {
    base path-computation-error-reason;
    description
      "Path computation error no server error reason";
  }
  identity path-computation-error-path-not-found {
    base path-computation-error-reason;
    description
      "Path computation no path found error reason";
  }

  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 tunnel-p2mp-ref {
    type leafref {
      path "/te:te/te:tunnels/te:tunnel-p2mp/te:name";
    }
    description
      "This type is used by data models that need to reference
       configured P2MP TE tunnel.";
    reference "RFC4875";
  }

  typedef path-ref {
    type union {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
               "te:p2p-primary-paths/te:p2p-primary-path/te:name";
      }
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
             "te:p2p-secondary-paths/te:p2p-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.";
  }

  /**
   * TE tunnel generic groupings
   */
  grouping p2p-secondary-path-properties {
    description "tunnel path properties.";
    uses p2p-path-properties;
    uses path-constraints-common;
    uses protection-restoration-properties;
    uses p2p-path-properties-state;
  }

  grouping p2p-primary-path-properties {
    description
      "TE tunnel primary path properties grouping";
    uses p2p-path-properties;
    uses path-constraints-common;
    uses p2p-path-properties-state;
  }

  grouping path-properties {
    description "TE computed path properties grouping";
    container path-properties {
      description "The TE path computed properties";
      list path-metric {
        key metric-type;
        description "TE path metric type";
        leaf metric-type {
          type identityref {
            base te-types:path-metric-type;
          }
          description "TE path metric type";
        }
        leaf accumulative-value {
          type uint64;
          description "TE path metric accumulative value";
        }
      }
      uses te-types:generic-path-affinities;
      uses te-types:generic-path-srlgs;
      container path-route-objects {
        config 'false';
        description
          "Container for the list of computed route objects
           as returned by the computation engine";
```

```
        list path-route-object {
          key index;
          ordered-by user;
          description
            "List of computed route objects returned by the
             computation engine";
          leaf index {
            type uint32;
            description
              "Route object entry 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;
        }
      }
      uses shared-resources-tunnels;
    }
  }

  grouping p2p-path-properties-state {
    description "TE per path state parameters";
    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";
        }
      }
    }
    uses computed-path-error-info;
    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 compatutation was performed.";
        }
      }
    }
    container lsps {
      config 'false';
      description "TE LSPs container";
      list lsp {
        key "lsp-id";
        description "List of LSPs associated with the tunnel.";

        uses lsp-provisioning-error-info;
        uses lsp-properties-state;
        uses shared-resources-tunnels-state;
        uses lsp-record-route-information-state;
        uses path-properties {
          description "The TE path actual properties";
        }
      }
    }
  }

  grouping computed-path-error-info {
    description
      "Grouping for path computation error information";
    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 compuation.";
        }
        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 compuation.";
        }
        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 p2p-path-properties-common {
    description
      "TE tunnel common path properties configuration grouping";
```

```
    leaf name {
      type string;
      description "TE path name";
    }
    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      default te-types:path-setup-static;
      description
        "Signaling protocol used to set up this tunnel";
    }
    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).";
    }
    leaf path-computation-server {
      when "../path-computation-method = "+
      "'te-types:path-externally-queried'" {
        description
          "The path-computation server when the path is
           externally queried";
      }
      type inet:ip-address;
      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 "../path-computation-method =" +
      " 'te-types:path-locally-computed'";
      type boolean;
      default 'true';
      description "A CSPF dynamically computed path";
    }
```

```
    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";
    }
  }

  grouping p2p-reverse-path-properties {
    description
      "TE tunnel reverse path properties configuration
       grouping";
    uses p2p-path-properties-common;
    uses te-types:generic-path-optimization;
    leaf named-path-constraint {
      if-feature te-types:named-path-constraints;
      type leafref {
        path "../../../../../../globals/"
        + "named-path-constraints/named-path-constraint/"
        + "name";
      }
      description
        "Reference to a globally defined named path
        constraint set";
    }
  }

  grouping p2p-primary-reverse-path-properties {
    description "TE P2P tunnel primary reverse path properties.";
    reference "RFC7551";
    container p2p-primary-reverse-path {
      description "Tunnel reverse primary path properties";
      uses p2p-reverse-path-properties;
      uses path-constraints-common;
      uses p2p-path-properties-state;
      container p2p-secondary-reverse-path {
        description "Tunnel reverse secondary path properties";
        uses p2p-secondary-reverse-path-properties;
      }
    }
```

```
  }

  grouping p2p-path-properties {
    description
      "TE tunnel path properties configuration grouping";
    uses p2p-path-properties-common;
    uses te-types:generic-path-optimization;
    leaf preference {
      type uint8 {
        range "1..255";
      }
      default 1;
      description
        "Specifies a preference for this path. The lower the
        number higher the preference";
    }
    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";
    }
    leaf named-path-constraint {
      if-feature te-types:named-path-constraints;
      type leafref {
        path "../../../../../globals/"
        + "named-path-constraints/named-path-constraint/"
        + "name";
      }
      description
        "Reference to a globally defined named path
        constraint set";
    }
  }

  grouping hierarchical-link-properties {
    description
      "Hierarchical link grouping";
    reference "RFC4206";
    container hierarchical-link {
      description
        "Identifies a hierarchical link (in client layer)
         that this tunnel is associated with.";
      leaf local-te-node-id {
        type te-types:te-node-id;
        default "0.0.0.0";
```

```
        description
          "Local TE node identifier";
      }
      leaf local-te-link-tp-id {
        type te-types:te-tp-id;
        default 0;
        description
          "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;
    }
  }

  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 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 extenal
        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 extenal 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 p2p-dependency-tunnels-properties {
    description
      "Groupong for tunnel dependency list of tunnels";
    container dependency-tunnels {
      description "Dependency tunnels list";
      list dependency-tunnel {
        key "name";
        description "Dependency tunnel entry";
        leaf name {
          type leafref {
            path "../../../../../tunnels/tunnel/name";
            require-instance 'false';
          }
          description "Dependency tunnel name";
        }
        leaf encoding {
          type identityref {
            base te-types:lsp-encoding-types;
          }
          default te-types:lsp-encoding-packet;
          description "LSP encoding type";
          reference "RFC3945";
        }
        leaf switching-type {
```

```
          type identityref {
            base te-types:switching-capabilities;
          }
          default te-types:switching-psc1;
          description "LSP switching type";
          reference "RFC3945";
        }
      }
    }
  }

  grouping tunnel-p2p-config {
    description
      "Configuration parameters relating to TE tunnel";
    leaf name {
      type string;
      description "TE tunnel name.";
    }
    leaf identifier {
      type uint16;
      description
        "TE tunnel Identifier.";
      reference "RFC3209";
    }
    leaf description {
      type string;
      default 'None';
      description
        "Textual description for this TE tunnel";
    }
    leaf encoding {
      type identityref {
        base te-types:lsp-encoding-types;
      }
      default te-types:lsp-encoding-packet;
      description "LSP encoding type";
      reference "RFC3945";
    }
    leaf switching-type {
      type identityref {
        base te-types:switching-capabilities;
      }
      default te-types:switching-psc1;
      description "LSP switching type";
      reference "RFC3945";
    }
    leaf provisioning-state {
      type identityref {
```

```
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      description "TE tunnel administrative state.";
    }
    leaf preference {
      type uint8 {
        range "1..255";
      }
      default 100;
      description
        "Specifies a preference for this tunnel.
         A lower number signifies a better preference";
    }
    leaf reoptimize-timer {
      type uint16;
      units seconds;
      description
       "frequency of reoptimization of a traffic engineered LSP";
    }
    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 ID";
    }
    leaf src-tp-id {
      type yang:hex-string;
      default '00:00:00:00';
      description
        "TE tunnel source termination point identifier.";
    }
    leaf dst-tp-id {
      type yang:hex-string;
      default '00:00:00:00';
      description
        "TE tunnel destination termination point identifier.";
    }
    leaf bidirectional {
      type boolean;
      default 'false';
      description "TE tunnel bidirectional";
    }
    uses tunnel-p2p-associations-properties;
    uses protection-restoration-properties;
    uses te-types:tunnel-constraints;
```

```
    uses p2p-dependency-tunnels-properties;
    uses hierarchical-link-properties;
  }

  grouping tunnel-p2p-associations-properties {
    description "TE tunnel association grouping";
    container association-objects {
      description "TE tunnel associations";
      list association-object {
        key "type ID source global-source";
        description "List of association base objects";
        reference "RFC4872";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
          reference "RFC4872";
        }
        leaf ID {
          type uint16;
          description "Association ID";
          reference "RFC4872";
        }
        leaf source {
          type te-types:te-node-id;
          description "Association source";
          reference "RFC4872";
        }
        leaf global-source {
          type te-types:te-node-id;
          description "Association global source";
          reference "RFC4872";
        }
      }
      list association-object-extended {
        key "type ID source global-source extended-ID";
        description "List of extended association objects";
        reference "RFC6780";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
        }
        leaf ID {
          type uint16;
          description "Association ID";
```

```
          reference "RFC4872";
        }
        leaf source {
          type te-types:te-node-id;
          description "Association source";
        }
        leaf global-source {
          type te-types:te-node-id;
          description "Association global source";
          reference "RFC4872";
        }
        leaf extended-ID {
          type yang:hex-string;
          description "Association extended ID";
          reference "RFC4872";
        }
      }
    }
  }

  grouping path-access-segment-info {
    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";
    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
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      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
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
  }

  /* TE tunnel configuration/state grouping */
  grouping tunnel-p2mp-properties {
    description
```

```
      "Top level grouping for P2MP tunnel properties.";
    leaf name {
      type string;
      description "TE tunnel name.";
    }
    leaf identifier {
      type uint16;
      description
        "TE tunnel Identifier.";
      reference "RFC3209";
    }
    leaf description {
      type string;
      default 'None';
      description
        "Textual description for this TE tunnel";
    }
    leaf operational-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      config 'false';
      description "TE tunnel administrative state.";
    }
  }

  grouping p2p-path-candidate-secondary-path-config {
    description
      "Configuration parameters relating to a secondary path which
      is a candidate for a particular primary path";

    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      default te-types:path-setup-static;
      description
```

```
        "Signaling protocol used to set up this tunnel";
    }
  }

  grouping p2p-secondary-reverse-path-properties {
    description
      "Configuration parameters relating to a secondary path which
       is a candidate for a particular primary path";

    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      default te-types:path-setup-static;
      description
        "Signaling protocol used to set up this tunnel";
    }
  }

  grouping tunnel-p2p-properties {
    description
      "Top level grouping for tunnel properties.";
    leaf operational-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      config 'false';
      description "TE tunnel administrative state.";
    }
    uses tunnel-p2p-config;
    container p2p-primary-paths {
      description "Set of P2P primary aths container";
      list p2p-primary-path {
        key "name";
        description
          "List of primary paths for this tunnel.";
        uses p2p-primary-path-properties;
```

```
        uses p2p-primary-reverse-path-properties;
        container candidate-p2p-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-p2p-secondary-path {
            key "secondary-path";
            description
              "List of secondary paths for this tunnel.";
            uses p2p-path-candidate-secondary-path-config;

            leaf active {
              type boolean;
              config 'false';
              description
                "Indicates the current active path option that has
                been selected of the candidate secondary paths";
            }
          }
        }
      }
    }
    container p2p-secondary-paths {
      description "Set of P2P secondary paths container";
      list p2p-secondary-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses p2p-secondary-path-properties;
      }
    }
  }

  grouping shared-resources-tunnels-state {
    description
      "The specific tunnel that is using the shared secondary path
       resources";
    leaf lsp-shared-resources-tunnel {
      type tunnel-ref;
      description
```

```
        "Reference to the tunnel that sharing secondary path
        resources with this tunnel";
    }
  }
  grouping shared-resources-tunnels {
    description
      "Set of tunnels that share secondary path resources with
      this tunnnel";
    container shared-resources-tunnels {
      description
        "Set of tunnels that share secondary path resources with
        this tunnnel";
      leaf-list lsp-shared-resources-tunnel {
        type tunnel-ref;
        description
          "Reference to the tunnel that sharing secondary path
          resources with this tunnel";
      }
    }
  }

  grouping tunnel-actions {
    description "Tunnel actions";
    action tunnel-action {
      description "Tunnel action";
      input {
        leaf action-type {
          type identityref {
            base te-types:tunnel-action-type;
          }
          description "Tunnel action type";
        }
      }
      output {
        leaf action-result {
          type identityref {
            base te-types:te-action-result;
          }
          description "The result of the RPC 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 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 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 "recorded route information grouping";
    container lsp-record-route-information {
      description "RSVP recorded route object information";
      list lsp-record-route-information {
        when "../../origin-type = 'ingress'" {
          description "Applicable on ingress LSPs only";
        }
        key "index";
        description "Record route list entry";
        uses te-types:record-route-state;
      }
    }
  }

  grouping lsps-state-grouping {
    description
      "LSPs state operational data grouping";
    container lsps-state {
      config 'false';
      description "TE LSPs state container";
      list lsp {
        key
          "source destination tunnel-id lsp-id "+
          "extended-tunnel-id";
        description "List of LSPs associated with the tunnel.";
```

```
        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-properties {
    description
      "Global named SRLGs configuration grouping";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
        interface named srlg";
    }
    leaf group {
      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";
    }
  }

  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";
        uses named-srlgs-properties;
      }
    }
  }

  /* Global named paths constraints configuration data */
  grouping path-constraints-state {
    description "TE path constraints state";
    leaf bandwidth {
      type te-types:te-bandwidth;
      config 'false';
      description
        "A technology agnostic requested bandwidth to use
         for path computation";
    }
    leaf disjointness-type {
      type te-types:te-path-disjointness;
```

```
      config 'false';
      description
        "The type of resource disjointness.";
    }
  }

  grouping path-constraints-common {
    description
      "Global named path constraints configuration
       grouping";
    uses te-types:common-path-constraints-attributes;
    uses te-types:generic-path-disjointness;
    uses te-types:path-constraints-route-objects;
    uses shared-resources-tunnels {
      description
        "Set of tunnels that are allowed to share secondary path
         resources of this tunnel";
    }
    uses path-access-segment-info {
      description
        "Tunnel constraints induced by other segments.";
    }
  }

  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 "P2P TE tunnels list.";
        uses tunnel-p2p-properties;
        uses tunnel-actions;
        uses tunnel-protection-actions;
      }
      list tunnel-p2mp {
        key "name";
        unique "identifier";
        description "P2MP TE tunnels list.";
        uses tunnel-p2mp-properties;
      }
    }
  }

  /* 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
          "Tunnel endpoint address extracted from
          SESSION object";
        reference "RFC3209";
      }
      leaf tunnel-id {
        type uint16;
        description
          "Tunnel identifier used in the SESSION
          that remains constant over the life
          of the tunnel.";
        reference "RFC3209";
      }
      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 extended-tunnel-id {
        type yang:dotted-quad;
         description
          "Extended Tunnel ID of the LSP.";
        reference "RFC3209";
      }
      leaf operational-state {
        type identityref {
          base te-types:lsp-state-type;
        }
        description "LSP operational state.";
      }
      leaf path-setup-protocol {
        type identityref {
          base te-types:path-signaling-type;
        }
        default te-types:path-setup-static;
        description
          "Signaling protocol used to set up this tunnel";
      }
      leaf origin-type {
        type enumeration {
          enum ingress {
            description
              "Origin ingress";
          }
```

```
        enum egress {
          description
            "Origin egress";
        }
        enum transit {
          description
            "transit";
        }
      }
      default 'ingress';
      description
        "Origin type of 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 type";
      reference "RFC4872, section 4.2.1";
    }

    uses protection-restoration-properties-state;
  }
  /*** End of TE global groupings ***/

  /**
   * TE configurations container
   */
  container te {
    presence "Enable TE feature.";
    description
       "TE global container.";

    /* TE Global Configuration Data */
```

```
    uses globals-grouping;

    /* TE Tunnel Configuration Data */
    uses tunnels-grouping;

    /* TE LSPs State Data */
    uses lsps-state-grouping;

  }

  /* TE Global RPCs/execution Data */
  rpc globals-rpc {
    description
      "Execution data for TE global.";
  }

  /* TE interfaces RPCs/execution Data */
  rpc interfaces-rpc {
    description
      "Execution data for TE interfaces.";
  }

  /* TE Tunnel RPCs/execution Data */
  rpc tunnels-rpc {
    description "TE tunnels RPC nodes";
    input {
      container tunnel-info {
        description "Tunnel Identification";
        choice type {
          description "Tunnel information type";
          case tunnel-p2p {
            leaf p2p-id {
              type tunnel-ref;
              description "P2P TE tunnel";
            }
          }
          case tunnel-p2mp {
            leaf p2mp-id {
              type tunnel-p2mp-ref;
              description "P2MP TE tunnel";
            }
          }
        }
      }
    }
    output {
      container result {
        description
```

```
          "The container result of the RPC operation";
        leaf result {
          type enumeration {
            enum success {
              description "Origin ingress";
            }
            enum in-progress {
              description "Origin egress";
            }
            enum fail {
              description "transit";
            }
          }
          description "The result of the RPC operation";
        }
      }
    }
  }

  /* TE Global Notification Data */
  notification globals-notif {
    description
      "Notification messages for Global TE.";
  }

  /* TE Tunnel Notification Data */
  notification tunnels-notif {
    description
      "Notification messages for TE tunnels.";
  }
}
<CODE ENDS>
```

Figure 7: TE generic YANG module

The TE device 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 [I-D.ietf-teas-yang-te-types]

- ietf-te defined in this document

```
   <CODE BEGINS> file "ietf-te-device@2019-11-02.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 generic types */
     import ietf-te {
       prefix te;
       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 "draft-ietf-teas-yang-te-types: A YANG Data Model for
                  Common Traffic Engineering Types";
     }

     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-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, RPC and notifications. 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 "2019-11-02" {
    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-state {
        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
              "lsp life time";
          }

          leaf time-to-install {
            type uint32;
            units seconds;
            description
              "lsp installation delay time";
          }

          leaf time-to-destroy {
            type uint32;
            units seconds;
            description
              "lsp expiration delay time";
          }
        }

        container downstream-info {
          when "../te:origin-type != 'egress'" {
            description "Applicable to ingress LSPs only";
          }
          description
            "downstream information";

          leaf nhop {
            type inet:ip-address;
            description
              "downstream nexthop.";
          }

          leaf outgoing-interface {
            type if:interface-ref;
            description
              "downstream interface.";
          }
```

```
        leaf neighbor {
          type inet:ip-address;
          description
            "downstream neighbor.";
        }

        leaf label {
          type rt-types:generalized-label;
          description
            "downstream label.";
        }
      }

      container upstream-info {
        when "../te:origin-type != 'ingress'" {
          description "Applicable to non-ingress LSPs only";
        }
        description
          "upstream information";

        leaf phop {
          type inet:ip-address;
          description
            "upstream nexthop or previous-hop.";
        }

        leaf neighbor {
          type inet:ip-address;
          description
            "upstream neighbor.";
        }

        leaf label {
          type rt-types:generalized-label;
          description
            "upstream label.";
        }
      }
    }

    /**
     * Device general groupings.
     */
    grouping tunnel-device-config {
      description "Device TE tunnel configs";
      leaf path-invalidation-action {
        type identityref {
          base te-types:path-invalidation-action-type;
```

```
        }
        description "Tunnel path invalidtion action";
      }
    }

    grouping lsp-device-timers-config {
      description "Device TE LSP timers configs";
      leaf lsp-install-interval {
        type uint32;
        units seconds;
        description
          "lsp installation delay time";
      }
      leaf lsp-cleanup-interval {
        type uint32;
        units seconds;
        description
          "lsp cleanup delay time";
      }
      leaf lsp-invalidation-interval {
        type uint32;
        units seconds;
        description
          "lsp path invalidation before taking action delay time";
      }
    }

    /**
     * TE global device generic groupings
     */

    /* TE interface container data */
    grouping interfaces-grouping {
      description
        "Interface TE 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 generic 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 administrativei
                  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 "Interface TE metric grouping";
        leaf te-metric {
          type te-types:te-metric;
          description "Interface TE 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-up flooding trigger";
            }
            enum threshold-up {
              description
                "Bandwidth reservation up threshold";
            }
            enum threshold-down {
              description
                "Bandwidth reservation down threshold";
            }
            enum bandwidth-change {
              description "Banwidth 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 advertized-level-areas {
          key level-area;
          description
            "List of areas the TE interface is advertised
            in";
          leaf level-area {
            type uint32;
            description
              "The IGP area or level where the TE
              interface 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-config;
     }

     /* TE tunnels device configuration augmentation */
     augment "/te:te/te:tunnels/te:tunnel" {
       description
```

```
         "Tunnel device dependent augmentation";
       uses lsp-device-timers-config;
     }

     /* TE LSPs device state augmentation */
     augment "/te:te/te:lsps-state/te:lsp" {
        description
          "LSP device dependent augmentation";
        uses lsps-device-state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
       "/te:p2p-secondary-path/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device-state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
       "/te:p2p-primary-path/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device-state;
     }

     /* TE interfaces RPCs/execution Data */
     rpc interfaces-rpc {
       description
         "Execution data for TE interfaces.";
     }

     /* TE Interfaces Notification Data */
     notification interfaces-notif {
       description
         "Notification messages for TE interfaces.";
     }
   }
   <CODE ENDS>
```

Figure 8: TE device specific YANG module

5. 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
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-te-device
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:     ietf-te
reference:  RFCXXXX

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

6. 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 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. Following 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 configured TE tunnels on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/lsps-state": This list specifies the state derived LSPs. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/interfaces": This list specifies the configured TE interfaces on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

7. 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 also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, Anurag Sharma, and Xian Zhang for their comments and providing valuable feedback on this document.

8. 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

9. References

9.1. Normative References

[I-D.ietf-teas-yang-path-computation]
Busi, I. and S. Belotti, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-06 (work in progress), July 2019.

[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-11 (work in progress), July 2019.

[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-11 (work in progress), October 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>.

[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., 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>.

[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>.

9.2. Informative References

[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>.

Authors' Addresses

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

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin
Individual

Email: i_bryskin@yahoo.com

TEAS Working Group — Xufeng Liu
Internet Draft — Volta Networks
Intended status: Standards Track — 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

Copyright Notice

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

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

```
                                | +---+          __
                                | |   | TE Node  \/ TTP  o LTP
                                | +---+
                                |
                                | ----- TE Link
                                | ***** Node Connectivity Matrix,
                                |       TTP Local Link Connectivity
                                | @@@@@ TE Tunnel
                                o----------------------------------

      Node-1                                            Node-3
   +------------+                                    +------------+
   |    TTP-1   |                                    |    TTP-1   |
   |LTP  __     |           TE-Tunel-1               |     __     |
   |-6   \/@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\/     |
   o    *  *    oLTP-1           Node-2         LTP-6o    *  *    o
   |   *    *   |           +------------+           |   *    *   |
   |  * TTP-2*  |           |            |           |  * TTP-2*  |
   | *   __   * |LTP-2 LTP-6|            |LTP-1 LTP-5| *   __   * |
   o*    \/    *o-----------o************o-----------o*    \/    *o
   |LTP *  *    | Link-12   |          * | Link-23   |    *  *    |
   |-5 *    *   |      LTP-5|        *   |LTP-2      |   *    *   |
   +--o------o--+           o************o           +--o------o--+
    LTP-4  LTP-3            |  *   *    * |            LTP-4  LTP-3
                            |  **     *  |
                            +--o------o--+
                             LTP-4  LTP-3
```

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.

```
                                |                 |
                                | +---+          ---
                                | |   | TE Node  \ / Transitional
                                | +---+           |  Link
                                |
                                | ----- Client Layer Link
                                | ===== Server Layer Link
                                | ***** Layer Boundary
                                o-----------------------------------

      +------------------+
      | +------+         |                          +------+
   -----|Client|------+  |              Client -----|Client|
      | |Layer |---+  |  |              Layer       |Layer |
   -----|Switch|-+ |  |  |              Links  -----|Node  |
      | +------+ | |  |  |                          +------+
      |          | |  |  | Client                    | |  |
      |          | | ---_| Layer                     --- ---
   ***|**********|*| \ /*|***************************\ /*\ /****
      |          ---  |  | Server       Transitional  |   |
      |    Layer \ /  |  | Layer              Links  |   |
      |    Term.  |   |  |                            |   |
      |           |   |  |                            |   |
      |         +------+ |                          +------+
   =============|Server|=====           Server ====|Server|====
      |         |Layer | |             Layer      |Layer |
   =============|Switch|=====           Links  ====|Node  |====
      |         +------+ |                          +------+
      +------------------+

       Physical Node View                      TE-Topology View
```

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.

```
                                | +---+          __
                                | |   | TE Node  \/ TTP  o LTP
                                | +---+
                                |
                                | ----- TE Link
                                | ***** TTP Local Link Connectivity
                                o----------------------------------

             (IL-1) C-LTP-1 +------------+   C-LTP-2 (IL-1)
                    --------O   (IL-1)   O--------
             (IL-1) C-LTP-3 |   S-TTP-1  |   C-LTP-4 (IL-1)
                    --------O     __     0--------
             (IL-1) C-LTP-5 |    *\/*    |   C-LTP-5 (IL-1)
                    --------O   *    *   O--------
                            |  *(IL-1)*  |
                    S-LTP-3 | * S-TTP-2* |   S-LTP-4
                    --------o*    __    *o--------
                            |    *\/*    |
                            |   *    *   |
                            +--o------o--+
                       S-LTP-1 |      | S-LTP-2
```

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.

```
                                |  +---+
                                |  |   | TE Node
                                |  +---+
                                |  ----- TE Link
                                o---------------------------------

     +---+              +---+
     |s31|--------------|S5 |
     +---+\          /  +---+
           \        /
            \      /
             \+---+/                +---+
             /|AN1|\----------------|S8 |
            / +---+ \               +---+
     +---+ /         \ +---+
     |S9 |-------------|S11|
     +---+             +---+
           Abstract TE Topology

            +---+                    +---+
            |S1 |--------------------|S2 |
            +---+                    +---+
             /                          \
            /                            \
     +---+ /                  +---+       \ +---+
     |s3 |--------------------|S4 |---------|S5 |
     +---+\                   +---+         +---+
           \                      \             \
            \                      \             \
             \+---+                 +---+         +---+
             /|S6 |\                |S7 |---------|S8 |
            / +---+ \               +---+\       /+---+
     +---+ /         \ +---+              +---+ /
     |S9 |-------------|S10|--------------|S11|/
     +---+             +---+              +---+
           Native 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.

```
                                | +---+
                                | |   | TE Node
                                | +---+
                                | ----- TE Link
                                o----------------------------------

     +---+       +---+        +---+         +---+         +---+
     | R1|-------| R2|--------| R3|---------| R4|---------| R5|
     +---+       +---+        +---+         +---+         +---+
       |                      /   \         /   \         /
       |                     /     \       /     \       /
       |                    /       \     /       \     /
       |                   /         \   /         \   /
       |                  /           \ /           \ /
     +---+             +---+         +---+         +---+
     | R6|-------------| R7|         | R8|---------| R9|
     +---+             +---+         +---+         +---+
```

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.

```
          ---------------                   |  [ ] TE Node
          | Native        |                 |  +++ TE Link
          | TE-Topology |                   o--------------
          ---------------

      [R1] ++++ [R2] ++++ [R3] ++++ [R4] ++++ [R5]
        +                +    +    +    +    +
        +               +      +  +      +  +
        +              +        ++        ++
      [R6] +++++++++ [R7]      [R8] ++++ [R9]
```

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.

```
                                     :
   TE info distributed via ISIS-TE   :  TE info distributed via OSPF-TE
                                     :
          +---+       +---+        +---+         +---+         +---+
          | R1|-------| R2|--------| R3|---------| R4|---------| R5|
          +---+       +---+        +---+         +---+         +---+
            |                      / : \         /   \         /
            |                     /  :  \       /     \       /
            |                    /   :   \     /       \     /
            |                   /    :    \   /         \   /
            |                  /     :     \ /           \ /
          +---+             +---+    :    +---+         +---+
          | R6|-------------| R7|    :    | R8|---------| R9|
          +---+             +---+    :    +---+         +---+
                                     :
```

Figure 6a: Example Network Topology

```
           -----------------------   :  -----------------------
           |Native TE Topology   |   :  |Native TE Topology   |
           |Info-Source: ISIS-TE |   :  |Info-Source: OSPF-TE |
           -----------------------   :  -----------------------
                                     :
            [R1] ++++ [R2] ++++ [R3] : [R3’] ++++ [R4] ++++ [R5]
             +                +      :      +    +    +    +
             +               +       :       +  +      +  +
             +              +        :        ++        ++
            [R6] +++++++++ [R7]      :        [R8] ++++ [R9]
```

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.

```
                                  | +---+          /-\
                                  | |   | Router  (   ) WDM
                                  | +---+ Node     \-/  node
                                  |
                                  o----------------------------

     +---+        /-\          /-\           /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/          \-/           \-/          +---+
                              /   \         /   \
                             /     \       /     \
                            /       \     /       \
                           /         \   /         \
                          /           \ /           \
         +---+          /-\           /-\           /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

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.

```
                                            | *****  B-F WDM Path
                                            | @@@@@  B-E WDM Path
                                            | $$$$$  A-E WDM Path
                                            o--------------------

     +---+        /-\ $$$$$$$$ /-\ $$$$$$$$$ /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/         @\-/ @@@@@@@@@ \-/          +---+
                             @/   \         /   \
                            @/     \       /     \
                           @/       \     /       \
                          @/         \   /         \
                         @/           \ /           \
         +---+          /-\ ********* /-\ ********* /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

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.

```
     /---\      +---+    +---+      +---+    +---+      /---\
     |s3 |------|S13|----|S15|------|S23|----|S25|------|C21|
     \---/      +---+\   +---+      +---+   /+---+      \---/
                      \                    /
                       \                  /
                        \+---+      +---+/   +---+      /---\
                         |S18|------|S24|    |S28|------|C22|
                         +---+      +---+\  /+---+      \---/
                                          \/
                                          /\
     /---\      +---+    +---+      +---+/  \+---+      /---\
     |C12|------|S19|----|S17|------|S29|----|S27|------|C23|
     \---/      +---+    +---+      +---+    +---+      \---/


        Domain 1 TE Topology              Domain 2 TE Topology

             +---+    +---+                  +---+    +---+
        -----|S13|----|S15|----          ----|S23|----|S25|----
             +---+\   +---+                  +---+   /+---+
                   \                                /
                    \                              /
                     \+---+                  +---+/   +---+
                      |S18|----          ----|S24|    |S28|----
                      +---+                  +---+\  /+---+
                                                   \/
                                                   /\
             +---+    +---+                  +---+/  \+---+
        -----|S19|----|S17|----          ----|S29|----|S27|----
             +---+    +---+                  +---+    +---+
```

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

```
      Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1

             +---+    +---+                  +---+    +---+
        -----|S13|----|S15|----          ----|S23|----|S25|----
             +---+\   +---+                  +---+   /+---+
                   \                                /
                    \                              /
                     \+---+                  +---+/   +---+
                      |S18|----          ----|S24|    |S28|----
                      +---+                  +---+\  /+---+
                                                   \/
                                                   /\
             +---+    +---+                  +---+/  \+---+
        -----|S19|----|S17|----          ----|S29|----|S27|----
             +---+    +---+                  +---+    +---+


      Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1

             +------------+                  +------------+
        -----|            |----          ----|            |----
             |            |                  |            |
             |     AN1    |----          ----|     AN1    |----
             |            |                  |            |
        -----|            |----          ----|            |----
             +------------+                  +------------+
```

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.

```
             Client's Merged TE Topology 2

  /---\      +------------+      +------------+      /---\
  |s3 |------|            |------|            |------|C21|
  \---/      |            |      |            |      \---/
             |            |      |            |
             |            |      |            |      /---\
             |    AN11    |------|    AN21    |------|C22|
             |            |      |            |      \---/
             |            |      |            |
  /---\      |            |      |            |      /---\
  |C12|------|            |------|            |------|C23|
  \---/      +------------+      +------------+      \---/


             Client's Merged TE Topology 3

  /---\      +------------+      +---+    +---+      /---\
  |s3 |------|            |------|S23|----|S25|------|C21|
  \---/      |            |      +---+   /+---+      \---/
             |            |             /
             |            |            /
             |            |      +---+/   +---+      /---\
             |    AN11    |------|S24|    |S28|------|C22|
             |            |      +---+\  /+---+      \---/
             |            |            \/
             |            |            /\
  /---\      |            |      +---+/  \+---+      /---\
  |C12|------|            |------|S29|----|S27|------|C23|
  \---/      +------------+      +---+    +---+      \---/
```

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].

```
+------------------------+
|                        |
| Network Topology Model |
| (ietf-network-topology)|
+------------------------+
             |
             |
             |
             V
+------------------------+
|       TE Topology      |
|         Model          |
|                        |
+------------------------+
```

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.

```
                  +------------------------------+
                  |       TE Topology Model      |
                  |  (Defined in This Document)  |
                  +------------------------------+
                                 |
            +-------------+-------------+-------------+
            |             |             |             |
            V             V             V             V
      +-------------+                           +-------------+
      | Technology  |                           | Technology  |
      |  Specific   |    ......................  |  Specific   |
      | TE Topology |                           | TE Topology |
      |   Model 1   |                           |   Model n   |
      +-------------+                           +-------------+
```

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
        ...........
        |  |  +--ro 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 admin-status?           te-types:te-admin-status
      |  |  ....................
      |  +--rw underlay-topology {te-topology-hierarchy}?
      |     +--rw network-ref?   leafref

augment /nw:networks/nw:network/nt:link:
   +--rw te!
      +--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
         "Describining 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
                   destinatioin 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 terminiation 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] Shiomoto, 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., Shiomoto, 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 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 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 generic?   int32
         |  |     |        +--rw range-bitmap?   yang:hex-string
         |  |     +--rw to
         |  |     |  +--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 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-
   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 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?
           |  |  |     |     |             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?
           |  |  |  |  |       -> /nw:networks/network/network-id
           |  |  |  |  +--ro path-element* [path-element-id]
           |  |  |  |     +--ro path-element-id             uint32
           |  |  |  |     +--ro (type)?
           |  |  |  |        +--:(numbered-node-hop)
           |  |  |  |        |  +--ro numbered-node-hop
           |  |  |  |        |     +--ro node-id     te-node-id
           |  |  |  |        |     +--ro hop-type?   te-hop-type
           |  |  |  |        +--:(numbered-link-hop)
           |  |  |  |        |  +--ro numbered-link-hop
           |  |  |  |        |     +--ro link-tp-id    te-tp-id
           |  |  |  |        |     +--ro hop-type?     te-hop-type
           |  |  |  |        |     +--ro direction?    te-link-direction
           |  |  |  |        +--:(unnumbered-link-hop)
           |  |  |  |        |  +--ro unnumbered-link-hop
           |  |  |  |        |     +--ro link-tp-id    te-tp-id
           |  |  |  |        |     +--ro node-id       te-node-id
           |  |  |  |        |     +--ro hop-type?     te-hop-type
           |  |  |  |        |     +--ro direction?    te-link-direction
           |  |  |  |        +--:(as-number)
           |  |  |  |        |  +--ro as-number-hop
           |  |  |  |        |     +--ro as-number    inet:as-number
           |  |  |  |        |     +--ro hop-type?    te-hop-type
           |  |  |  |        +--:(label)
           |  |  |  |           +--ro label-hop
           |  |  |  |              +--ro te-label
           |  |  |  |                 +--ro (technology)?
```

```
        |  |  |  |                 |  +--:(generic)
        |  |  |  |                 |     +--ro generic?
        |  |  |  |                 |             rt-types:generalized-
label
        |  |  |  |                 +--ro direction?
        |  |  |  |                         te-label-direction
        |  |  |  +--ro backup-path* [index]
        |  |  |     +--ro index            uint32
        |  |  |     +--ro network-ref?
        |  |  |     |       -> /nw:networks/network/network-id
        |  |  |     +--ro path-element* [path-element-id]
        |  |  |        +--ro path-element-id              uint32
        |  |  |        +--ro (type)?
        |  |  |           +--:(numbered-node-hop)
        |  |  |           |  +--ro numbered-node-hop
        |  |  |           |     +--ro node-id     te-node-id
        |  |  |           |     +--ro hop-type?   te-hop-type
        |  |  |           +--:(numbered-link-hop)
        |  |  |           |  +--ro numbered-link-hop
        |  |  |           |     +--ro link-tp-id    te-tp-id
        |  |  |           |     +--ro hop-type?     te-hop-type
        |  |  |           |     +--ro direction?    te-link-direction
        |  |  |           +--:(unnumbered-link-hop)
        |  |  |           |  +--ro unnumbered-link-hop
        |  |  |           |     +--ro link-tp-id    te-tp-id
        |  |  |           |     +--ro node-id       te-node-id
        |  |  |           |     +--ro hop-type?     te-hop-type
        |  |  |           |     +--ro direction?    te-link-direction
        |  |  |           +--:(as-number)
        |  |  |           |  +--ro as-number-hop
        |  |  |           |     +--ro as-number    inet:as-number
        |  |  |           |     +--ro hop-type?    te-hop-type
        |  |  |           +--:(label)
        |  |  |              +--ro label-hop
        |  |  |                 +--ro te-label
        |  |  |                    +--ro (technology)?
        |  |  |                    |  +--:(generic)
        |  |  |                    |     +--ro generic?
        |  |  |                    |             rt-types:generalized-
label
        |  |  |                    +--ro direction?
```

```
     |  |  |  |                          te-label-direction
     |  |  |  +--ro protection-type?            identityref
     |  |  |  +--ro tunnel-termination-points
     |  |  |  |  +--ro source?        binary
     |  |  |  |  +--ro destination?   binary
     |  |  |  +--ro tunnels
     |  |  |     +--ro sharing?   boolean
     |  |  |     +--ro tunnel* [tunnel-name]
     |  |  |        +--ro tunnel-name    string
     |  |  |        +--ro sharing?       boolean
     |  |  +--ro path-constraints
     |  |  |  +--ro te-bandwidth
     |  |  |  |  +--ro (technology)?
     |  |  |  |     +--:(generic)
     |  |  |  |        +--ro generic?   te-bandwidth
     |  |  |  +--ro link-protection?          identityref
     |  |  |  +--ro setup-priority?           uint8
     |  |  |  +--ro hold-priority?            uint8
     |  |  |  +--ro signaling-type?           identityref
     |  |  |  +--ro path-metric-bounds
     |  |  |  |  +--ro path-metric-bound* [metric-type]
     |  |  |  |     +--ro metric-type    identityref
     |  |  |  |     +--ro upper-bound?   uint64
     |  |  |  +--ro path-affinities-values
     |  |  |  |  +--ro path-affinities-value* [usage]
     |  |  |  |     +--ro usage    identityref
     |  |  |  |     +--ro value?   admin-groups
     |  |  |  +--ro path-affinity-names
     |  |  |  |  +--ro path-affinity-name* [usage]
     |  |  |  |     +--ro usage            identityref
     |  |  |  |     +--ro affinity-name* [name]
     |  |  |  |        +--ro name    string
     |  |  |  +--ro path-srlgs-lists
     |  |  |  |  +--ro path-srlgs-list* [usage]
     |  |  |  |     +--ro usage     identityref
     |  |  |  |     +--ro values*   srlg
     |  |  |  +--ro path-srlgs-names
     |  |  |  |  +--ro path-srlgs-name* [usage]
     |  |  |  |     +--ro usage    identityref
     |  |  |  |     +--ro names*   string
     |  |  |  +--ro disjointness?            te-path-disjointness
```

```
     |  |  +--ro optimizations
     |  |  |  +--ro (algorithm)?
     |  |  |     +--:(metric) {path-optimization-metric}?
     |  |  |     |  +--ro optimization-metric* [metric-type]
     |  |  |     |  |  +--ro metric-type
     |  |  |     |  |  |       identityref
     |  |  |     |  |  +--ro weight?
     |  |  |     |  |  |       uint8
     |  |  |     |  |  +--ro explicit-route-exclude-objects
     |  |  |     |  |  |  +--ro route-object-exclude-object*
     |  |  |     |  |  |          [index]
     |  |  |     |  |  |     +--ro index
     |  |  |     |  |  |     |       uint32
     |  |  |     |  |  |     +--ro (type)?
     |  |  |     |  |  |        +--:(numbered-node-hop)
     |  |  |     |  |  |        |  +--ro numbered-node-hop
     |  |  |     |  |  |        |     +--ro node-id     te-node-id
     |  |  |     |  |  |        |     +--ro hop-type?   te-hop-type
     |  |  |     |  |  |        +--:(numbered-link-hop)
     |  |  |     |  |  |        |  +--ro numbered-link-hop
     |  |  |     |  |  |        |     +--ro link-tp-id    te-tp-id
     |  |  |     |  |  |        |     +--ro hop-type?
     |  |  |     |  |  |        |     |       te-hop-type
     |  |  |     |  |  |        |     +--ro direction?
     |  |  |     |  |  |        |             te-link-direction
     |  |  |     |  |  |        +--:(unnumbered-link-hop)
     |  |  |     |  |  |        |  +--ro unnumbered-link-hop
     |  |  |     |  |  |        |     +--ro link-tp-id    te-tp-id
     |  |  |     |  |  |        |     +--ro node-id
     |  |  |     |  |  |        |     |       te-node-id
     |  |  |     |  |  |        |     +--ro hop-type?
     |  |  |     |  |  |        |     |       te-hop-type
     |  |  |     |  |  |        |     +--ro direction?
     |  |  |     |  |  |        |             te-link-direction
     |  |  |     |  |  |        +--:(as-number)
     |  |  |     |  |  |        |  +--ro as-number-hop
     |  |  |     |  |  |        |     +--ro as-number
     |  |  |     |  |  |        |     |       inet:as-number
     |  |  |     |  |  |        |     +--ro hop-type?
     |  |  |     |  |  |        |             te-hop-type
     |  |  |     |  |  |        +--:(label)
```

```
           |  |  |     |  |  |       |  +--ro label-hop
           |  |  |     |  |  |       |     +--ro te-label
           |  |  |     |  |  |       |        +--ro (technology)?
           |  |  |     |  |  |       |        |  +--:(generic)
           |  |  |     |  |  |       |        |     +--ro generic?
           |  |  |     |  |  |       |        |             rt-
   types:generalized-label
           |  |  |     |  |  |       |        +--ro direction?
           |  |  |     |  |  |       |                te-label-direction
           |  |  |     |  |  |       +--:(srlg)
           |  |  |     |  |  |          +--ro srlg
           |  |  |     |  |  |             +--ro srlg?   uint32
           |  |  |     |  |  +--ro explicit-route-include-objects
           |  |  |     |  |     +--ro route-object-include-object*
           |  |  |     |  |             [index]
           |  |  |     |  |        +--ro index
           |  |  |     |  |        |       uint32
           |  |  |     |  |        +--ro (type)?
           |  |  |     |  |           +--:(numbered-node-hop)
           |  |  |     |  |           |  +--ro numbered-node-hop
           |  |  |     |  |           |     +--ro node-id     te-node-id
           |  |  |     |  |           |     +--ro hop-type?   te-hop-type
           |  |  |     |  |           +--:(numbered-link-hop)
           |  |  |     |  |           |  +--ro numbered-link-hop
           |  |  |     |  |           |     +--ro link-tp-id    te-tp-id
           |  |  |     |  |           |     +--ro hop-type?
           |  |  |     |  |           |     |       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
```

```
           |  |  |  +--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 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
```

```
         |  |     |        +--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 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-
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 is-allowed?         boolean
        |  |     +--ro underlay {te-topology-hierarchy}?
        |  |     |  +--ro enabled?                     boolean
        |  |     |  +--ro primary-path
        |  |     |  |  +--ro network-ref?
        |  |     |  |  |       -> /nw:networks/network/network-id
        |  |     |  |  +--ro path-element* [path-element-id]
        |  |     |  |     +--ro path-element-id              uint32
        |  |     |  |     +--ro (type)?
        |  |     |  |        +--:(numbered-node-hop)
        |  |     |  |        |  +--ro numbered-node-hop
        |  |     |  |        |     +--ro node-id     te-node-id
        |  |     |  |        |     +--ro hop-type?   te-hop-type
        |  |     |  |        +--:(numbered-link-hop)
        |  |     |  |        |  +--ro numbered-link-hop
        |  |     |  |        |     +--ro link-tp-id    te-tp-id
        |  |     |  |        |     +--ro hop-type?     te-hop-type
        |  |     |  |        |     +--ro direction?
        |  |     |  |        |             te-link-direction
        |  |     |  |        +--:(unnumbered-link-hop)
        |  |     |  |        |  +--ro unnumbered-link-hop
        |  |     |  |        |     +--ro link-tp-id    te-tp-id
        |  |     |  |        |     +--ro node-id       te-node-id
        |  |     |  |        |     +--ro hop-type?     te-hop-type
        |  |     |  |        |     +--ro direction?
        |  |     |  |        |             te-link-direction
        |  |     |  |        +--:(as-number)
        |  |     |  |        |  +--ro as-number-hop
        |  |     |  |        |     +--ro as-number    inet:as-number
        |  |     |  |        |     +--ro hop-type?    te-hop-type
```

```
           |  |     |  |        +--:(label)
           |  |     |  |           +--ro label-hop
           |  |     |  |              +--ro te-label
           |  |     |  |                 +--ro (technology)?
           |  |     |  |                 |  +--:(generic)
           |  |     |  |                 |     +--ro generic?
           |  |     |  |                 |             rt-
   types:generalized-label
           |  |     |  |                 +--ro direction?
           |  |     |  |                         te-label-direction
           |  |     |  +--ro backup-path* [index]
           |  |     |     +--ro index           uint32
           |  |     |     +--ro network-ref?
           |  |     |     |       -> /nw:networks/network/network-id
           |  |     |     +--ro path-element* [path-element-id]
           |  |     |        +--ro path-element-id              uint32
           |  |     |        +--ro (type)?
           |  |     |           +--:(numbered-node-hop)
           |  |     |           |  +--ro numbered-node-hop
           |  |     |           |     +--ro node-id     te-node-id
           |  |     |           |     +--ro hop-type?   te-hop-type
           |  |     |           +--:(numbered-link-hop)
           |  |     |           |  +--ro numbered-link-hop
           |  |     |           |     +--ro link-tp-id    te-tp-id
           |  |     |           |     +--ro hop-type?     te-hop-type
           |  |     |           |     +--ro direction?
           |  |     |           |             te-link-direction
           |  |     |           +--:(unnumbered-link-hop)
           |  |     |           |  +--ro unnumbered-link-hop
           |  |     |           |     +--ro link-tp-id    te-tp-id
           |  |     |           |     +--ro node-id       te-node-id
           |  |     |           |     +--ro hop-type?     te-hop-type
           |  |     |           |     +--ro direction?
           |  |     |           |             te-link-direction
           |  |     |           +--:(as-number)
           |  |     |           |  +--ro as-number-hop
           |  |     |           |     +--ro as-number    inet:as-number
           |  |     |           |     +--ro hop-type?    te-hop-type
           |  |     |           +--:(label)
           |  |     |              +--ro label-hop
           |  |     |                 +--ro te-label
```

```
     |  |     |  |                  +--ro (technology)?
     |  |     |  |                  |  +--:(generic)
     |  |     |  |                  |     +--ro generic?
     |  |     |  |                  |             rt-
   types:generalized-label
     |  |     |  |                  +--ro direction?
     |  |     |  |                          te-label-direction
     |  |     |  +--ro protection-type?            identityref
     |  |     |  +--ro tunnel-termination-points
     |  |     |  |  +--ro source?        binary
     |  |     |  |  +--ro destination?   binary
     |  |     |  +--ro tunnels
     |  |     |     +--ro sharing?   boolean
     |  |     |     +--ro tunnel* [tunnel-name]
     |  |     |        +--ro tunnel-name    string
     |  |     |        +--ro sharing?       boolean
     |  |     +--ro path-constraints
     |  |     |  +--ro te-bandwidth
     |  |     |  |  +--ro (technology)?
     |  |     |  |     +--:(generic)
     |  |     |  |        +--ro generic?   te-bandwidth
     |  |     |  +--ro link-protection?         identityref
     |  |     |  +--ro setup-priority?          uint8
     |  |     |  +--ro hold-priority?           uint8
     |  |     |  +--ro signaling-type?          identityref
     |  |     |  +--ro path-metric-bounds
     |  |     |  |  +--ro path-metric-bound* [metric-type]
     |  |     |  |     +--ro metric-type    identityref
     |  |     |  |     +--ro upper-bound?   uint64
     |  |     |  +--ro path-affinities-values
     |  |     |  |  +--ro path-affinities-value* [usage]
     |  |     |  |     +--ro usage    identityref
     |  |     |  |     +--ro value?   admin-groups
     |  |     |  +--ro path-affinity-names
     |  |     |  |  +--ro path-affinity-name* [usage]
     |  |     |  |     +--ro usage            identityref
     |  |     |  |     +--ro affinity-name* [name]
     |  |     |  |        +--ro name    string
     |  |     |  +--ro path-srlgs-lists
     |  |     |  |  +--ro path-srlgs-list* [usage]
     |  |     |  |     +--ro usage     identityref
```

```
     |  |     |  |     +--ro values*   srlg
     |  |     |  +--ro path-srlgs-names
     |  |     |  |  +--ro path-srlgs-name* [usage]
     |  |     |  |     +--ro usage    identityref
     |  |     |  |     +--ro names*   string
     |  |     |  +--ro disjointness?
     |  |     |          te-path-disjointness
     |  |     +--ro optimizations
     |  |     |  +--ro (algorithm)?
     |  |     |     +--:(metric) {path-optimization-metric}?
     |  |     |     |  +--ro optimization-metric* [metric-type]
     |  |     |     |  |  +--ro metric-type
     |  |     |     |  |  |       identityref
     |  |     |     |  |  +--ro weight?
     |  |     |     |  |  |       uint8
     |  |     |     |  |  +--ro explicit-route-exclude-objects
     |  |     |     |  |  |  +--ro route-object-exclude-object*
     |  |     |     |  |  |          [index]
     |  |     |     |  |  |     +--ro index
     |  |     |     |  |  |     |       uint32
     |  |     |     |  |  |     +--ro (type)?
     |  |     |     |  |  |        +--:(numbered-node-hop)
     |  |     |     |  |  |        |  +--ro numbered-node-hop
     |  |     |     |  |  |        |     +--ro node-id
     |  |     |     |  |  |        |     |       te-node-id
     |  |     |     |  |  |        |     +--ro hop-type?
     |  |     |     |  |  |        |             te-hop-type
     |  |     |     |  |  |        +--:(numbered-link-hop)
     |  |     |     |  |  |        |  +--ro numbered-link-hop
     |  |     |     |  |  |        |     +--ro link-tp-id
     |  |     |     |  |  |        |     |       te-tp-id
     |  |     |     |  |  |        |     +--ro hop-type?
     |  |     |     |  |  |        |     |       te-hop-type
     |  |     |     |  |  |        |     +--ro direction?
     |  |     |     |  |  |        |             te-link-direction
     |  |     |     |  |  |        +--:(unnumbered-link-hop)
     |  |     |     |  |  |        |  +--ro unnumbered-link-hop
     |  |     |     |  |  |        |     +--ro link-tp-id
     |  |     |     |  |  |        |     |       te-tp-id
     |  |     |     |  |  |        |     +--ro node-id
     |  |     |     |  |  |        |     |       te-node-id
```

```
           |  |     |     |  |  |        |     +--ro hop-type?
           |  |     |     |  |  |        |     |       te-hop-type
           |  |     |     |  |  |        |     +--ro direction?
           |  |     |     |  |  |        |             te-link-direction
           |  |     |     |  |  |        +--:(as-number)
           |  |     |     |  |  |        |  +--ro as-number-hop
           |  |     |     |  |  |        |     +--ro as-number
           |  |     |     |  |  |        |     |       inet:as-number
           |  |     |     |  |  |        |     +--ro hop-type?
           |  |     |     |  |  |        |             te-hop-type
           |  |     |     |  |  |        +--:(label)
           |  |     |     |  |  |        |  +--ro label-hop
           |  |     |     |  |  |        |     +--ro te-label
           |  |     |     |  |  |        |        +--ro (technology)?
           |  |     |     |  |  |        |        |  +--:(generic)
           |  |     |     |  |  |        |        |     +--ro generic?
           |  |     |     |  |  |        |        |             rt-
   types:generalized-label
           |  |     |     |  |  |        |        +--ro direction?
           |  |     |     |  |  |        |                te-label-
   direction
           |  |     |     |  |  |        +--:(srlg)
           |  |     |     |  |  |           +--ro srlg
           |  |     |     |  |  |              +--ro srlg?   uint32
           |  |     |     |  |  +--ro explicit-route-include-objects
           |  |     |     |  |     +--ro route-object-include-object*
           |  |     |     |  |             [index]
           |  |     |     |  |        +--ro index
           |  |     |     |  |        |       uint32
           |  |     |     |  |        +--ro (type)?
           |  |     |     |  |           +--:(numbered-node-hop)
           |  |     |     |  |           |  +--ro numbered-node-hop
           |  |     |     |  |           |     +--ro node-id
           |  |     |     |  |           |     |       te-node-id
           |  |     |     |  |           |     +--ro hop-type?
           |  |     |     |  |           |             te-hop-type
           |  |     |     |  |           +--:(numbered-link-hop)
           |  |     |     |  |           |  +--ro numbered-link-hop
           |  |     |     |  |           |     +--ro link-tp-id
           |  |     |     |  |           |     |       te-tp-id
           |  |     |     |  |           |     +--ro hop-type?
```

```
          |  |     |     |  |           |     |       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
     |  |          +--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
```

```
             |  |  |  +--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 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 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-
   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
             +--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
```

```
           |  |  |        |     +--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
```

```
          +--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 Working Group
Internet Draft
Intended status: Informational
Intended status: Informational
Expires: April 2017

K. Lam
E. Varma
Nokia
P. Doolan
Coriant
N. Davis
Ciena
B. Zeuner
Deutsche Telekom
M. Betts
ZTE
I. Busi
Huawei
S. Mansfield
Ericsson
R. Vilalta
CTTC
V. Lopez
Telefonica
October 27, 2016

Usage of IM for network topology to support TE Topology YANG Module Development
draft-lam-teas-usage-info-model-net-topology-04.txt

Status of this Memo

Abstract

The benefits of using a common Information Model (IM) as a foundation for deriving purpose and protocol specific interfaces, particularly

for complex networking domains, has been described in draft-betts-netmod-framework-data-schema-uml. This draft describes existing information model relevant to Network Topology and illustrates how it can be used to help ensure the consistency and completeness of the YANG data modeling for TE topologies solutions work in TEAS.

Table of Contents

1. Introduction

This draft describes existing information modeling (IM) relevant to Network Topology [ONF TR-512] [OSSDN SNOWMASS] and illustrates how it can be used to help ensure the consistency and completeness of the YANG data model (DM) for TE topologies solutions development work in TEAS.

2. Background and Motivation

Information Models (IM) and Data Models (DM) are related but different. An IM provides an abstract, conceptual view of the system being modeled in terms of its constituent parts (objects), independent of any specific implementations or protocols used to transport the data; it hides all protocol and implementation details (RFC 3444, TM Forum/NGCOR, ITU-T SG 15). A DM is a concrete specification in a particular language of an interface to, in this case, a controlled/managed system. The intention of the distinction between IMs and DMs has been to separate the modeling of problem space semantics from the modeling of the implementation of those semantics (though the dividing line has not always been clearly articulated).

A DM may be derived from an IM though it is often created without (explicit or obviously implicit) reference to one. When a DM is derived from an IM, the DM and the components of the system it provides control/management access to are traceable to the definitions provided in the IM. There is no ambiguity between designer, developer, user or operator regarding the name, function, and information elements that are associated with a particular managed object.

As described in [I-D.betts], when DMs are created "in isolation" solely for the purpose of encoding specific interfaces, they may do that job adequately for any particular interface but in complex domains may create opportunities for confusion, duplication of effort, lack of interoperability, and lack of extensibility. In the past, ad-hoc development of DMs has caused significant operational and implementation inefficiencies in our industry.

Since March 2014, upon IESG recommendation that SNMP no longer be used for new work re configuration and that NETCONF/YANG be used instead, there has been an explosion of YANG DM development in IETF. It has consequently been recognized as essential to assure proper coordination of YANG DM development (including reaching out to different SDOs/consortia), as well as to assure that the YANG modules themselves provide a good representation of what is being modeled, to meet expectations of functionality, quality, and interoperability. In order to facilitate this objective, guidance from available pertinent IMs can be valuable.

This draft first describes an existing information model relevant to Network Topology [ONF TR-512], which is part of the Common Information Model (ONF-CIM) of network resources (as described in [I-D.betts]), that can be leveraged to assess the consistency and completeness of related YANG modules under development. It also describes an transport application-specific IM [OSSDN SNOWMASS], derived from CIM pruning and refactoring as explained in [I-D.betts], that is intended to enable further clarity in understanding the modeling. Being part of a Common Information Model, it will not lead to development of incompatible/uncoordinated models that can be difficult to maintain as other purpose-specific interfaces are developed.

3. The Common Information Model

This section provides a high level introduction to the ONF Common Information Model (ONF-CIM), and in particular its Core Model (see [ONF TR-512]), to provide an overall context for the topology relevant subset. The ONF-CIM has been developed through collaboration among several SDOs, including ITU-T, TM Forum, and ONF, and also published as ITU-T Recommendation G.7711 [G.7711].

An information model describes the things in a domain in terms of objects, their properties (represented as attributes), and their relationships.

The ONF-CIM is expressed in a formal language called UML (Unified Modeling Language). UML has a number of basic model elements, called UML artifacts. In order to assure a consistent and harmonized modeling approach, only a selected subset of these UML artifacts were used in the development of the ONF-CIM according to guidelines for creating an information model expressed in UML (see the UML Guidelines document in the ONF TR-514 [ONF TR-514]).

The ONF-CIM has been developed using the Papyrus open source UML Tool, for which a detailed guidelines document is available (see the Papyrus Guidelines document in the ONF TR-515 [ONF TR-515]). This guidelines document also describes how the modelers constructing the ONF-CIM can cooperate in the GitHub environment to allow for separate and still coordinated development of the ONF-CIM fragments.

The OMF-CIM includes all of the artifacts (objects, attributes, associations, etc.) that are necessary to describe the domain for the applications being developed.

It will be necessary to continually expand and refine the ONF-CIM over time as, for example to add, new applications, capabilities or forwarding technologies, or to refine the ONF-CIM as new insights are gained. To allow these extensions to be made in a seamless manner, the ONF-CIM is structured into a number of sub-models. This modeling approach enables application specific and forwarding technology specific extensions to be developed by domain experts with appropriate independence. This approach is further articulated in ONF TR-513 [ONF TR-513] and [I-D.betts].

3.1. Core Model

The Core Model of the ONF-CIM consists of model artifacts that are intended for use by multiple applications and/or forwarding technologies.

For navigability, the Core Model is further sub-structured into sub-models. Currently, these consist of the Core Network Model (CNM), Core Foundation Model, Core Physical Model, and the Core Specification Model. The following sub-sections provide an overview of these sub-models. A detailed description is contained in ONF TR-512 [ONF TR-512].

3.1.1. Core Network Model

The Core Network Model (CNM) consists of artifacts that model the essential network aspects that are neutral to the forwarding technology of the network. The CNM currently encompasses Topology, Termination, and Forwarding aspects (subsets of the CNM) as described below:

- Topology Subset of CNM

  The Topology subset of the CNM supports the modeling of network topology information, which can be used to build the topology database and depict the topology. Object classes representing topological entities include:

  o Forwarding Domain (FD): Offers the potential to enable forwarding of information.

  o Link (L): Models the adjacency between two or more FDs. A Link has LinkPorts.

  o Logical Termination Point (LTP): Models the ports of a link. It encapsulates the termination, adaptation, and OAM functions of one or more transport layers.

  o Network Element (NE): While not actually part of topology, a NE brings meaning to the FD and the LTP contexts (and hence the links). A NE represents physical equipment "bundling" to provide a view of management scope, management access, and session.

The Topology subset of the CNM supports network topology abstraction and virtualization. FD abstraction is supported via recursive aggregation and virtualization via partitioning of resources according to the resource dedication criterion.

- Forwarding Subset of CNM

  The Forwarding subset of the CNM (not covered in detail in this draft) supports configuration of forwarding entities, including their setup, modification, and tear down. Artifacts representing the forwarding construct include:

  o ForwardingConstruct (FC): Also known as SNC. In conjunction with the FcPort, FC models the enabled forwarding between two FcPorts across a FD.

  o FcPort: Models the access to the FC, and associates the FC to the LTP. When the FC supports protection, the FcPort also indicates its role in the protection scheme, i.e., whether it is a working or protection FcPort.

  o FcRoute: Also known as SncRoute. It models the individual routes of an FC.

  o FcSwitch: Also known as SncSwitch. It models the switched forwarding of traffic (traffic flow) between EPs and is present where there is protection functionality in the FD.

- Termination Subset of CNM

  The Termination subset of the CNM (not covered in detail in this draft) supports modeling of the processing of transport characteristic information, such as termination, adaptation, OAM, etc. Artifacts representing the termination and adaptation and OAM construct include:

  o Logical Termination Point (LTP): See the LTP description in the Topology Subset o Layer Protocol (LP): This identifies the type of signal and is the anchor for transport layer protocol specific definitions, which are modeled in, e.g., [G.874.1] for OTN, [G.8052] for transport Ethernet, and [G.8152] for MPLS-TP.

- Resilience Subset of CNM

  The Resilience subset provides a view of the model for resilience (including protection and restoration) and encompasses:

  o The basic resilience model structure

  o The key attributes relevant to resilience

  o The application of the resilience model to various cases

3.1.2. Core Foundation Model

To communicate about an entity, it is important to have some way of referring to that entity, i.e., to have some way of referencing it. The Core Foundation model defines the artifacts for referencing entities; i.e.:

- Global Unique ID (GUID):

  An identifier that is globally unique where an identifier is a property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)

- Local ID:

  An identifier that is unique in the context of some scope that is less than the global scope (where an identifier is as defined in GUID above).

- Name:

A property of an entity with a value that is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.

- Label:

  A property of an entity with a value that is not expected to be unique and is allowed to change. A label carries no semantics with respect to the purpose of the entity and has no effect on the entity behavior or state.

The Core Foundation model also provides the opportunity to extend any entity using the Extension structure.

The model also defines two foundation object classes:

- GlobalClass:

  Super class of object classes for which their instances can exist on their own right, e.g. NE, LTP, FD, Link, and FC. Global classes shall have one and only one globally unique identifier (GUID) and may have zero or more local identifiers, zero or more names, zero or more labels, zero or more extensions.

- LocalClass:

  Super class of object classes for which the existence of their instances depends on instances of global classes; e.g., LP (of LTP), EP (of FC), and LE (of Link). Local classes shall have at least one local identifier, may have zero or more names, zero or more labels, zero or more extensions.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I             Artifacts for Referencing of Entities               I
I                                                                  I
```

```
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
--------------------------------------------------------------------
```

Figure 3-1 Artifacts for Referencing of Entities

The Core Foundation model also defines a State_Pac artifact, which is a package of state attributes. The State_Pac is inherited by GlobalClass and LocalClass object classes. The State_Pac consists of the following state-related attributes:

- Operational State:

  Read-only with values: DISABLED, ENABLED

- Administrative State:

  Read-write with values: LOCKED, UNLOCKED

- Lifecycle State:

  Read-write with values: PLANNED, POTENTIAL, INSTALLED, PENDING_REMOVAL

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
```

```
   I                                                                  I
   I                       States of Objects                          I
   I                                                                  I
   I                                                                  I
   I                     (only in PDF version)                        I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   --------------------------------------------------------------------
```

Figure 3-2 States of Objects

3.1.3. Core Physical Model

The Physical model provides a view of the model for physical entities (including equipment, holders and connectors). This model also specifies the relationship between the connector and the LTP, and the relationship between physical and functional views.

```
   --------------------------------------------------------------------
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                       Basic Equipment Pattern                    I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                     (only in PDF version)                        I
   I                                                                  I
```

```
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   --------------------------------------------------------------------
```

Figure 3-3 Basic Equipment Pattern

3.1.4. Core Specification Model

There are several related needs that have given rise to the Specification model:

- Provide machine readable form of specific localized behavior:
  - o Representing rules related to restrictions of specific cases of use of the model
  - o Representing capabilities of specific cases of use
- Enable the introduction of run time schema where the essential structure of the model is known up front (at compile time) but the details are not
- Reduce the clutter in a representation where a set of details take the same values for all instances that are related to a specific case
- Allow leverage of existing standards definitions (e.g., technology/application specific) in a machine readable language

The combination of the above resulted in a separation in the model of definitions of structure and content such that an instance of a class from one model fragment could have an association instance to another model fragment to enable the provision of a fragment of definition of the class and of subordinates.

The aim of all specification definitions is that they be rigorous definitions of specific cases of usage and enable machine interpretation where traditional interface designs would only allow human interpretation.

The following dedicated spec structures have been considered:

- FC spec: Main focus to provide a representation of the effective internal structure of a ForwardingConstruct (FC)

- LTP and LP spec: Main focus to provide a representation of Layer Protocol (LP) specific parameters for the Logical Termination Point (LTP)

- FD and Link spec: Main focus on capacity and forwarding enablement restrictions

- Equipment spec: Main focus to provide a representation of equipping constraints

```
   --------------------------------------------------------------------
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I          Class Diagram of the Spec Model of LTP and LP           I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                     (only in PDF version)                        I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   I                                                                  I
   --------------------------------------------------------------------
```

Figure 3-4 Class Diagram of the Spec Model of LTP and LP

3.2. Other Models

In addition to the Core Model, the ONF-CIM includes forwarding technology and application specific models. The forwarding technology models of the ONF-CIM (see [ONF TR-512]) encompasses transport technology layers 0, 1, and 2.

4. High Level Description of the Topology Subset of the CNM

This section provides a high-level overview of the Topology Subset of the CNM. Figure 4-1 below is a skeleton class diagram illustrating the key object classes. To avoid cluttering the figure, not all associations have been shown and all of the attributes were omitted.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I               Overview of the CNM Topology Subset                I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
--------------------------------------------------------------------
```

Figure 4-1 Overview of the CNM Topology Subset

4.1. Object Classes of the CNM Topology Subset

This section describes the object classes of the Topology Subset of the CNM. Relationships between these classes are described in section 4.2 below

4.1.1. LogicalTerminationPoint (LTP) and LayerProtocol (LP)

The LogicalTerminationPoint (LTP) object class encapsulates the termination, adaptation and OAM functions of one or more transport protocol layers. The structure of the LTP supports all transport protocols including circuit and packet forms. Each transport layer is represented by a LayerProtocol (LP) instance. The LayerProtocol instances of the LTP can be used for controlling the termination and OAM functionality of that layer. It can also be used for controlling the adaptation (i.e. encapsulation and/or multiplexing of client signal). Where the client/server relationship is fixed 1:1 and immutable, the different layers can be encapsulated in a single LTP instance. Where there is a n:1 relationship between client and server, the layers must be split over separate instances of LTP.

The LP object class is defined with generic attributes "layerProtocolName" for indicating the supported transport layer protocol.

Transport layer specific properties (such as layer-specific termination and adaptation properties) are modeled as attributes of conditional packages (called "_Pacs" in the UML notation of the ONF-CIM) associated with the LP object class.

4.1.2. ForwardingDomain (FD)

The ForwardingDomain (FD) object class models the switching and routing capabilities (see "subnetwork" topological component in [G.852.2] and [TMF612]), which is used to effect forwarding of transport characteristic information and offers the potential to enable forwarding. It represents the resource that supports flows across the FD. The FD object can hold zero or more instances of ForwardingConstruct (FC) (representing constrained forwarding, not discussed further in this document, covering connections, VLANs etc) of one or more layer networks; e.g., OCh, ODU, ETH, and MPLS-TP. The

FD object provides the context for operations that create/modify/delete FCs.

The FD object class supports a recursive aggregation relationship such that the internal construction of an FD can be exposed as multiple lower level FDs and associated Links (partitioning) (see section 4.2.1.)

At the lowest level of recursion, a FD (within a network element) could represent a switch matrix (i.e., a fabric).

Note that an NE can encompass multiple switch matrices (FDs), as described in section 4.2.2. An instance of FD is associated with zero or more LTP objects, as described in section 4.2.3.

4.1.3. Link and Link Port

The Link object class models the adjacency between two or more ForwardingDomains (FDs).

In its basic form (i.e., point-to-point Link) it associates a set of LTP clients on one FD with an equivalent set of LTP clients on another FD. Like the FC, the Link has endpoints (LinkPort) which take roles in the context of the function of the Link. A point-to-point Link can be a TE Link and support parameters such as capacity, delay etc. These parameters depend on the type of technology that supports the link.

A Link can be terminated on two or more FDs. This provides support for technologies such as PON and Layer 2 MAC in MAC configurations.

The LinkPort further details the relationship between FD and Link for asymmetric cases.

A FD may aggregate Links (see section 4.2.5).

The Link can support multiple transport layers via the associated LTP object. An instance of Link can be formed with the necessary properties according to the degree of virtualization. For implementation optimization, multiple layer-specific links can be merged and represented as a single Link instance.

4.1.4. Network Element (NE)

The NetworkElement (NE) object class represents a network element (traditional NE) in the data plane or a virtual network element visible in an interface where virtualization is used.

In the direct interface from a SDN controller to a network element in the data plane, the NE object defines the scope of control for the resources within the network element, e.g., internal transfer of user information between the external terminations (ports), encapsulation, multiplexing/demultiplexing, and OAM functions, etc. The NE provides the scope of the naming space for identifying objects representing the resources within the network element.

Where virtualization is employed, the NE object represents a virtual NE (VNE). The mapping of the VNE to the NEs is the internal matter of the SDN controller that offers the view of the VNE. Via the interface between hierarchical SDN controllers, NE instances can be created (or deleted) for providing (or removing) virtual views of the combination of slices of network elements in the data plane.

4.2. Relationships between Object Classes of the Topology Subset

4.2.1. ForwardingDomain Recursive Aggregation (HigherLevelFdEncompassesLowerLevelFds Aggregation)

Figure 4-2 below provides a pictorial example of ForwardingDomain (FD) recursion with Links.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
```

```
   I              ForwardingDomain recursion with Links             I
   I                                                                I
   I                                                                I
   I                                                                I
   I                                                                I
   I                     (only in PDF version)                      I
   I                                                                I
   I                                                                I
   I                                                                I
   I                                                                I
   I                                                                I
   -------------------------------------------------------------------
```

Figure 4-2 ForwardingDomain recursion with Links

Figure 4-2 shows a UML fragment including the Link and ForwardingDomain (FD). For simplicity it is assumed here that the Links and FDs are for a single LayerProtocol (LP) although it can be seen from the detailed figure earlier in this section that both a FD and link can support a list of LPs.

The pictorial form shows a number of instances of FD interconnected by Links and shows nesting of FDs. The recursive aggregation "HigherLevelFdEncompassesLowerLevelFds" relationship (represented by an open diamond) supports the FD nesting but it should be noted that this is intentionally showing no lifecycle dependency between the lower FDs and the higher ones that nest them (to do this composition, a black diamond would have been used instead of the open diamond). This is to allow for rearrangements of the FD hierarchy (e.g. when regions of a network are split or merged). This emphasizes that the nesting is an abstraction rather than decomposition. The underlying network still operates regardless of how it is perceived in terms of aggregating FDs. The model allows for only one hierarchy.

4.2.2. Network Elements encompassing ForwardingDomains (NeEncompassesFds Aggregation)

Figure 4-3 below provides a pictorial example of ForwardingDomain (FD) recursion with Links and NEs.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I          ForwardingDomain recursion with Links and NEs           I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
--------------------------------------------------------------------
```

Figure 4-3 ForwardingDomain recursion with Links and NEs

Figure 4-3 above shows an overlay of NetworkElement (NE) on the ForwardingDomains and a corresponding fragment of UML showing only the ForwardingDomain and NetworkElement classes.

The figure emphasizes that one level of abstraction of ForwardingDomain is bounded by an NE. This is represented in the UML fragment by the composition association (black diamond) that explains that there is a lifecycle dependency in that the ForwardingDomain at this level that cannot exist without the NE. The figure also shows that a ForwardingDomain need not be bounded by an NE (as explained in the UML fragment by the 0..1 composition) and that a ForwardingDomain may have smaller scope than the whole NE (even when considering only a single LayerProtocol as described below).

In one of the cases depicted (e.g., the right hand side NE encompassing two FDs), the two ForwardingDomains in the NE are completely independent. In the other cases depicted (e.g., the left hand side NE encompassing three FDs) the subordinate ForwardingDomains are themselves joined by Links emphasizing that the NE does not necessarily represent the lowest level of relevant network decomposition.

The figure also emphasizes that just because one ForwardingDomain at a particular level of decomposition of the network happens to be the one bounded by an NE does not mean that all ForwardingDomains at that level are also bounded by NEs.

4.2.3. ForwardingDomain association with LTPs (FdAggregatesLtps Composition)

An instance of FD is associated with zero or more LTP objects via the "FdAggregatesLtps" composition.

4.2.4. ForwardingDomain aggregating Links (FdEncompassesLinks)

A ForwardingDomain can aggregate links. An example of ForwardingDomain Recursive Aggregation with Links is shows in section 4.2.1 above.

However, the FdAggregatesLink association is not modeled because this association can be inferred from the higherLevelFdContainsLowerLevelFd association together with the linkHasAssociatedFds association.

4.2.5. ForwardingDomain aggregating NEs

A ForwardingDomain can aggregate Network Elements. An example of ForwardingDomain Recursive Aggregation with Links and NEs is shown in section 4.2.2 above.

However, the FdAggregatesNe association is not modeled because this association can be inferred from higherLevelFdContainsLowerLevelFd association and together with the NeEncompassesFd association.

5. Detailed Description of the Topology Subset

The two key classes related to Topology are the ForwardingDomain (FD) and the Link. For simple cases the FD represents the switching capability in the network and the Link represents adjacency. These are depicted in the context of other model classes in Figure 5-1.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I     Object Classes and Relationships in the Topology Subset      I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
--------------------------------------------------------------------
```

Figure 5-1 Object Classes and Relationships in the Topology Subset

Figure 5-1 shows a lightweight view of the model omitting the attributes (where appropriate these will be described later in this section).

The FD and Link will be described in detail later in the document. Figure 5-1 focuses on interrelationships and these will be the focus of this section. The figure shows that:

- An FD may be a subordinate part of a NetworkElement (NE) or may be larger than, and independent of, any NE.

- An FD may encompass lower level FDs. This may be such that:

  o A FD directly contained in an NE is divided into smaller parts

   - A FD not encompassed by an NE is divided into smaller parts some of which may be encompassed by NEs

   - The FD represents the whole network

- An FD encompasses Links that interconnect any FDs encompassed by the FD

- A Link may aggregate Links in several ways

   - In parallel where several links are considered as one

   - In series where Links chain to form a Link of a greater span

      - Note that this case requires further development in the model

- A Link has associated FDs that it interconnects

   - A Link may interconnect 2 or more FDs

      - Note that it is usual for a Link to interconnect 2 FDs but there are cases where many FDs may be interconnected by a Link

- A Link has LinkPorts that represent the ports of the Link itself

   - LinkPorts are especially relevant for multi-ended asymmetric Link

- A LinkPort aggregates LogicalTerminationPoints (LTPs) that bound the Link. The LTP represent a stack LayerProtocol terminations where the details of each is held in the LayerProtocol (LP). The LTP may be:

   - Part of an NE

   - Conceptually independent from any NE

- A LinkPort references LTPs on which the Link associated to the LE terminates

Both the Link and FD are subclasses of ForwardingEntity (an abstract class, i.e. a class that will never be instantiated) and hence they can acquire contents from the conditional packages (_Pacs). The conditional packages provide all key topology properties.

5.1. Forwarding Entity

As noted in the previous section the two key topology classes are Forwarding Domain (FD) and Link (L).

The FD topological component is used to show the potential to enable forwarding. At the lowest level of recursion, an FD (within a network element (NE)) represents a switch matrix (e.g., a fabric). Note that an NE can encompass multiple switch matrices (FDs).

As noted earlier the Link models adjacency between two or more Forwarding Domains (FD).

Both the link and the FD have the potential to handle more than one layerProtocol (both have a layerProtocolNameList attribute).

As shown in Figure 5-1 an object class "ForwardingEntity" has been defined to collect topology-related properties (characteristics etc.) that are common for FD and Link.

A ForwardingEntity is an abstract representation of the emergent effect of the combined functioning of an arrangement of components (running hardware, software running on hardware, etc). The effect can be considered as the realization of the potential for apparent communication adjacency for entities that are bound to the terminations at the boundary of the ForwardingEntity.

The ForwardingEntity enables the creation of constrained forwarding to achieve the apparent adjacency. The apparent adjacency has intended performance degraded from perfect adjacency and a statement of that degradation is conveyed via the attributes of the packages associated with this class. In the model both ForwardingDomain and Link are ForwardingEntities.

This abstract class is used as a modeling approach to apply packages of attributes to both Link and ForwardingDomain. Link and ForwardingDomain are the key ForwardingEntities.

5.2. Characteristics of Topological Entity

As noted above the characteristic of a TopologicalEnity are covered by the conditional packages (_PACs).

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I           Conditional Packages of Topological Entity             I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
```

Figure 5-2 Conditional Packages of Topological Entity

5.2.1. Risk (RiskParameter_Pac)

The risk characteristics of a ForwardingEntity come directly from the underlying physical realization.

The risk characteristics propagate from the physical realization to the client and from the server layer to the client layer, this propagation may be modified by protection.

A ForwardingEntity may suffer degradation or failure as a result of a problem in a part of the underlying realization.

The realization can be partitioned into segments which have some relevant common failure modes.

There is a risk of failure/degradation of each segment of the underlying realization.

Each segment is a part of a larger physical/geographical unit that behaves as one with respect to failure (i.e. a failure will have a high probability of impacting the whole unit (e.g. all fibers in the same cable).

Disruptions to that larger physical/geographical unit will impact (cause failure/errors to) all ForwardingEntities that use any part of that larger physical/geographical entity.

Any ForwardingEntity that uses any part of that larger physical/geographical unit will suffer impact and hence each ForwardingEntity shares risk.

The identifier of each physical/geographical unit that is involved in the realization of each segment of a Topological entity can be listed in the RiskParameter_Pac of that ForwardingEntity.

A segment has one or more risk characteristic.

Shared risk between two ForwardingEntities compromises the integrity of any solution that use one of those ForwardingEntity as a backup for the other.

Where two ForwardingEntities have a common risk characteristic they have an elevated probability of failing simultaneously compared to two ForwardingEntities that do not share risk characteristics.

- riskCharacteristicList: A list of risk characteristics (RiskCharacteristic) for consideration in an analysis of shared risk. Each element of the list represents a specific risk consideration.

- RiskCharacteristic: The information for a particular risk characteristic where there is a list of risk identifiers related to that characteristic. It includes:

    o riskCharacteristicName: The name of the risk characteristic. The characteristic may be related to a specific degree of closeness. For example a particular characteristic may apply to failures that are localized (e.g. to one side of a road) where as another characteristic may relate to failures that have a broader impact (e.g. both sides of a road that crosses a bridge). Depending upon the importance of the traffic being routed different risk characteristics will be evaluated.

    o riskIdentifierList: A list of the identifiers of each physical/geographic unit (with the specific risk characteristic) that is related to a segment of the ForwardingEntity.

5.2.2. TransferCost_Pac

The cost characteristics of a ForwardingEntity not necessarily correlated to the cost of the underlying physical realization.

They may be quite specific to the individual ForwardingEntity e.g. opportunity cost. Relates to layer capacity

There may be many perspectives from which cost may be considered for a particular ForwardingEntity and hence many specifc costs and potentially cost algorithms.

Using an entity will incur a cost.

- costCharcteristicList: The list of costs (CostCharacteristic) where each cost relates to some aspect of the Link

    o CostCharcteristic: The information for a particular cost characteristic

        . costName: The cost characteristic will related to some aspect of the ForwardingEntity (e.g. $ cost, routing weight). This aspect will be conveyed by the costName

        . costValue: The specific cost.

        . costAlgorithm: The cost may vary based upon some properties of the ForwardingEntity. The rules for the variation are conveyed by the costAlgorithm.

5.2.3. TransferTiming_Pac

A link will suffer effects from the underlying physical realization related to the timing of the information passed by the link.

- fixedLatencyCharacteristic: A ForwardingEntity suffers delay caused by the realization of the servers (e.g. distance related; FEC encoding etc.) along with some client specific processing. This is the total average latency effect of the ForwardingEntity

- jitterCharacteristic: High frequency deviation from true periodicity of a signal and therefore a small high rate of change of transfer latency. Applies to TDM systems (i.e., not packet based systems).

- wanderCharacteristics: Low frequency deviation from true periodicity of a signal and therefore a small low rate of change of transfer latency. Applies to TDM systems (i.e., not packet based systems).

- queuingLatencyList: The effect on the latency of a queuing process. This only has significant effect for packet based systems and has a complex characteristic (QueuingLatency).

  o QueuingLatency: Provides information on latency characteristic for a particular stated trafficProperty.

5.2.4. TransferIntegrity_Pac

Transfer integrity characteristic covers expected (specified) error, loss and duplication signal content as well as any damage of any form to total link and to the client signals.

- errorCharacteristic: describes the degree to which the signal propagated can be errored. Applies to TDM systems as the errored signal will be propagated and not packet as errored packets will be discarded.

- lossCharacteristic: Describes the acceptable characteristic of lost packets where loss may result from discard due to errors or overflow. Applies to packet systems and not TDM (as for TDM errored signals are propagated unless grossly errored and overflow/underflow turns into timing slips).

- repeatDeliveryCharacteristic: Primarily applies to packet systems where a packet may be delivered more than once (in fault recovery for example). It can also apply to TDM where several frames may be received twice due to switching in a system with a large differential propagation delay.

- deliveryOrderCharacteristic: Describes the degree to which packets will be delivered out of sequence. Does not apply to TDM as the TDM protocols maintain strict order.

- unavailableTimeCharacteristic: Describes the duration for which there may be no valid signal propagated.

- serverIntegrityProcessCharacteristic: Describes the effect of any server integrity enhancement process on the characteristics of the ForwardingEntity.

5.2.5. TransferCapcity_Pac

The ForwardingEntity derives capacity from the underlying realization.

A ForwardingEntity may be an abstraction and virtualization of a subset of the underlying capability offered in a view or may be directly reflecting the underlying realization.

A ForwardingEntity may be directly used in the view or may be assigned to another view for use.

The clients supported by a multi-layer ForwardingEntity may interact such that the resources used by one client may impact those available to another. This is derived from the LTP spec details.

A ForwardingEntity represents the capacity available to user (client) along with client interaction and usage.

A ForwardingEntity may reflect one or more client protocols and one or more members for each profile.

- totalPotentialCapacity: A "best case" view of the capacity of the ForwardingEntity assuming that any shared capacity is available to be taken.

Note that this area is still under development to cover concepts such as:

- exclusiveCapacityList: The capacity allocated to this ForwardingEntity for its exclusive use

- sharedCapacityList: The capacity allocated to this ForwardingEntity that is not exclusively available as it is shared with others.

- assignedAsExclusiveCapacityList: The capacity assigned from this TopologicalEnity to another ForwardingEntity for its exclusive use

- assignedAsSharedCapacityList: The capacity assigned to one or more other ForwardingEntities for shared use where the interaction follows some stated algorithm.

- Capacity which includes:

    - o totalSize

    - o numberOfUsageInstances

    - o maximumUsageSize

    - o numberingRange

5.2.6. Validation_Pac

Validation covers the various adjacenct discovery and reachability verification protocols. Also may cover Information source and degree of integrity.

- validationMechanismList: Provides details of the specific validation mechanism(s) used to confirm the presence of an intended ForwardingEntity.

5.2.7. LayerProtocolTransition_Pac

Relevant for a Link that is formed by abstracting one or more LTPs (in a stack) to focus on the flow and deemphasize the protocol transformation.

This abstraction is relevant when considering multi-layer routing.

The layer protocols of the LTP and the order of their application to the signal is still relevant and need to be accounted for. This is derived from the LTP spec details.

This Pac provides the relevant abstractions of the LTPs and provides the necessary association to the LTPs involved.

Links that included details in this Pac are often referred to as Transitional Links.

- transitionedLayerProtocolList: Provides the ordered structure of layer protocol transitions encapsulated in the ForwardingEntity. The ordering relates to the LinkEnd role.

6. Purpose Specific IM Example - Transport API Topology Service

In order to provide some further clarity, this section provides a high level introduction to a Purpose Specific IM, the Transport API (T-API) Topology service, which has been derived from the ONF Common Information Model (ONF-CIM) according to the principles in [I-D.betts].

The context of the T-API refers to the scope and control and naming that a particular SDN controller, manager or a client application has with respect to the information it operates on internally or exchanges over an interface. The following sections further describe this purpose specific IM and relationship to the ONF-CIM.

6.1. T-API IM Constructs

The T-API IM uses terminology that is considered to be more familiar to the transport network management community and maps to the constructs defined in the ONF-CIM CNM Topology model. The following table provides a high level summary of the mapping of the constructs relevant to the T-API Topology Service.

Mapping of CIM and T-API IM Constructs

ONF-CIM CNM Terminology   T-API IM Terminology

NetworkControlDomain   Context

Topology   ForwardingDomain

(FD)   Node

Link   Link

TransitionalLink

NodeEdgePoint   LogicalTermi

nationPoint (LTP) ServideEndPoint

The following provides a brief description of these T-API IM constructs.

- o Link: A Link is an abstract representation of the effective adjacency between two or more associated Nodes in a Topology. It is terminated by Node-Edge-Points of the associated Nodes.

- o Node: A Node is an abstract representation of the forwarding-capabilities of a particular set of Network Resources. It is described in terms of an aggregation of set of ports (Node-Edge-Point) belonging to those Network Resources and the potential to enable forwarding of information between those edge ports.

- o Node-Edge-Point: A Node-Edge-Point represents the inward network-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides an encapsulation of addressing, mapping, termination, adaptation and OAM functions of one or more transport layers (including circuit and packet forms) performed at the entry and exit points of the Node.

- o Topology: A Topology is an abstract representation of the topological-aspects of a particular set of Network Resources. It is described in terms of a network of set of Nodes and Links that enable the forwarding-capabilities of that particular set of Network Resources.

- o Service-End-Point: A Service-End-Point represents the outward customer-facing aspects of the edge-port functions that access the forwarding capabilities provided by the Node. Hence it provides a limited, simplified view of interest to external clients (e.g. shared addressing, capacity, resource availability, etc) that enable the clients to request connectivity without the need to understand the provider network internals.

- o Transitional Link: A topological component that consists of the link port at the edge of one node and a corresponding link port at the edge of another node that operates on different layers or whose layer is the same but with different Layer

Information. A transitional link is supported/implemented by transport processing functions (e.g., adaptation/termination). A transitional link can be partitioned into parallel transitional links, or a concatenation of transitional links. It can also be partitioned into a concatenation of transitional links and zero or more links.

6.2. T-API Topology Service IM

The resultant high-level description for the T-API Topology Service constructs, based upon the pruned and refactored ONF-CIM, and the related Topology Service APIs are provided in Figure 6-1 below.

```
--------------------------------------------------------------------
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                    Topology Service Skeleton                       I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                     (only in PDF version)                        I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
I                                                                  I
--------------------------------------------------------------------
```

Figure 6-1 Topology Service Skeleton

The T-API Topology Service API enables the API client to, for example, retrieve Topology, Node, Link, and Edge-Point details.

- o Topology details: returns attributes of the Topology identified by the provided input ID. This includes references to lower-level Nodes and Links encompassed by that Topology. A NULL input value is expected to return the top-most Topology that corresponds to the scope of the entire Context including any Off-Network-Links.

- o Node details: Returns attributes of the Node identified by the provided input ID. Includes references to Node-Edge-Points aggregated by the Node, and attributes representing the identification, naming, states and forwarding capabilities of the Node.

- o Link details: Returns attributes of the Link identified by the provided input ID. Includes references to Node-Edge-Points terminating the Link, and references to the Nodes associated by the Link.

- o Node-Edge-Point details: Returns attributes of the Node-Edge-Point identified by the provided input ID, including references to Service-End-Points mapped to this Node-Edge-Point.

The API supports a retrieve-scope filter: LayerProtocol list. If set, the API call will return output that is relevant to the specified Layer only.

7. Usage of the IM Topology Subset regarding TE Topology DM

As discussed earlier, a data model (DM) may be derived from an IM. Examples of YANG DMs derived according to automated translation tools based upon mapping guidelines are provided in [OSSDN SNOMASS] at https://github.com/OpenNetworkingFoundation/Snowmass-ONFOpenTransport/tree/develop/YANG. It is possible to leverage the IM

Topology Subset to assess the consistency and completeness of related YANG modules under development.

8. Security Considerations

This informational document is intended only to provide a description of an interface-protocol-neutral information model, and the security concerns are therefore out of the scope of this document.

9. IANA Considerations

This document includes no request to IANA.

10. Conclusions

The information modeling described in this draft, which is relevant to Network Topology [ONF TR-512] [OSSDN SNOWMASS], can be leveraged in assessing the consistency and completeness of related YANG modules under development.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.betts] Betts, M., Davis, N., Lam, K., Zeuner, B., Mansfield, S. and P. Doolan, "Framework for Deriving Interface Data Schema from UML Information Models", draft-betts-netmod-framework-data-schema-uml-04 (work in progress), October 2016

[I-D.mansfield] Mansfield, S., Zeuner, B., Davis, N., Yun, X., Tochio, Y., Lam, K. and E. Varma, "Guidelines for Translation of UML Information Model to YANG Data Model", draft-mansfield-netmod-uml-to-yang-03 (work in progress), October 2016

[ONF TR-512] ONF TR-512 "ONF-CIM Core Model base document 1.2" (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-512_CIM_(CoreModel)_1.2.zip), September 2016

[ONF TR-513] ONF TR-513 " Common Information Model Overview 1.2" (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-513_CIM_Overview_1.2.pdf), September 2016

[ONF TR-514] ONF TR-514 "UML Modeling Guidelines 1.2" (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-514_UML_Modeling_Guidelines_v1.2.pdf), September 2016

[ONF TR-515] ONF TR-515 "Papyrus Guidelines 1.2" (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-515_Papyrus_Guidelines_v1.2.pdf), September 2016

[OSSDN SNOWMASS] Open Source SDN SNOWMASS/Open Transport API Specifications (https://github.com/OpenNetworkingFoundation/Snowmass-ONFOpenTransport)

[G.7711] Recommendation ITU-T G.7711/Y.17022 "Generic protocol-neutral information model for transport resources", September 2016

[G.874.1] Recommendation ITU-T G.874.1 "Optical transport network: Protocol-neutral management information model for the network element view", September 2016

[G.8052] Recommendation ITU-T G.8052/Y.1346 "Protocol-neutral management information model for the Ethernet Transport capable network element", September 2016

[G.8152] Recommendation ITU-T G.8152/Y.1375 "Protocol-neutral management information model for the MPLS-TP network element", September 2016

[G.852.2] Recommendation ITU-T G.852.2 "Enterprise viewpoint description of transport network resource model", March 1999

[TMF612] TM Forum 612 "MTOSI Information Agreement", October 2014

12. Contributors

Karthik Sethuraman
NEC

Email: karthik.sethuraman@necam.com

13. Acknowledgments

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

Authors' Addresses

Kam Lam
Alcatel-Lucent, USA

Phone: +1 732 331 3476
Email: kam.lam@alcatel-lucent.com

Eve Varma
Alcatel-Lucent, USA

Email: eve.varma@alcatel-lucent.com

Paul Doolan
Coriant, Germany

Phone: +1 972 357 5822
Email: paul.doolan@coriant.com

Malcolm Betts
ZTE, China

Phone: +1 678 534 2542
Email: malcolm.betts@zte.com.cn

Nigel Davis
Ciena, UK

Email: ndavis@ciena.com

Bernd Zeuner
Deutsche Telekom, Germany

Phone: +49 6151 58 12086
Email: b.zeuner@telekom.de

Italo Busi
Huawei, China

Email: Italo.Busi@huawei.com

Scott Mansfield
Ericsson, Sweden

Phone: 1 724 931 9316
Email: scott.mansfield@ericsson.com

Yuji Tochio
Fujitsu, Japan

Phone: 81 44 754 8829
Email: tochio@jp.fujitsu.com

Ricard Vilalta
CTTC, Spain

Phone:
Email: ricard.vilalta@cttc.es

Victor Lopez
Telefonica, Spain

Phone:
Email: victor.lopezalvarez@telefonica.com

Teas Working Group
Internet Draft

Intended status: Informational

Expires: April 2017

Young Lee
Huawei

Sergio Belotti
Nokia

Dhruv Dhody
Huawei

Daniele Ceccarelli
Ericsson

Bin Young Yun
ETRI

October 24, 2016

Information Model for Abstraction and Control of TE Networks (ACTN)

draft-leebelotti-teas-actn-info-05.txt

Abstract

This draft provides an information model for Abstraction and Control of Traffic Engineered (TE) networks (ACTN).

Status of this Memo

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

This Internet-Draft will expire on April 24, 2015.

Copyright Notice

Table of Contents

1. Introduction

This draft provides an information model for the requirements identified in the ACTN requirements [ACTN-Req] and the ACTN interfaces identified in the ACTN architecture and framework document [ACTN-Frame].

The purpose of this draft is to put all information elements of ACTN in one place before proceeding to development work necessary for protocol extensions and data models.

The ACTN reference architecture identified a three-tier control hierarchy as depicted in Figure 1:

- Customer Network Controllers (CNC)
- Multi-Domain Service Coordinator (MDSC)
- Physical Network Controllers (PNC).

```
   +-------+                 +-------+                  +-------+
   | CNC-A |                 | CNC-B |                  | CNC-C |
   +-------+                 +-------+                  +-------+
         \____________           |            _____________ _/
          ----------             | CMI         ------------
                     \           |            /
                      +------------------------+
                      |         MDSC           |
                      +------------------------+
            _________/           |            \__________
            --------             | MPI          ------------____
           /                     |                          \
   +-------+                 +-------+                  +-------+
   |  PNC  |                 |  PNC  |                  |  PNC  |
   +-------+                 +-------+                  +-------+
```

Figure 1: A Three-tier ACTN control hierarchy

The two interfaces with respect to the MDSC, one north of the MDSC and the other south of the MDSC are referred to as CMI (CNC-MDSC Interface) and MPI (MDSC-PNC Interface), respectively. It is intended to model these two interfaces and derivative interfaces thereof (e.g., MDSC to MSDC in a hierarchy of MDSCs) with one common model.

Appendix A provides some relevant ACTN use-cases extracted from [ACTN-Req]. Appendix A is information only and may help readers understand the context of key use-cases addressed in [ACTN-Req].

1.1. Terminology

- o A Virtual Network is a client view (typically a network slice) of the transport network. It is presented by the provider as a set of physical and/or abstracted resources. Depending on the agreement between client and provider various VN operations and VN views are possible. There are three aspects related to VN:

  1) VN Creation: VN could be pre-configured and created via static negotiation between customer and provider. In other cases, VN could also be created dynamically based

on the request from the customer with given SLA attributes which satisfy the customer's objectives.

2) Dynamic Operations: VN could be further modified and deleted based on customer request to request changes in the network resources reserved for the customer. The customer can further act upon the virtual network resources to perform E2E tunnel management (set-up/release/modify). These changes will incur subsequent LSP management on the operator's level.

3) VN View: (a) VN can be seen as an (or set of) e2e tunnel(s) from a customer point of view where an e2e tunnel is referred as a VN member. Each VN member (i.e., e2e tunnel) can then be formed by recursive aggregation of lower level paths at a provider level. Such end to end tunnels may comprise of customer end points, access links, intra domain paths and inter-domain link. In this view VN is thus a list of VN members. (b) VN can also be seen as a terms of topology comprising of physical and abstracted nodes and links. The nodes in this case include physical customer end points, border nodes, and internal nodes as well as abstracted nodes. Similarly the links includes physical access, inter-domain and intra-domain links as well as abstracted links. The abstracted nodes and links in this view can be pre-negotiated or created dynamically.

o A Virtual Network Service (VNS) is the creation and offering of a Virtual Network by a provider to a customer in accordance with SLA agreements reached between them (e.g., re satisfying the customer's objectives).

o 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 it 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 [RFC7926].

o Abstract topology: Every lower controller in the provider network, when is representing its network topology to a higher layer, it may want to selective hide details of the actual network topology, as suggested for abstraction in [RFC7926]. In such case, an abstract topology may be used for this purpose.

Abstract topology enhances scalability for the MDSC to operate multi-domain networks.

2. ACTN Common Interfaces Information Model

This section provides ACTN common interface information model to describe in terms of primitives, objects, their properties (represented as attributes), their relationships, and the resources for the service applications needed in the ACTN context.

Basic primitives (messages) are required between the CNC-MDSC and MDSC-PNC controllers. These primitives can then be used to support different ACTN network control functions like network topology request/query, VN service request, path computation and connection control, VN service policy negotiation, enforcement, routing options, etc.

The standard interface is described between a client controller and a server controller. A client-server relationship is recursive between a CNC and a MDSC and between a MDSC and a PNC. In the CMI, the client is a CNC while the server is a MDSC. In the MPI, the client is a MDSC and the server is a PNC. There may also be MDSC-MDSC interface(s) that need to be supported. This may arise in a hierarchy of MDSCs in which workloads may need to be partitioned to multiple MDSCs.

Basic primitives (messages) are required between the CNC-MDSC and MDSC-PNC controllers. These primitives can then be used to support different ACTN network control functions like network topology request/query, VN service request, path computation and connection control, VN service policy negotiation, enforcement, routing options, etc.

At a minimum, the following VN action primitives should be supported:

- VN Instantiate (See Section 2.1.1. for the description)

- VN Modify (See Section 2.1.2. for the description)

- VN Delete (See Section 2.1.3. for the description)

- VN Update ((See Section 2.1.4. for the description)

- VN Path Compute (See Section 2.1.5. for the description)

- VN Query (See Section 2.1.6. for the description)

In addition to VN action primitives, TE Update primitive should also be supported (See Section 2.1.7. for the description).

2.1. VN Action Primitives

This section provides a list of main primitives necessary to satisfy ACTN requirements specified in [ACTN-REQ].

<VN Action> describes main primitives. VN Action can be one of the following primitives: (i) VN Instantiate; (ii) VN Modify; (iii) VN Delete; (iv) VN Update; (v) VN Path Compute; (vi) VN Query.

```
<VN Action> ::= <VN Instantiate> |

                <VN Modify> |

                <VN Delete> |

                <VN Update> |

                <VN Path Compute> |

                <VN Query>
```

2.1.1. VN Instantiate

<VN Instantiate> refers to an action from customers/applications to request their VNs. This primitive can also be applied from an MDSC to a PNC requesting a VN (if the domain the PNC supports can instantiate the entire VN) or a part of VN elements. Please see the definition of VN in the section 2.

2.1.2. VN Modify

<VN Modify> refers to an action from customers/applications to modify an existing VN (i.e., instantiated VN). This primitive can also be applied from an MDSC to a PNC requesting a VN (if the domain the PNC supports can instantiate the entire VN) or a part of VN elements.

2.1.3. VN Delete

<VN Delete> refers to an action from customers/applications to delete an existing VN. This primitive can also be applied from an MDSC to a PNC requesting a VN (if the domain the PNC supports can instantiate the entire VN) or a part of VN elements.

2.1.4. VN Update

<VN Update> refers to any update to the VN that need to be updated to the subscribers. VN Update fulfills a push model at CMI level, to make aware customers of any specific changes in the topology details related to VN instantiated.

Note the VN Update means the connection-related information (e.g., LSPs) update that has association with VNs.

2.1.5. VN Path Compute

<VN Path Compute> consists of Request and Reply. Request refers to an action from customers/applications to request a VN path computation. This primitive can also be applied from an MDSC to a PNC requesting a VN (if the domain the PNC supports can instantiate the entire VN) or a part of VN elements.

<VN Path Compute> Reply refers to the reply in response to <VN Path Compute> Request.

<VN Path Compute> Request/Reply is to be differentiated from a VN Instantiate. The purpose of VN Path Compute is a priori exploration to estimate network resources availability and getting a list of possible paths matching customer/applications constraints. To make this type of request Customer/application controller can have a shared (with lower controller) view of an abstract network topology on which to get the constraints used as input in a Path Computation request. The list of paths obtained by the request can be used by customer/applications to give path constrains during VNS connectivity request and to compel the lower level controller (e.g. MDSC) to select the path that Client/application controller has chosen among the set of paths returned by the Path Computation primitives. The importance of this primitives is for example in a scenario like multi-domain in which the optimal path obtained by an orchestrator as sum of optimal paths for different domain controller

cannot be the optimal path in the Client/application controller prospective. This only applies between CNC and MDSC.

2.1.6. VN Query

<VN Query> refers to any query pertaining to the VN that has been already instantiated. VN Query fulfills a pull model and permit to get topology view.

<VN Query Reply> refers to the reply in response to <VN Query>.

2.1.7. TE Update (for TE resources)

<TE Update> it is a primitives specifically related to MPI interface to provide TE resource update between any domain controller towards MDSC regarding the entire content of any "domain controller" TE topology or an abstracted filtered view of TE topology depending on negotiated policy.

<TE Update> ::= [<Abstraction>]<TE-topology...>

<TE-topology> ::= <TE-Topology-reference> <Node-list> <Link-list>

<Node-list> ::= <Node>[<Node-list>]

<Node> ::= <Node> <TE-Termination Points>

<Link-list> ::= <Link>[<Link-list>]

Where

<Abstraction> provides information on level of abstraction (as determined a priori).

<TE-topology-reference> ::= information related to the specific te-topology related to nodes and links present in this TE-topology.

<Node-list> ::= detailed information related to a specific node belonging to a te-topology e.g. te-node-attributes [TE-TOPO].

<Link-list> ::= information related to the specific link related belonging to a te-topology e.g. te-link-attributes [TE-TOPO].

<TE-Termination Points> ::= information details associated to the termination point of te-link related to a specific node e.g. interface-switching-capability [TE-TOPO].

2.2. VN Objects

This section provides a list of objects associated to VN action primitives.

2.2.1. VN Identifier

<VN Identifier> is a unique identifier of the VN.

2.2.2. VN Service Characteristics

VN Service Characteristics describes the customer/application requirements against the VNs to be instantiated.

```
<VN Service Characteristics> ::= <VN Connectivity Type>

                                 (<VN Traffic Matrix>...)

                                 <VN Survivability>
```

Where

<VN Connectivity Type> ::= <P2P>|<P2MP>|<MP2MP>|<MP2P>|<Multi-destination>

The Connectivity Type identifies the type of required VN Service. In addition to the classical type of services (e.g. P2P/P2MP etc.), ACTN defines the "multi-destination" service that is a new P2P service where the end points are not fixed. They can be chosen among a list of pre-configured end points or dynamically provided by the CNC.

```
<VN Traffic Matrix> ::= <Bandwidth>

                        [<VN Constraints>]
```

The VN Traffic Matrix represents the traffic matrix parameters required against the service connectivity required and so the VN request instantiation between service related Access Points [ACTN-Frame]. Bandwidth is a mandatory parameter and a number of optional constrains can be specified in the <VN Constrains> (e.g. diversity, cost). They can include objective functions and TE metrics bounds as specified in [RFC5441].

Further details on the VN constraints are specified below:

```
<VN Constraints> ::= [<Layer Protocol>]

                     [<Diversity>]

                     [<Shared Risk>]

                     <Metric>
```

Where:

<Layer Protocol> Identifies the layer at which the VN service is requested. It could be for example MPLS, ODU, and OCh.

<Diversity> This allows asking for diversity constraints for a VN Instantiate/Modify or a VN Path Compute. For example, a new VN or a path is requested in total diversity from an existing one (e.g. diversity exclusion).

```
<Diversity> ::= <VN-exclusion> (<VN-id>...) |

                <VN-E2E Tunnel-exclusion> (<Tunnel-id>...)
```

<Shared Risk> Based on the realization of VN required, group of physical resources can be impacted by the same risk. An E2E tunnel can be impacted by this shared risk. This is used to get the SRLG associated with the different tunnels composing a VN.

<Metric> can include all the Metrics (cost, delay, delay variation, latency), bandwidth utilization parameters defined and referenced by [RFC3630] and [RFC7471].

<VN Survivability> describes all attributes related to the VN recovery level and its survivability policy enforced by the customers/applications.

```
<VN Survivability> ::= <VN Recovery Level>

                     [<VN Tunnel Recovery Level>]

                     [<VN Survivability Policy>]
```

Where:

<VN Recovery Level> It is a value representing the requested level of resiliency required against the VN. The following values are defined:

. Unprotected VN
. VN with per tunnel recovery: The recovery level is defined against the tunnels composing the VN and it is specified in the <VN Tunnel Recovery Level>.

<VN Tunnel Recovery Level> ::= <0:1>|<1+1>|<1:1>|<1:N>|<M:N>|

<On the fly restoration>

The VN Tunnel Recovery Level indicates the type of protection or restoration mechanism applied to the VN. It augments the recovery types defined in [RFC4427].

<VN Survivability Policy> ::= [<Local Reroute Allowed>]

[<Domain Preference>]

[<Push Allowed>]

[<Incremental Update>]

Where:

<Local Reroute Allowed> is a delegation policy to the Server to allow or not a local reroute fix upon a failure of the primary LSP.

<Domain Preference> is only applied on the MPI where the MDSC (client) provides a domain preference to each PNC (server).e.g. when a inter-domain link fails, then PNC can choose the alternative peering with this info.

<Push Allowed> is a policy that allows a server to trigger an updated VN topology upon failure without an explicit request from the client. Push action can be set as default unless otherwise specified.

<Incremental Update> is another policy that triggers an incremental update from the server since the last period of

update. Incremental update can be set as default unless otherwise specified.

2.2.3. VN End-Point

<VN End-Point> Object describes the VN's customer end-point characteristics.

<VN End-Point> ::= (<Access Point Identifier>

[<Access Link Capability>]

[<Source Indicator>])...

Where:

<Access point identifier> It represents a unique identifier of the client end-point. They are used by the customer to ask for the setup of a virtual network creation. A <VN End-Point> is defined against each AP in the network and is shared between customer and provider. Both the customer and the provider will map it against his own physical resources.

<Access Link Capability> An optional object that identifies the capabilities of the access link related to the given access point. (e.g., max-bandwidth, bandwidth availability, etc.)

<Source Indicator> indicates if an End-point is source or not.

2.2.4. VN Objective Function

The VN Objective Function applies to each VN member (i.e., each E2E tunnel) of a VN.

The VN Objective Function can reuse objective functions defined in [RFC5541] section 4.

For a single path computation, the following objective functions are defined:

o MCP is the Minimum Cost Path with respect to a specific metric (e.g. shortest path).

- o MLP is the Minimum Load Path, that means find a path composted by te-link least loaded.
- o MBP is the Maximum residual Bandwidth Path.

For a concurrent path computation, the following objective functions are defined:

- o MBC is to Minimize aggregate Bandwidth Consumption.
- o MLL is to Minimize the Load of the most loaded Link.
- o MCC is to Minimize the Cumulative Cost of a set of paths.

2.2.5. VN Action Status

<VN Action Status> is the status indicator whether the VN has been successfully instantiated, modified, or deleted in the server network or not in response to a particular VN action.

Note that this action status object can be implicitly indicated and thus not included in any of the VN primitives discussed in Section 2.3.

2.2.6. VN Associated LSP

<VN Associated LSP> describes the instantiated LSPs that is associated with the VN. <VN Associated LSP> is used between each domain PNC and the MDSC as part of VN Update once the VN is instantiated in each domain network and when CNC want to have more details about the topology instantiated as consequence of a VN Instantiate.

<VN Associated LSP> ::= <VN Identifier> (<LSP>...)

2.2.7. VN Computed Path

The VN Computed Path is the list of paths obtained after the VN path computation request from higher controller. Note that the computed path is to be distinguished from the LSP. When the computed path is signaled in the network (and thus the resource is reserved for that path), it becomes an LSP.

<VN Computed Path> ::= (<Path>...)

2.2.8. VN Service Preference

This section provides VN Service preference. VN Service is defined in Section 2.

```
<VN Service Preference> ::= [<Location Service Preference >]

                            [<Client-specific Preference >]

                            [<End-Point Dynamic Selection Preference >]
```

Where

<Location Service Preference describes the End-Point Location’s (e.g. Data Centers) support for certain Virtual Network Functions (VNFs) (e.g., security function, firewall capability, etc.)and is used to find the path that satisfies the VNF constraint.

<Client-specific Preference> describes any preference related to Virtual Network Service (VNS) that application/client can enforce via CNC towards lower level controllers. For example, permission the correct selection from the network of the destination related to the indicated VNF It is e.g. the case of VM migration among data center and CNC can enforce specific policy that can permit MDSC/PNC to calculate the correct path for the connectivity supporting the data center interconnection required by application.

<End-Point Dynamic Selection Preference> describes if the End-Point (e.g. Data Center) can support load balancing, disaster recovery or VM migration and so can be part of the selection by MDSC following service Preference enforcement by CNC.

2.3. Mapping of VN Primitives with VN Objects

This section describes the mapping of VN Primitives with VN Objects based on Section 2.2.

```
<VN Instantiate> ::= <VN Service Characteristics>

                     <VN Objective Function>

                     <VN End-Point>
```

```
                        [<VN Service Preference>]

   <VN Modify> ::= <VN identifier>
                    <VN Service Characteristics>
                   [<VN Objective Function>]
                    <VN End-Point>
                   [<VN Service Preference>]

   <VN Delete> ::= <VN Identifier>

   <VN Update> :: = <VN Identifier>
                     <VN Associated LSP>

   <VN Path Compute Request> ::= <VN Service Characteristic>
                           <VN Objective Function>
                           <VN End-Point>


   <VN Path Compute Reply> ::= <VN Computed Path>

   <VN Query> ::= <VN Identifier>

   <VN Query Reply> ::= <VN Identifier>
                          <VN Associated LSP>
```

3. References

3.1. Normative References

[DRAFT-SER-AWARE] Dhruv Dhody, Qin Wu, Vishwas Manral, Zafar Ali, and Kenji Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP).," June 2016, draft-ietf-pce-pcep-service-aware-10.

3.2. Informative References

[TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.Informative References

[ACTN-Req] Y. Lee, et al., "Requirements for Abstraction and Control of Transport Networks", draft-lee-teas-actn-requirements, work in progress.

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

[Stateful-PCE] E. Crabbe, et al., "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce, work in progress.

[RFC5541] JL. Le Roux, JP. Vasseur and Y. Lee, "Encoding of Objective Functio
ns
in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

[RFC7926] A.Farrel, et al., "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.

4. Contributors

Contributors' Addresses

Authors' Addresses

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

Sergio Belotti (Editor)
Alcatel Lucent
Via Trento, 30
Vimercate, Italy
Email: sergio.belotti@alcatel-lucent.com

Dhruv Dhody
Huawei Technologies,
Divyashree Technopark, Whitefield
Bangalore, India
Email: dhruv.ietf@gmail.com

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

Bin Young Yun
ETRI
Email: byyun@etri.re.kr

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

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

Appendix A: ACTN Applications

A.1. Coordination of Multi-destination Service Requirement/Policy

```
                             +----------------+
                             |       CNC      |
                             |   (Global DC   |
                             |   Operation    |
                             |    Control)    |
                             +--------+-------+
                                      | |  Service Requirement/Policy:
                                      | |  - Endpoint/DC location info
                                      | |  - Endpoint/DC dynamic
                                      | |    selection policy
                                      | |    (for VM migration, DR, LB)
                                      | v
                            +---------+---------+
                            |  Multi-domain     | Service policy-driven
                            |Service Coordinator| dynamic DC selection
                            +-----+---+---+-----+
                                  |   |   |
                                  |   |   |
                 +----------------+   |   +----------------+
                 |                    |                    |
           +-----+-----+       +-----+------+      +------+-----+
           |   PNC for |       |  PNC for   |      |  PNC for   |
           | Transport |       | Transport  |      | Transport  |
           | Network A |       | Network B  |      | network C  |
           +-----------+       +------------+      +------------+
                 |                    |                   |
+---+      ------               ------              ------       +---+
|DC1|--////      \\\\       ////      \\\\      ////      \\\\---+DC5|
+---+ |              |     |              |    |              |  +---+
      |     TN A     +-----+     TN B     +----+      TN C    |
      /              |     |              |    |              |
     / \\\\      ////     / \\\\      ////      \\\\      ////
   +---+   ------        /      ------    \         ------ \
   |DC2|                /                  \                \+---+
   +---+               /                    \                |DC6|
                     +---+                   \ +---+         +---+
                     |DC3|                    \|DC4|
                     +---+                     +---+

                                                DR: Disaster Recovery
                                                LB: Load Balancing
            Figure A.1: Service Policy-driven Data Center Selection
```

Figure A.1 shows how VN service policies from the CNC 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 service policy plays an important role for virtual network operation. Service policy can be static or dynamic. Dynamic service policy for data center selection may be placed as a result of utilization of data center resources supporting VNs. The MDSC would then incorporate this information to meet the service objective of this application.

A.2. Application Service Policy-aware Network Operation

```
                           +----------------+
                           |       CNC      |
                           |   (Global DC   |
                           |   Operation    |
                           |    Control)    |
                           +--------+-------+
                                    | | Application Service Policy
                                    | | - VNF requirement (e.g.
                                    | |   security function, etc.)
                                    | | - Location profile for each VNF
                                    | v
                          +---------+---------+
                          |  Multi-domain     |  Dynamically select the
                          |Service Coordinator|  network destination to
                          +-----+---+---+-----+  meet VNF requirement.
                                |   |   |
                                |   |   |
                +---------------+   |   +-----------------+
                |                   |                     |
         +------+-----+       +-----+------+       +------+-----+
         |   PNC for  |       |  PNC for   |       |  PNC for   |
         | Transport  |       | Transport  |       | Transport  |
         | Network A  |       | Network B  |       | network C  |
         |            |       |            |       |            |
         +------------+       +------------+       +------------+
                |                   |                     |
{VNF b}         |                   |                     |  {VNF b,c}
+---+      ------              ------              ------     +---+
|DC1|--////      \\\\      ////      \\\\      ////      \\\\-|DC5|
+---+ |              |     |              |    |              |+---+
     |      TN A      +---+     TN B      +--+      TN C       |
      /              |     |              |    |              |
     / \\\\      ////     / \\\\      ////     \\\\      ////
   +---+   ------        /      ------    \        ------ \
   |DC2|                /                  \               \\+---+
   +---+               /                    \                 |DC6|
    {VNF a}         +---+                    +---+            +---+
                    |DC3|                    |DC4|      {VNF a,b,c}
                    +---+                    +---+
            {VNF a, b}                     {VNF a, c}
```

Figure A.2: Application Service Policy-aware Network Operation

This scenario is similar to the previous case in that the VN service policy for the application can be met by a set of multiple destinations that provide the required virtual network functions (VNF). Virtual network functions can be, for example, security functions required by the VN application. The VN service policy by the CNC would indicate the locations of a certain VNF that can be fulfilled. This policy information is critical in finding the optimal network path subject to this constraint. As VNFs can be dynamically moved across different DCs, this policy should be dynamically enforced from the CNC to the MDSC and the PNCs.

A.3. Network Function Virtualization Service Enabled Connectivity

```
                           +----------------+
                           |       CNC      |
                           |   (Global DC   |
                           |   Operation    |
                           |    Control)    |
                           +--------+-------+
                                    | | Service Policy related to VNF
                                    | | (e.g., firewall, traffic
                                    | | optimizer)
                                    | |
                                    | v
                          +---------+---------+
                          |  Multi-domain     | Select network
                          |Service Coordinator| connectivity subject to
                          +-----+---+---+-----+ meeting service policy
                                |   |   |
                                |   |   |
                +---------------+   |   +----------------+
                |                   |                    |
         +------+-----+       +-----+------+      +------+-----+
         |   PNC for  |       |  PNC for   |      |  PNC for   |
         | Transport  |       | Transport  |      | Transport  |
         | Network A  |       | Network B  |      | network C  |
         |            |       |            |      |            |
         +------------+       +------------+      +------------+
                |                   |                    |
                |                   |                    |
+---+      ------               ------              ------     +---+
|DC1|--////      \\\\       ////      \\\\      ////      \\\\-|DC5|
+---+ |              |     |              |    |              |+---+
     |      TN A      +---+     TN B       +--+      TN C      |
      /              |     |              |    |              |
     / \\\\      ////     / \\\\      ////      \\\\      ////
   +---+   ------        /      ------    \         ------ \
   |DC2|                /                  \                \\+---+
   +---+               /                    \                 |DC6|
                    +---+                    +---+            +---+
                    |DC3|                    |DC4|
                    +---+                    +---+
```

Figure A.3: Network Function Virtualization Service Enabled Connectivity

Network Function Virtualization Services are usually setup between customers’ premises and service provider premises and are provided mostly by cloud providers or content delivery providers. The context may include, but not limited to a security function like firewall, a traffic optimizer, the provisioning of storage or computation capacity where the customer does not care whether the service is implemented in a given data center or another. The customer has to provide (and CNC is providing this)the type of VNF he needs and the policy associated with it (e.g. metric like estimated delay to reach where VNF is located in the DC). The policy linked to VNF is requested as part of the VN instantiation. These services may be hosted virtually by the provider or physically part of the network. This allows the service provider to hide his own resources (both network and data centers) and divert customer requests where most suitable. This is also known as "end points mobility" case and introduces new concepts of traffic and service provisioning and resiliency (e.g., Virtual Machine mobility).

A.4. Dynamic Service Control Policy Enforcement for Performance and Fault Management

```
+------------------------------------------------+
|            Customer Network Controller         |
+------------------------------------------------+
1.Traffic|  /|\4.Traffic           | /|\
Monitor& |   |  Monitor            |  | 8.Traffic
Optimize |   |  Result   5.Service |  | modify &
Policy   |   |             modify& |  | optimize
        \|/  |       optimize Req.\|/ | result
+------------------------------------------------+
|          Multi-domain Service Coordinator      |
+------------------------------------------------+
2. Path  |  /|\3.Traffic           | /|\
Monitor  |   | Monitor             |  |7.Path
Request  |   | Result     6.Path   |  | modify &
         |   |            modify&  |  | optimize
        \|/  |       optimize Req.\|/ | result
+------------------------------------------------+
|           Physical Network Controller          |
+------------------------------------------------+
```

Figure A.4: Dynamic Service Control for Performance and Fault Management

Figure A.4 shows the flow of dynamic service control policy enforcement for performance and fault management initiated by customer per VN. The feedback loop and filtering mechanism tailored for VNs performed by the MDSC differentiates this ACTN scope from traditional network management paradigm. VN level dynamic OAM data model is a building block to support this capability.

A.5. E2E VN Survivability and Multi-Layer (Packet-Optical) Coordination for Protection/Restoration

```
                        +-----------------+
                        |   Customer      |
                        |   Network       |
                        |   Controller    |
                        +--------*--------+
                                 *   |     E2E VN Survivability Req.
                                 *   |     - VN Protection/Restoration
                                 *   v        - 1+1, Restoration, etc.
                          +------*------+   - End Point (EP) info.
                          |             |
                          |    MDSC     | MDSC enforces VN survivability
                          |             | requirement, determining the
                          |             | optimal combination of Packet/
                          +------*------+ Optical protection/restoration
                                 *        Optical bypass, etc.
                                 *
                                 *
              ***********************************************
              *               *              *              *
         +----*-----+    +----*----+    +----*-----+   +----*----+
         |PNC for   |    |PNC for  |    |PNC for   |   |PNC for  |
         |Access N. |    |Packet C.|    |Optical C.|   |Access N.|
         +----*-----+    +----*----+    +----*-----+   +---*-----+
              *             --*---         *             *
              *          ///      \\\        *             *
            --*---      |   Packet    |      *         ----*-
         ///      \\\   |    Core     +------+------///      \\\
        |  Access    +----\\      ///        *     |   Access   |
        |  Network   |      ---+--           *     |   Network  |  +---+
        |\\\      ///          |             *      \\\      ///---+EP6|
        |   +---+-  |          |        -----*         -+---+      +---+
      +-+-+     |   |          +----///      \\\       |   |
      |EP1|     |   +---------------+  Optical   |      |   |  +---+
      +---+     |                   |    Core    +-------+   +--+EP5|
              +-+-+                  \\\      ///          +---+
              |EP2|                     ------ |
              +---+                      |     |
                                     +--++    ++--+
                                     |EP3|    |EP4|
                                     +---+    +---+
```

Figure A.5: E2E VN Survivability and Multi-layer Coordination for Protection and Restoration

Figure A.5 shows the need for E2E protection/restoration control coordination that involves CNC, MDSC and PNCs to meet the VN survivability requirement. VN survivability requirement and its policy need to be translated into multi-domain and multi-layer network protection and restoration scenarios across different controller types. After an E2E path is setup successfully, the MDSC has a unique role to enforce policy-based flexible VN survivability requirement by coordinating all PNC domains.

As seen in Figure A.5, multi-layer (i.e., packet/optical) coordination is a subset of this E2E protection/restoration control operation. The MDSC has a role to play in determining an optimal protection/restoration level based on the customer's VN survivability requirement. For instance, the MDSC needs to interface the PNC for packet core as well as the PNC for optical core and enforce protection/restoration policy as part of the E2E protection/restoration. Neither the PNC for packet core nor the PNC for optical core is in a position to be aware of the E2E path and its protection/restoration situation. This role of the MDSC is unique for this reason. In some cases, the MDSC will have to determine and enforce optical bypass to find a feasible reroute path upon packet core network failure which cannot be resolved the packet core network itself.

To coordinate this operation, the PNCs will need to update its domain level abstract topology upon resource changes due to a network failure or other factors. The MDSC will incorporate all these update to determine if an alternative E2E reroute path is necessary or not based on the changes reported from the PNCs. It will need to update the E2E abstract topology and the affected CN's VN topology in real-time. This refers to dynamic synchronization of topology from Physical topology to abstract topology to VN topology.

MDSC will also need to perform the path restoration signaling to the affected PNCs whenever necessary.

TEAS Working Group Y. Lee (Editor)
Dhruv Dhody (Editor)
Huawei

Daniele Ceccarelli
Ericsson

Internet Draft
Intended Status: Informational
Expires: March 2017

September 14, 2016

Architecture and Requirement for Distribution of Link-State and TE Information via PCEP.

draft-leedhody-teas-pcep-ls-03

Abstract

In order to compute and provide optimal paths, Path Computation Elements (PCEs) require an accurate and timely Traffic Engineering Database (TED). Traditionally this Link State and TE information has been obtained from a link state routing protocol (supporting traffic engineering extensions).

This document provides possible architectural alternatives for link-state and TE information distribution and their potential impacts on PCE, network nodes, routing protocols etc.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

In Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS), a Traffic Engineering Database (TED) is used in computing paths for connection oriented packet services and for circuits. The TED contains all relevant information that a Path Computation Element (PCE) needs to perform its computations. It is important that the TED should be complete and accurate anytime so that the PCE can perform path computations.

In MPLS and GMPLS networks, Interior Gateway routing Protocols (IGPs) have been used to create and maintain a copy of the TED at each node. One of the benefits of the PCE architecture [RFC4655] is the use of computationally more sophisticated path computation algorithms and the realization that these may need enhanced processing power not necessarily available at each node participating in an IGP.

Section 4.3 of [RFC4655] describes the potential load of the TED on a network node and proposes an architecture where the TED is maintained by the PCE rather than the network nodes. However it does not describe how a PCE would obtain the information needed to populate its TED. PCE may construct its TED by participating in the IGP ([RFC3630] and [RFC5305] for MPLS-TE; [RFC4203] and [RFC5307] for GMPLS). An alternative is offered by [BGP-LS].

[RFC7399] touches upon this issue: "It has also been proposed that the PCE Communication Protocol (PCEP) [RFC5440] could be extended to serve as an information collection protocol to supply information from network devices to a PCE. The logic is that the network devices may already speak PCEP and so the protocol could easily be used to report details about the resources and state in the network, including the LSP state discussed in Sections 14 and 15."

[Stateful-PCE] describes a set of extensions to PCEP to provide stateful control. A stateful PCE has access to not only the information carried by the network's Interior Gateway Protocol (IGP), but also the set of active paths and their reserved resources for its computations. PCC can delegate the rights to modify the LSP parameters to an Active Stateful PCE. This requires PCE to quickly be updated on any changes in the Topology and TEDB, so that PCE can meet the need for updating LSPs effectively and in a timely manner.

The fastest way for a PCE to be updated on TED changes is via a direct interface with each network node and with incremental update from each network node.

[PCE-initiated] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed. This model requires timely topology and TED update at the PCE.

This document proposes alternative architecture approaches for learning and maintaining the Link State (and TE) information directly on a PCE from network nodes as an alternative to IGPs and BGP transport and investigate the impact from the PCE, routing protocol, and network node perspectives.

2. Applicability

Recent development of a stateful PCE Model changes the PCE operation from path computation alone to include the support of PCE-initiated LSPs. With a stateful PCE model, it is also noted that LSP-DB is maintained by the PCE. For LSP state synchronization of stateful PCEs in GMPLS networks, the LSP attributes, such as its bandwidth, associated route as well as protection information etc, should be updated by PCCs to PCE LSP database (LSP-DB) [S-PCE-GMPLS]. To support all these recent changes in a stateful PCE model, a direct PCE interface to each PCC has to be supported. Relevant TE resource and sate information can also be transported from each node to PCE using this PCC-PCE interface via PCEP. Any resource changes in the node and links can also be quickly updated to PCE using this interface. Convergence time of IGP in GMPLS networks may not be quick enough to support on-line dynamic connectivity required for some applications.

New application areas for GMPLS and PCE in optical transport networks include Wavelength Switched Optical Networking (WSON) and Optical Transport Networks (OTN). WSON scenarios can be divided into routing wavelength assignment (RWA) problems where a PCE requires detailed information about switching node asymmetries and wavelength constraints as well as detailed up to date information on wavelength usage per link [RFC6163]. As more data is anticipated to be made available to PCE with addition of OTN and Flex-grid and possible with some optical impairment data even with the minimum set specified in [G.680], the total amount of data requires significantly more information to be held in the TED than is required for other traffic engineered networks. Related to this issue published by [HWANG] indicated that long convergence time and large number of LSAs flooded in the network might cause scalability problems in OSPF-TE and impose limitations on OSPF-TE applications.

There are two main applicability of this alternative proposed by this draft:

- Where there is a need for a faster incremental link-state and TE resource and state population and convergence at the stateful PCE.

  - Where there is no IGP or BGP-LS running in the network nodes.

  - Where there is IGP or BGP-LS running but with a need for a faster incremental link-state (and TE) resource and state population and convergence at the PCE.

    - A PCE may receive partial Link-state (and TE) resource and state information (say basic TE) from IGP-TE and other information (optical and impairment) from PCEP.

    - A PCE may receive full Link-state (and TE) resource and state information from both IGP-TE and PCEP.

- Where there is no IGP or BGP-LS running at the PCE to learn Link-state (and TE) resource and state information.

A PCC may further choose to send only local TE resource and state information or both local and remote learned TE resource and state information. How a PCE manages the TE resource and state information is implementation specific and thus out of scope of this document.

This is also applicable for transporting (abstract) Link-State and TE information from child PCE to a Parent PCE in H-PCE [RFC6805]; as well as for Physical Network Controller (PNC) to Multi-Domain Service Coordinator (MDSC) in Abstraction and Control of TE Networks (ACTN) [ACTN].

This draft does not advocate that the alternative methods specified in this draft should completely replace the IGP-TE as the method of creating the TED. The split between the data to be distributed via an IGP-TE and the information conveyed via one of the alternatives in this document depends on the nature of the network situation. One could potentially choose to have some traffic engineering information distributed via an IGP-TE while other more specialized traffic information is only conveyed to the PCEs via an alternative interface discussed here.

In addition, the methods specified in this draft is only relevant to a set of architecture options where routing decisions are wholly or partially made in the PCE. On the other hand, the networks that do not support IGP-TE/BGP-LS, the method proposed by this draft may be very relevant.

3. Architecture Options

(1) There are two general architectural alternatives based on how nodes get their local link-state (and TE) resource information to the PCEs:

(1.1) All Nodes send local link-state (and TE) resource information to all PCEs;

(1.2) All Nodes send local link-state (and TE) resource information to a designated PCE and have the PCEs share this information with each other.

(2) Further, a designated node (PCC) can share both local and remote link-state (and TE) information to the PCEs, the remote information might be learned at the node via IGP:

(2.1) Designated Node(s) send local and remote link-state (and TE) resource information to all PCEs;

An important functionality that needs to be addressed in each of these approaches is how a new PCE gets initialized in a reasonably timely fashion.

Figures 1-2 show examples of two options for nodes to share local TE resource information with multiple PCEs. As in the IGP case we assume that switching nodes know their local properties and state including the state of all their local links. In these figures the data plane links are shown with the character "o"; Link-state and TE resource information flow from nodes to PCE by the characters "|", "-", "/", or "\"; and PCE to PCE link-state and TE information, if any, by the character "i".

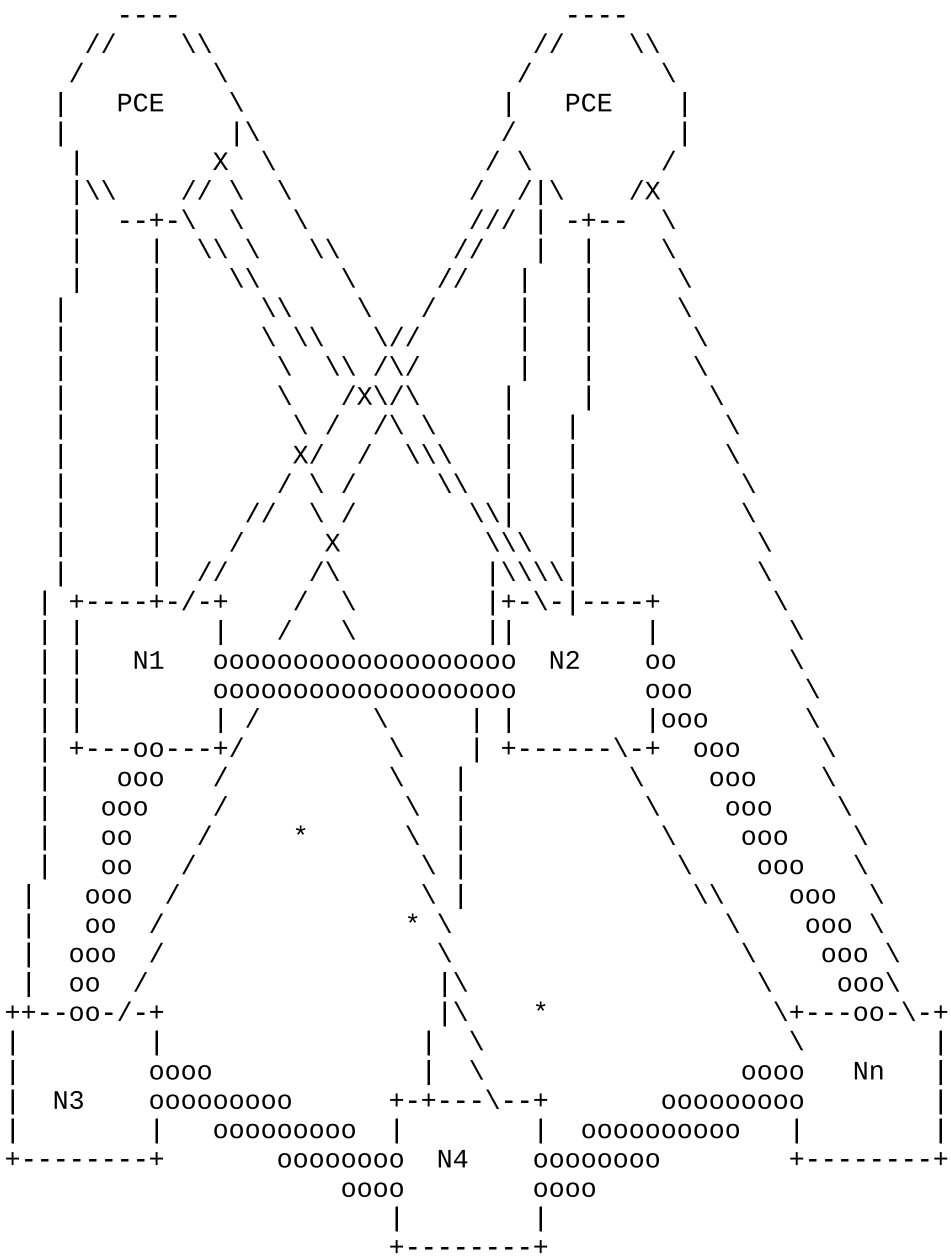

Figure 1 . Nodes send local Link-state and TE information directly to all PCEs

```
                         iiiiiiiiiiiiiiiiii
        iiiiii   ----  iii                 iiiii        ----
       ii    ii//    \\i                       iiiiiiii/    \\
      ii      /        \                             /        \
      i      |   PCE1   |                           |   PCE2   |
     i       |          |                           |          |ii
     i        \        /                             X        /  ii
    i          \\    //                            // \\    //    ii
    i            -//-                             /     --+-       i
    i           //                              //        |        i
   i     +-----/--+                       +----/---+       |        i
   i     |        |                       |        |       |        i
   i     |   N1   ooooooooooooooooooooooooo  N2    oooo    |        i
   i     |        ooooooooooooooooooooooooo        oooo    |        i
   i     |        |                       |        |  oo    |       i
   i     +---oo---+                       +--------+   oo   |        i
   i         oo                                        oo   |        i
   i         oo                                          oo  |        i
   i        oo           *                               oo  |        i
   i       oo                                            oo   |       i
   i       oo                                            oo   |       i
   i       oo                   *                         oo  |       i
   i       oo                                              oo  |      i
   i   +---oo---+                       *               +---oo-+-+    i
   i   |        |                                       |        |    i
   i   |        oooo                                 oooo   Nn   |    i
   i   |  N3    oooooooo       +--------+       ooooooooo        |    i
   ii  |        |    oooooo    |        |  ooooooooooo  |        |   ii
    i  +---\----+       oooooooo  N4    ooooooooo       +--------+   i
    i       \              ooooo        oooo                         i
    ii       \                 |        |                            i
     i        \\               +--------+                           ii
     ii         \              ---                                  i
      ii         \   ----  ---                                      i
       ii         \//    \--                                       i
        ii        /        \                                      ii
          ii     |   PCE3   |                                  iiii
            iiiii|          |                              iiiii
                  \        /                             iii
                   \\    // iiiiiiiii                 iii
                     ----           iiiiiiiiiiiiiiiiiii
```

Figure 2 . Nodes send local Link-state and TE information to one PCE and have the PCEs share TED information

```
          ----                       ----
        /      \                   /      \
       /        \                 /        \
      |   PCE    \               |   PCE    |
      |                        --+          |
       \        /            /     \        /
         \     /            /        \     /
          +-++             /          ----
            |             /
            |            /
            |           /
            |          /
            |         /
            |        /
            |       /
            |      /
            |     /
            |    /
            |   /
            |  /
       +----+-/-+                 +--------+
       |        +                 |        |
       |   N1   ooooooooooooooooooo   N2     oo
       |        ooooooooooooooooooo          ooo
       |        +                 |        |ooo
       +--+oo+--+                 +--------+  ooo
          ooo                                  ooo
         ooo                                    ooo
         oo                                      ooo
         oo                                       ooo
        ooo                                         ooo
        oo                                           oo
       ooo                                            ooo
       oo                                              ooo
   +---oo---+                                       +---oo---+
   |        |                                       |        |
   |        oo                                   oooo   Nn   |
   |  N3    ooooooooo      +--------+       ooooooooo        |
   |        |   ooooooooo  |        |  oooooooooo   |        |
   +--------+       oooooooo  N4    oooooooo        +--------+
                        oooo        oooo
                           |        |
                           +--------+
```

Figure 3. Designated Node sends local and remote Link-state and TE information directly to all PCEs

3.1. Option 1.1: All Nodes Send Local Link-State and TE Info to all PCEs

Architectural alternative 1 shown in Figure 1 illustrates nodes sending their local link-state (and TE) resource information to all PCEs within their domain. As the number of PCEs grows we may have scalability concerns. In particular, each node needs to keep track of which PCE it has sent information to and update that information periodically. However, if we are only talking about 2-3 PCEs, then we do not have this scalability concern.

If a new PCE is added to the domain all nodes must send all its local link-state and TE resource information to that PCE rather than just sending status updates.

3.2. Option 1.2: Each Node Sends Local Link-State and TE Info to one PCE

In this architectural alternative, shown in Figure 2, each node would be associated with one PCE. This implies that each PCE will only have partial link-state (and TE) resource information directly from the nodes. It would be the responsibility of a node to get its local information to its associated PCE, then the PCEs within a domain would then need to share the partial link-state (and TE)resource information they learned from their associated nodes with each other so that they can create and maintain the complete link-state (and TE)resource information.

To allow for this sharing of information PCEs would need to peer with each other. PCE discovery extensions [RFC4674] could be used to allow PCEs to find other PCEs. If a new PCE is added to the domain it would need to peer with at least one other PCE and then PCE synchronization mechanism could then be used to initialize the new PCEs link-state (and TE)resource information.

A number of approaches can be used to ensure control plane resilience in this architecture. (1) Each node can be configured with a primary and a secondary PCE to send its information to; In case of failure of communications with the primary PCE the node would send its information to a secondary PCE (warm standby). (2) Each node could be configured to send its information to two different PCEs (hot standby).

3.3. Option 2.1: Designated Node(s) Send Local and Remote Link-State and TE Info to all PCEs

In this architectural alternative, shown in Figure 3, illustrates designated node(s) sending their local and remote link-state (and TE) resource information to all PCEs within their domain. Designated Node may learn remote information via IGP or BGP-LS. More than one designated node may be used to ensure control plane resilience in this architecture.

3.4. Hierarchy of PCE

Incase of H-PCE [RFC6805], a parent PCE can utilize the mechanism described in this document to prepare the domain topology map by transporting inter-domain link information from child PCE to Parent PCE.

```
                        !!!!!!!!!!!!!!
                        ! Parent PCE !
                        !!!!!!!!!!!!!!
                       / |      |   \
                      /  |      |    \
                     /   |      |     \
                    /    |      |      \
                   /     |       \      \
                  /      |        \     |
                 /       |         \     \
                /        |         |      \
               /         |         |       \
              /     +----|------+  |        \
             /      |  ~~|~~~   |  |         \
            /       |  ~PCE2~   |  |          \
           /        |  ~~~~~~   |  |           \
          /         | Domain 2  |  |            \
         |          |           |  |       +-----\----+
         |          +-----------+  |       |  ~~~~\~  |
        /                          |       |  ~PCE4~  |
   +---/------+                    |       |  ~~~~~~  |
   |  /~~~~~  |                    |       | Domain 4 |
   |  ~PCE1~  |                    |       |          |
   |  ~~~~~~  |                    |       +----------+
   | Domain 1 |                    |
   |          |            +-------|----+
   +----------+            |  ~~~~~|    |
                           |  ~PCE3~    |
                           |  ~~~~~~    |
                           | Domain 3   |
                           |            |
                           +------------+
```

Figure 4. Child PCE sends some Link-state and TE information to Parent PCE

Further abstracted topology information can be transported from PNC to MDSC in ACTN [ACTN] using this technique described in this document.

3.5. Key Architectural Issues

3.5.1. Nodes Finding PCEs

In all cases, nodes need to send TE information directly to PCEs. Path Computation Clients (PCCs) and network nodes participating in an IGP (with or without TE extensions) have a mechanism to discover a PCE and its capabilities. [RFC4674] outlines the general requirements for this mechanism and extensions have been defined to provide information so that PCCs can obtain key details about available PCEs in OSPF [RFC5088] and in IS-IS [RFC5089].

After finding candidate PCEs, a node would need to see which if any of the PCEs actually want to receive TE information directly from this node.

3.5.2. Node TE Information Update Procedures

First a node must establish an association between itself and a PCE that will be maintaining a link-state and TE information. It is the responsibility of the node to share link-state (and TE) information. This includes local information, e.g., links and node properties or remote information learned from neighbors. General and technology specific information models would specify the content of this information while the specific protocols would determine the format. Note that data plane neighbor information would be passed to the PCE embedded in TE link information.

There will be cases where the node would have to send to the PCE only a subset of TE link information depending on the path computation option. For instance, if the node is responsible for routing while the PCE is responsible for wavelength assignment for the route, the node would only need to send the PCE the WSON link usage information. This path computation option is referred to as separate Fouting (R) and Wavelength Assignment (WA) option in [RFC7449].

3.5.3. PCE Link-state (and TE) Resource Information Maintenance Procedures

The PCE is responsible for creating and maintaining the link-state (and TE) resource information that it will use. Key functions include:

1. Establishing and authenticating communications between the PCE and sources of link-state (and TE) resource information.

2. Timely updates of the link-state (and TE) resource with information received from nodes, peers or other entities.

3. Verifying the validity of link-state (and TE) resource information, i.e., ensure that the network information obtained from nodes or elsewhere is relatively timely, or not stale. By analogy with similar functionality provided by IGPs this can be done via a process where discrete "chunks" of TE resource information are "aged" and discard when expired. This combined with nodes periodically resending their local TE resource information leads to a timely update of TE resource information.

4. Requirements for PCEP extension

The key requirements associated with link-state (and TE) distribution are identified for PCEP and listed in [PCEP-LS].

These new functions required in PCEP to support distribution of link-state (and TE) information are described in [PCEP-LS].

5. New Functions to distribute link-state and TE via PCEP

Several new functions are required in PCEP to support distribution of link-state and TE information. The new functions are:

- o Capability advertisement: Advertise capability for link-state and TE information distribution

- o Link-State and TE synchronization: Ability to synchronize the Link-state and TE information after session initialization.

o Link-State and TE Report (C-E): a PCC sends a LS and TE report to a PCE whenever the Link-State and TE information changes.

These are listed in some detail in [PCEP-LS]. Also see [PCEP-LS-Optical] for optical extension of [PCEP-LS].

6. Security Considerations

This draft discusses an alternative technique for PCEs to build and maintain a link-state and traffic engineering database. In this approach network nodes would directly send traffic engineering information to a PCE. It may be desirable to protect such information from disclosure to unauthorized parties in addition it may be desirable to protect such communications from interference (modification) since they can be critical to the operation of the network. In particular, this information is the same or similar to that which would be disseminated via a link state routing protocol with traffic engineering extensions.

7. IANA Considerations

This version of this document does not introduce any items for IANA to consider.

8. References

8.1. Normative References

[RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4674] Le Roux, J., Ed., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, October 2006.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, July 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

8.2. Informative References

[JMS] Java Message Service, Version 1.1, April 2002, Sun Microsystems.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[BGP-LS] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S.Ray, "North-Bound Distribution of Link-State and TE information using BGP", draft-ietf-idr-ls-distribution, work in progress.

[S-PCE-GMPLS] X. Zhang, et. al, "Path Computation Element (PCE) Protocol Extensions for Stateful PCE Usage in GMPLS-controlled Networks", draft-ietf-pce-pcep-stateful-pce-gmpls, work in progress.

[RFC7399] A. Farrel and D. king, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, October 2015.

[RFC7449] Y. Lee, G. Bernstein, "Path Computation Element Communication Protocol (PCEP) Requirements for Wavelength Switched Optical Network (WSON) Routing and Wavelength Assignment", RFC 7449, February 2015.

[RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)", RFC 4456, April 2006.

[RFC6163] Y. Lee, G. Bernstein, W. Imajuku, "Framework for GMPLS and PCE Control of Wavelength Switched Optical Networks", RFC 6163,

[HWANG] S. Hwang, et al, "An Experimental Analysis on OSPF-TE Convergence Time", Proc. SPIE 7137, Network Architectures, Management, and Applications, November 19, 2008.

[G.680] ITU-T Recommendation G.680, Physical transfer functions of optical network elements, July 2007.

[ACTN] Y. Lee, D. Dhody, S. Belotti, K. Pithewan, and D. Ceccarelli, "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.

[RFC6805] A. Farrel and D. King, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, November 2012.

[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.

[PCEP-LS-Optical] Y. Lee and D. Dhody (editors), "PCEP Extension for Distribution of Link-State and TE information for Optical Networks", draft-lee-pce-pcep-ls-optical, work in progress.

[Stateful-PCE] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce, work in progress.

[PCE-Initiated] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp, work in progress.

Author's Addresses

Young Lee (Editor)
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Email: leeyoung@huawei.com

Dhruv Dhody (Editor)
Huawei Technologies
Divyashree Technopark, Whitefield
Bangalore, Karnataka 560037
India

EMail: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm
Sweden

EMail: daniele.ceccarelli@ericsson.com

Haomian Zheng
Huawei Technologies Co., Ltd.
F3-1-B R&D Center, Huawei Base,
Bantian, Longgang District
Shenzhen 518129 P.R.China

Email: zhenghaomian@huawei.com

Xian Zhang
Huawei Technologies Co., Ltd.
F3-1-B R&D Center, Huawei Base,
Bantian, Longgang District
Shenzhen 518129 P.R.China

Email: zhangxian@huawei.com

q

Network Working Group J. George
Internet-Draft Google
Intended status: Informational L. Fang
Expires: April 21, 2016 Microsoft
E. Osborne
Level 3
R. Shakir
Jive Communications
October 19, 2015

MPLS / TE Model for Service Provider Networks
draft-openconfig-mpls-consolidated-model-02

Abstract

This document defines a framework for a YANG data model for configuring and managing label switched paths, including the signaling protocols, traffic engineering, and operational aspects based on carrier and content provider operational requirements.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a YANG [RFC6020] data model for MPLS and traffic engineering, covering label switched path (LSP) configuration, as well as signaling protocol configuration. The model is intended to be vendor-neutral, in order to allow operators to manage MPLS in heterogeneous environments with physical or virtual devices (routers, switches, servers, etc.) supplied by multiple vendors. The model is also intended to be readily mapped to existing implementations, to facilitate support from as large a set of routing hardware and software vendors as possible.

1.1. Goals and approach

The focus area of the model in this revision, is to set forth a framework for MPLS, with hooks into which information specific to various signaling-protocols can be added. The framework is built around functionality from a network operator perspective rather than a signaling protocol-centric approach. For example, a traffic-engineered LSP will have configuration relating to its path computation method, regardless of whether it is signaled with RSVP-TE or with segment routing. Thus, rather than creating separate per-signaling protocol models and trying to stitch them under a common umbrella, this framework focuses on functionality, and adds signaling protocol-specific information under it where applicable.

This model does not aim to be feature complete (i.e., cover all possible aspects or features of MPLS). Rather its development is driven by examination of actual production configurations in use across a number of operator network deployments.

Configuration items that are deemed to be widely available in existing major implementations are included in the model. Those configuration items that are only available from a single implementation are omitted from the model with the expectation they will be available in companion modules that augment the current model. This allows clarity in identifying data that is part of the vendor-neutral model.

An important aspect of the model is the representation of operational state data. This draft takes the approach described in [I-D.openconfig-netmod-opstate] and models configuration and operational state together. Thus, rather than building a separate tree of operational state, the operational state and configuration data are located in parallel containers at the leaves of the data model. This approach allows easy reuse of groupings across models, as well as making it easier to correlate configuration and state.

The consolidated MPLS model encompasses the signaling protocols, label-switched paths (configuration and operational state), and generic TE attributes. The model is designed from an operational and functional perspective, rather than focusing on protocol-centric configuration. This allows protocol-independent functions to be logically separated from protocol-specific details.

One question that arises in this approach is how the consolidated model is integrated with routing instances (e.g., VRFs). This model should be considered as part of a higher level network device model which includes definitions for other routing protocols and system services. For example, in [I-D.openconfig-netmod-model-structure], VRFs and other logical instances are defined with MPLS/TE components within VRFs as appropriate. In particular, some parts of the MPLS model would be instantiated within a VRF, while other parts would have common definitions across VRFs.

Where possible, naming in the model follows conventions used in available standards documents, and otherwise tries to be self-explanatory with sufficient descriptions of the intended behavior. Similarly, configuration data value constraints and default values, where used, are based on recommendations in current standards documentation. Since implementations vary widely in this respect, this version of the model specifies only a limited set of defaults and ranges with the expectation of being more prescriptive in future versions based on actual operator use.

Note that this version of the model is a work-in-progress in several respects. Although we present a complete framework for MPLS and traffic engineering from an operational perspective, some signaling protocol configuration will be completed in future revisions. The current revision has focus on traffic engineered LSPs signaled with RSVP.

2. Model overview

The overall MPLS model is defined across several YANG modules and submodules but at a high level is organized into 4 main sections:

- global -- configuration affecting MPLS behavior which exists independently of the underlying signaling protocol or label switched path configuration.

- te-global-attributes -- configuration affecting MPLS-TE behavior which exists independently of the underlying signaling protocol or label switched path configuration.

- signaling protocols -- configuration specific to signaling protocols used to setup and manage label switched paths.

- label switched paths -- configuration specific to instantiating and managing individual label switched paths.

The top level of the model is shown in the tree view below:

```
+--rw mpls!
      +--rw global
      |     ...
      +--rw te-global-attributes
      |     ...
      +--rw signaling-protocols
      |     ...
      +--rw lsps
            ...
```

2.1. MPLS global

The global section of the framework provides configuration data for MPLS items which exist independently of an individual label switched path or signaling protocol and are applicable to the MPLS protocol itself. Items such as the depth of the label stack supported, or specific label ranges may be included here.

2.2. TE global attributes

The TE global attributes section of the framework provides configuration control for MPLS-TE items which exist independently of an individual label switched path or signaling protocol. These standalone items are applicable to the entire logical routing device, and establish fundamental configuration such as the threshold for interface bandwidth change that triggers update events into the IGP traffic engineering database (TED). Timers are also specified which determine the length of time an LSP must be present before being considered viable for forwarding use (te-lsp-install-delay), and the length of time between LSP teardown and removal of the LSP from the network element’s forwarding information base (te-lsp-cleanup-delay). Also specified are the name to value mappings of MPLS administrative groups (mpls-admin-groups) and shared risk link groups (mpls-te-srlg).

```
    +--rw te-global-attributes
         |  +--rw mpls-te-srlg
         |  |  +--rw srlg* [srlg-name]
         |  |     +--rw srlg-name       leafref
         |  |     +--rw config
         |  |     |  +--rw srlg-name?    string
         |  |     |  +--rw srlg-value?   uint32
         |  |     |  +--rw srlg-cost?    uint32
         |  |     +--ro state
         ...
         |  |     +--rw members-list* [from-address]
         |  |        +--rw from-address    leafref
         |  |        +--rw config
         |  |        |  +--rw from-address?   inet:ip-address
         |  |        |  +--rw to-address?     inet:ip-address
         ...
         |  +--rw igp-flooding-bandwidth
         |  |  +--rw config
         |  |  |  +--rw threshold-type?            enumeration
         |  |  |  +--rw delta-percentage?          oc-types:percentage
         |  |  |  +--rw threshold-specification?   enumeration
         |  |  |  +--rw up-thresholds*             oc-types:percentage
         |  |  |  +--rw down-thresholds*           oc-types:percentage
         |  |  |  +--rw up-down-thresholds*        oc-types:percentage
         |  |  +--ro state
         ...
         |  +--rw mpls-admin-groups
         |  |  +--rw admin-group* [admin-group-name]
         |  |     +--rw admin-group-name    leafref
         |  |     +--rw config
         |  |     |  +--rw admin-group-name?    string
         |  |     |  +--rw admin-group-value?   uint32
         |  |     +--rw state
         ...
         |  +--rw te-lsp-timers
         |     +--rw config
         |     |  +--rw te-lsp-install-delay?      uint16
         |     |  +--rw te-lsp-cleanup-delay?      uint16
         |     |  +--rw te-lsp-reoptimize-timer?   uint16
         |     +--ro state
         ...
```

2.3. TE interface attributes overview

The TE interface attributes section of the framework provides configuration and state related to traffic engineering such as te-metric or shared risk link group configuration.

```
         +--rw te-intf-attributes
         |  +--rw interface* [interface-name]
         |     +--rw interface-name    leafref
         |     +--rw config
         |     |  +--rw interface-name?           ocif:interface-ref
         |     |  +--rw te-metric?                uint32
         |     |  +--rw srlg* [srlg-name]
         |     |        ...
         |     |  +--rw admin-group* [admin-group-name]
         |     |        ...
         |     |  +--rw igp-flooding-bandwidth
         |     |       ...
         |     +--ro state
         ...
```

2.4. Signaling protocol overview

The signaling protocol section of the framework provides configuration elements for configuring three major methods of signaling label switched paths: RSVP-TE, segment routing, and label distribution protocol (LDP). BGP-LU will be included in a future version of this draft by definitions in the BGP model ([I-D.ietf-idr-bgp-model]) and corresponding augmentations to the MPLS model.

```
   +--rw signaling-protocols
         |  +--rw rsvp-te
         |      ...
         |  +--rw segment-routing
         |      ...
         |  +--rw ldp
         |      ...
```

Configuration of RSVP-TE is centered around interfaces on the device which participate in the protocol. A key focus is to expose common RSVP-TE configuration parameters which are used to enhance scale and reliability. Items which are applicable globally in the RSVP-TE protocol such as graceful restart, soft preemption and various statistics are grouped into a global section under the protocol. RSVP neighbor and session state are also available in the RSVP section.

```
         +--rw rsvp-te
         |  |  +--rw rsvp-sessions
         |  |  |  +--rw config
         |  |  |  +--ro state
         |  |  |     +--ro rsvp-session* [source-port destination-port
                           source-address destination-address]
                     ...
         |  |  +--rw rsvp-neighbors
         |  |  |  +--rw config
         |  |  |  +--ro state
         |  |  |     +--ro rsvp-neighbor* [neighbor-address]
                     ...
         |  |  +--rw global
         |  |  |  +--rw graceful-restart
                     ...
         |  |  |  +--rw soft-preemption
                     ...
         |  |  |  +--ro statistics
         |  |  |     +--ro counters
                     ....
         |  |  +--rw interface-attributes
         |  |     +--rw interface* [interface-name]
         |  |        +--rw interface-name    leafref
                     ...
         |  |        +--rw rsvp-hellos
                     ...
         |  |        +--rw authentication
                     ...
         |  |        +--rw subscription
                     ...
         |  |        +--rw protection
                     ...
         ...
```

Containers for specifying signaling via segment routing and LDP are also present. Specific subelements will be added for those protocols, as well as for BGP labeled unicast, in the next revision.

2.5. LSP overview

This part of the framework contains LSP information. At the high level, LSPs are split into three categories: traffic-engineering-capable (constrained-path), non-traffic-engineered determined by the IGP (unconstrained-path), and hop-by-hop configured (static).

```
+--rw mpls!
      +--rw lsps
         +--rw constrained-path
         |     ...
         +--rw unconstrained-path
         |     ...
         +--rw static-lsps
               ...
```

The first two categories, constrained-path and unconstrained-path are the ones for which multiple signaling protocols exist, and are organized in protocol-specific and protocol-independent sections. For example, traffic-engineered (constrained path) LSPs may be set up using RSVP-TE or segment routing, and unconstrained LSPs that follow the IGP path may be signaled with LDP or with segment routing. IGP-determined LSPs may also be signaled by RSVP but this usage is not considered in the current version of the model.

A portion of the data model for constrained path traffic-engineered LSPs signaled with RSVP is shown below. It contains configuration for named explicit paths and for tunnels. Tunnel configuration differs for p2p and p2mp LSPs. In either case, some part of the model is signaling-protocol independent. For example for a p2p LSP, attributes such as the path computation method, the constraints for the the path, the bandwidth allocated to it, and even the frequency of reoptimization are signaling-protocol independent, while other data, such as the setup and hold priorities are protocol-specific and are specified in the protocol specific part of the model.

```
+--rw mpls!
      +--rw lsps
         +--rw constrained-path
        |  +--rw explicit-path* [name]
             ...
          |  +--rw tunnel* [name type]
          |     +--rw name       leafref
          |     +--rw type       leafref
          |     +--rw config
          |     |  +--rw name?                        string
          |     |  +--rw type?                        identityref
          |     |  +--rw local-id?                    union
          |     |  +--rw description?                 string
          |     |  +--rw admin-status?                identityref
          |     |  +--rw preference?                  uint8
          |     |  +--rw metric?                      te-metric-type
          |     |  +--rw (bandwidth)?
                   ...
          |     |  +--rw protection-style-requested?   identityref
          |     |  +--rw te-lsp-reoptimize-timer?      uint16
          |     |  +--rw (signaling-specific-tunnel-attributes)?
          |     |  |  +--:(RSVP)
          |     |  |     +--rw source?                      inet:ip-address
          |     |  |     +--rw soft-preemption?             boolean
          |     |  +--rw (tunnel-type)?
          |     |     +--:(p2p)
          |     |        |  +--rw destination?              inet:ip-address
          |     |        |  +--rw primary-paths* [name]
          |     |        |  |  +--rw name                   string
          |     |        |  |  +--rw preference?            uint8
          |     |        |  |  +--rw path-computation-method
                                ...
          |     |        |  |  +--rw admin-groups
                                ...
          |     |        |  |  +--rw no-cspf?               empty
          |     |        |  |  +--rw (sigaling-specific-path-attributes)?
          |     |        |  |     +--:(RSVP)
          |     |        |  |     |  +--rw setup-priority?           uint8
          |     |        |  |     |  +--rw hold-priority?            uint8
          |     |        |  |     |  +--rw retry-timer?              uint16
          |     |        |  |     +--:(SR)
          |     |        |  |        +--rw sid-selection-mode?       enumeration
          |     |        |  |        +--rw sid-protection-required?   boolean
          |     |        |  +--rw secondary-paths* [name]
                            ...
          |     +--ro state
                ...
```

Similarly, the partial model for non-traffic-engineered, or IGP-based, LSPs is shown below:

```
+--rw mpls!
      +--rw lsps
         +--rw unconstrained-path
            +--rw path-setup-protocol
               +--rw ldp!
               |     ...
               +--rw segment-routing!
                     ...
```

3. Example use cases

3.1. Traffic engineered p2p LSP signaled with RSVP

A possible scenario may be the establishment of a mesh of traffic-engineered LSPs where RSVP signaling is desired, and the LSPs use a local constrained path calculation to determine their path. These LSPs would fall into the category of a constrained-path LSP, and the tunnel type is p2p. Attributes such as metric, bandwidth or the style of protection desired are also defined at this (protocol-independent) level in the model. The path is defined to be locally-computed under the path-computation-method container, specifying the use of CSPF (use-cspf). Additional attributes for the path, such as its RSVP priorities are specified at the path level under the protocol-specific stanza.

```
+--rw mpls!
      +--rw lsps
         +--rw constrained-path
            ...
         |  +--rw tunnel* [name type]
         |     +--rw name      leafref
         |     +--rw type      leafref
         |     +--rw config
         |     |  +--rw name?                         string
         |     |  +--rw type?                         identityref
         |     |  +--rw metric?                       te-metric-type
         |     |  +--rw (bandwidth)?
                  ...
         |     |  +--rw protection-style-requested?   identityref
         |     |  +--rw te-lsp-reoptimize-timer?      uint16
                  ...
         |     |  +--rw (tunnel-type)?
         |     |     +--:(p2p)
         |     |     |  +--rw destination?                  inet:ip-address
         |     |     |  +--rw primary-paths* [name]
         |     |     |  |  +--rw name                       string
         |     |     |  |  +--rw preference?                uint8
         |     |     |  |  +--rw path-computation-method
                              ...
         |     |     |  |  +--rw admin-groups
                              ...
         |     |     |  |  +--rw no-cspf?                   empty
         |     |     |  |  +--rw (sigaling-specific-path-attributes)?
         |     |     |  |     +--:(RSVP)
         |     |     |  |     |  +--rw setup-priority?            uint8
         |     |     |  |     |  +--rw hold-priority?             uint8
         |     |     |  |     |  +--rw retry-timer?               uint16
         |     |     |  |     +--:(SR)
         |     |     |  |        +--rw sid-selection-mode?        enumeration
         |     |     |  |        +--rw sid-protection-required?   boolean
         |     |     |  +--rw secondary-paths* [name]
                        ...
         |     +--ro state
```

3.2. Traffic engineered LSP signaled with SR

A possible scenario may be the establishment of disjoint paths in a network where there is no requirement for per-LSP state to be held on midpoint nodes within the network, or RSVP-TE is unsuitable (as described in [I-D.ietf-spring-segment-routing-mpls] and [I-D.shakir-rtgwg-sr-performance-engineered-lsps]). Such LSPs fall in the constrained-path category. Similar to any other traffic engineered LSPs, the path computation method must be specified. Path attributes, such as the as lsp- placement-constraints (expressed as administrative groups) or metric must be defined. Finally, the path must be specified in a signaling- protocol specific manner appropriate for SR. The same configuration elements from the tree above apply in this case, except that path setup is done by the head-end by building a label stack, rather than signaled.

3.3. IGP-congruent LDP-signaled LSP

A possible scenario may be the establishment of a full mesh of LSPs. When traffic engineering is not an objective, no constraints are placed on the end-to-end path, and the best- effort path can be setup using LDP signaling simply for label distribution. The LSPs follow IGP-computed paths, and fall in the unconstrained-path category in the model. Protocol-specific configuration pertaining to the signaling protocol used, such as the FEC definition and metrics assigned are in the path- setup-protocol portion of the model.

The relevant part of the model for this case is shown below:

```
+--rw mpls!
    +--rw lsps
       +--rw unconstrained-path
          +--rw path-setup-protocol
             +--rw ldp!
                +--rw tunnel
                   +--rw tunnel-type?   mplst:tunnel-type
                   +--rw ldp-type?      enumeration
                   +--rw p2p-lsp
                   |  +--rw fec-address*   inet:ip-prefix
                   +--rw p2mp-lsp
                   +--rw mp2mp-lsp
```

A common operational issue encountered when using LDP is traffic blackholing under the following scenario: when an IGP failure occurs, LDP is not aware of it as these are two protocols running independently, resulting in traffic blackholing at the IGP failure point even though LDP is up and running. LDP-IGP synchronization [RFC5443] can be used to cost out the IGP failing point/segment to avoid the blackholing issue. The LDP-IGP synchronization function will be incorporated in a future version of this document.

Note that targeted LDP sessions are not discussed in this use case, and will be incorporated as a separate use case in a future version of this document.

4. Security Considerations

MPLS configuration has a significant impact on network operations, and as such any related protocol or model carries potential security risks.

YANG data models are generally designed to be used with the NETCONF protocol over an SSH transport. This provides an authenticated and secure channel over which to transfer BGP configuration and operational data. Note that use of alternate transport or data encoding (e.g., JSON over HTTPS) would require similar mechanisms for authenticating and securing access to configuration data.

Most of the data elements in the configuration model could be considered sensitive from a security standpoint. Unauthorized access or invalid data could cause major disruption.

5. IANA Considerations

This YANG data model and the component modules currently use a temporary ad-hoc namespace. If and when it is placed on redirected for the standards track, an appropriate namespace URI will be registered in the IETF XML Registry" [RFC3688]. The MPLS YANG modules will be registered in the "YANG Module Names" registry [RFC6020].

6. YANG modules

The modules and submodules comprising the MPLS configuration and operational model are currently organized as depcited below.

```
                                 +-------+
               +---------------->| MPLS  |<--------------+
               |                 +-------+               |
               |                     ^                   |
               |                     |                   |
          +----+-----+      +--------+-------+     +-----+-----+
          | TE LSPs  |      | IGP-based LSPs |     |static LSPs|
          |          |      |                |     |           |
          +----------+      +----------------+     +-----------+
              ^  ^                    ^  ^
              |  +----------------+   |  +--------+
              |                   |   |           |
              |   +------+      +-+---+-+      +--+--+
              +---+ RSVP |      |SEGMENT|      | LDP |
                  +------+      |ROUTING|      +-----+
                                +-------+
```

The base MPLS module includes submodules describing the three different types of support LSPs, i.e., traffic-engineered (constrained-path), IGP congruent (unconstrained-path), and static. The signaling protocol specific parts of the model are described in separate modules for RSVP, segment routing, and LDP. As mentioned earlier, support for BGP labeled unicast is also planned in a future revision.

A module defining various reusable MPLS types is included, and these modules also make use of the standard Internet types, such as IP addresses, as defined in RFC 6991 [RFC6991].

6.1. MPLS base modules

```
                    <CODE BEGINS> file openconfig-mpls.yang
module openconfig-mpls {

  yang-version "1";

  // namespace
  namespace "http://openconfig.net/yang/mpls";

  prefix "mpls";


  // import some basic types
  import openconfig-mpls-rsvp { prefix rsvp; }
  import openconfig-mpls-sr { prefix sr; }
  import openconfig-mpls-ldp { prefix ldp; }
  import openconfig-types { prefix oc-types; }
  import openconfig-interfaces { prefix ocif; }

  // include submodules
  include openconfig-mpls-te;
  include openconfig-mpls-igp;
  include openconfig-mpls-static;


  // meta
  organization "OpenConfig working group";

  contact
    "OpenConfig working group
    netopenconfig@googlegroups.com";

  description
    "This module provides data definitions for configuration of
```

Multiprotocol Label Switching (MPLS) and associated protocols for signaling and traffic engineering.

RFC 3031: Multiprotocol Label Switching Architecture

The MPLS / TE data model consists of several modules and submodules as shown below. The top-level MPLS module describes the overall framework. Three types of LSPs are supported:

i) traffic-engineered (or constrained-path)

ii) IGP-congruent (LSPs that follow the IGP path)

iii) static LSPs which are not signaled

The structure of each of these LSP configurations is defined in corresponding submodules. Companion modules define the relevant configuration and operational data specific to key signaling protocols used in operational practice.

```
                                 +-------+
               +---------------->| MPLS  |<--------------+
               |                 +-------+               |
               |                     ^                   |
               |                     |                   |
          +----+-----+      +--------+-------+     +-----+-----+
          | TE LSPs  |      | IGP-based LSPs |     |static LSPs|
          |          |      |                |     |           |
          +----------+      +----------------+     +-----------+
              ^  ^                    ^  ^
              |  +----------------+   |  +--------+
              |                   |   |           |
              |   +------+      +-+---+-+      +--+--+
              +---+ RSVP |      |SEGMENT|      | LDP |
                  +------+      |ROUTING|      +-----+
                                +-------+
```

```
  ";

revision "2015-10-14" {
  description
    "Work in progress";
  reference "TBD";
}

// extension statements

// feature statements
```

```
   // identity statements


   // grouping statements

   grouping mpls-admin-group_config {
     description
       "configuration data for MPLS link admin groups";

       leaf admin-group-name {
         type string;
         description "name for mpls admin-group";
       }

       leaf admin-group-value {
         type uint32;
         description "value for mpls admin-group";
       }

   }

   grouping mpls-admin-groups-top {

     description "top-level mpls admin-groups config
       and state containers";

     container mpls-admin-groups {
       description
         "Top-level container for admin-groups configuration
         and state";

       list admin-group {
         key admin-group-name;
         description "configuration of value to name mapping
           for mpls affinities/admin-groups";

         leaf admin-group-name {
           type leafref {
             path "../mpls:config/mpls:admin-group-name";
           }
           description
             "name for mpls admin-group";
         }
         container config {
           description "Configurable items for admin-groups";
           uses mpls-admin-group_config;
         }
```

```
          container state {
            description "Operational state for admin-groups";
            uses mpls-admin-group_config;
          }
        }
      }
    }

    grouping mpls-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 oc-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 oc-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 oc-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 oc-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";
      }
    }

    grouping mpls-te-igp-flooding-bandwdith-if {
      description "Interface-level group for traffic engineering
      database flooding options options";
```

```
        container igp-flooding-bandwidth {
         description "Interface bandwidth change percentages
         that trigger update events into the IGP traffic
         engineering database (TED)";
         uses  mpls-te-igp-flooding-bandwidth_config;
        }
     }


     grouping  mpls-te-igp-flooding-bandwidth {
       description "Top level group for traffic engineering
         database flooding options";
       container igp-flooding-bandwidth {
         description "Interface bandwidth change percentages
           that trigger update events into the IGP traffic
           engineering database (TED)";
         container config {
           description "Configuration parameters for TED
             update threshold ";
           uses  mpls-te-igp-flooding-bandwidth_config;
         }
         container state {
           config false;
           description "State parameters for TED update threshold ";
           uses  mpls-te-igp-flooding-bandwidth_config;
         }
       }
     }


     grouping te_lsp_delay_config {
       description "Group for the timers goerning the delay
       in installation and cleanup of TE LSPs";

       leaf te-lsp-install-delay {
         type uint16 {
           range 0..3600;
         }
         units seconds;
         description "delay the use of newly installed te lsp for a
         specified amount of time.";
       }

       leaf te-lsp-cleanup-delay {
         type uint16;
         units seconds;
         description "delay the removal of old te lsp for a specified
         amount of time";
```

```
      }
    }

    grouping te-interface-attributes-top {
      description
        "Top level grouping for attributes
        for TE interfaces.";

      list interface {
        key interface-name;
        description "List of TE interfaces";

        leaf interface-name {
          type leafref {
            path "../config/interface-name";
            require-instance true;
          }
          description "The interface name";
        }

        container config {
          description
            "Configuration parameters related to TE interfaces:";
          uses te-interface-attributes-config;
        }

        container state {
          config false;
          description "State parameters related to TE interfaces";
          uses te-interface-attributes-config;
        }
      }
    }

    grouping te-interface-attributes-config {
      description "global level definitions for interfaces
        on which TE is run";

      leaf interface-name {
        type ocif:interface-ref;
        description "reference to interface name";
      }

      leaf te-metric {
        type uint32;
        description "TE specific metric for the link";
      }
```

```
       list srlg {
         key srlg-name;
         description "list of shared risk link groups on the
           interface";
         leaf srlg-name {
           type string;
           description "The SRLG group identifier";
         }
       }

       list admin-group {
         key admin-group-name;
         description "list of admin groups on the
           interface";
         leaf admin-group-name {
           type string;
           description "The admin group identifier";
         }
       }

       uses  mpls-te-igp-flooding-bandwdith-if;

     }

     grouping mpls-te-lsp-timers {
       description
         "Grouping for traffic engineering timers";
       container te-lsp-timers {
         description
         "definition for delays associated with setup
         and cleanup of TE LSPs";

         container config {
           description
           "Configuration parameters related
           to timers for TE LSPs";

           uses te_lsp_delay_config;
           uses te-tunnel-reoptimize_config;
         }
         container state {
           config false;
           description "State related to timers for TE LSPs";

           uses te_lsp_delay_config;
           uses te-tunnel-reoptimize_config;
         }
       }
```

```
     }

   container mpls {
      presence "top-level container for MPLS config and operational
      state";

      description "Anchor point for mpls configuration and operational
      data";

      container global {
        // entropy label support, label ranges will be added here.
       description "general mpls configuration applicable to any
       type of LSP and signaling protocol - label ranges,
       entropy label supportmay be added here";
      }

      container te-global-attributes {
        description "traffic-engineering global attributes";
        uses mpls-te-srlg-top;
        uses mpls-te-igp-flooding-bandwidth;
        uses mpls-admin-groups-top;
        uses mpls-te-lsp-timers;
      }

      container te-intf-attributes {
        description "traffic engineering attributes specific
        for interfaces";
        uses te-interface-attributes-top;
      }

      container signaling-protocols {
        description "top-level signaling protocol configuration";

        uses rsvp:rsvp-global;
        uses sr:sr-global;
        uses ldp:ldp-global;
      }

      container lsps {
        description "LSP definitions and configuration";

        container constrained-path {
          description "traffic-engineered LSPs supporting different
          path computation and signaling methods";
          uses explicit-paths-top;
          uses te-tunnels-top;

        }
```

```
         container unconstrained-path {
           description "LSPs that use the IGP-determined path, i.e., non
           traffic-engineered, or non constrained-path";

           uses igp-lsp-common;
           uses igp-lsp-setup;

         }

         container static-lsps {
           description "statically configured LSPs, without dynamic
             signaling";

           uses static-lsp-main;
         }
       }
     }

     // augment statements

     // rpc statements

     // notification statements
   }
                    <CODE ENDS>



                    <CODE BEGINS> file openconfig-mpls-types.yang
   module openconfig-mpls-types {

     yang-version "1";

     // namespace
     namespace "http://openconfig.net/yang/mpls-types";

     prefix "mplst";

     // meta
     organization "OpenConfig working group";

     contact
       "OpenConfig working group
       netopenconfig@googlegroups.com";

     description
       "General types for MPLS / TE data model";
```

```
   revision "2015-10-04" {
     description
       "Work in progress";
     reference "TBD";
   }

   // extension statements

   // feature statements

   // identity statements

   // using identities rather than enum types to simplify adding new
   // signaling protocols as they are introduced and supported
   identity path-setup-protocol {
     description "base identity for supported MPLS signaling
     protocols";
   }

   identity path-setup-rsvp {
     base path-setup-protocol;
     description "RSVP-TE signaling protocol";
   }

   identity path-setup-sr {
     base path-setup-protocol;
     description "Segment routing";
   }

   identity path-setup-ldp {
     base path-setup-protocol;
     description "lDP - RFC 5036";
   }


   identity protection-type {
     description "base identity for protection type";
   }

   identity unprotected {
     base protection-type;
     description "no protection is desired";
   }

   identity link-protection-requested {
     base protection-type;
     description "link protection is desired";
   }
```

```
     identity link-node-protection-requested {
       base protection-type;
       description "node and link protection are both desired";
     }

     identity lsp-role {
       description
         "Base identity for describing the role of
          label switched path at the current node";
     }

     identity INGRESS {
       base "lsp-role";
       description
         "Label switched path is an ingress (headend)
          LSP";
     }

     identity EGRESS {
       base "lsp-role";
       description
         "Label switched path is an egress (tailend)
          LSP";
     }

     identity TRANSIT {
       base "lsp-role";
       description
         "Label switched path is a transit LSP";
     }


     identity tunnel-type {
       description
         "Base identity from which specific tunnel types are
         derived.";
     }

     identity P2P {
       base tunnel-type;
       description
         "TE point-to-point tunnel type.";
     }

     identity P2MP {
       base tunnel-type;
       description
         "TE point-to-multipoint tunnel type.";
```

```
     }


     identity lsp-oper-status {
       description
         "Base identity for LSP operational status";
     }

     identity DOWN {
       base "lsp-oper-status";
       description
         "LSP is operationally down or out of service";
     }

     identity UP {
       base "lsp-oper-status";
       description
         "LSP is operationally active and available
          for traffic.";
     }

     identity tunnel-admin-status {
       description
         "Base identity for tunnel administrative status";
     }

     identity ADMIN_DOWN {
       base "tunnel-admin-status";
       description
         "LSP is administratively down";
     }

     identity ADMIN_UP {
       base "tunnel-admin-status";
       description
         "LSP is administratively up";
     }

     // typedef statements
     typedef mpls-label {
       type union {
         type uint32 {
           range 16..1048575;
         }
         type enumeration {
           enum IPV4_EXPLICIT_NULL {
             value 0;
             description "valid at the bottom of the label stack,
```

```
            indicates that stack must be popped and packet forwarded
            based on IPv4 header";
          }
          enum ROUTER_ALERT {
            value 1;
            description "allowed anywhere in the label stack except
            the bottom, local router delivers packet to the local CPU
            when this label is at the top of the stack";
          }
          enum IPV6_EXPLICIT_NULL {
            value 2;
            description "valid at the bottom of the label stack,
            indicates that stack must be popped and packet forwarded
            based on IPv6 header";
          }
          enum IMPLICIT_NULL {
            value 3;
            description "assigned by local LSR but not carried in
            packets";
          }
          enum ENTROPY_LABEL_INDICATOR {
            value 7;
            description "Entropy label indicator, to allow an LSR
            to distinguish between entropy label and applicaiton
            labels RFC 6790";
          }
        }
      }
      description "type for MPLS label value encoding";
      reference "RFC 3032 - MPLS Label Stack Encoding";
    }

    typedef tunnel-type {
      type enumeration {
        enum P2P {
          description "point-to-point label-switched-path";
        }
        enum P2MP {
          description "point-to-multipoint label-switched-path";
        }
        enum MP2MP {
          description "multipoint-to-multipoint label-switched-path";
        }
      }
      description "defines the tunnel type for the LSP";
      reference
        "RFC 6388 - Label Distribution Protocol Extensions for
        Point-to-Multipoint and Multipoint-to-Multipoint Label Switched
```

```
        Paths
        RFC 4875 - Extensions to  Resource Reservation Protocol
        - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
        Label Switched Paths (LSPs)";
    }

    typedef bandwidth-kbps {
      type uint64;
      units kbps;
    }

    typedef bandwidth-mbps {
      type uint64;
      units mbps;
    }

    typedef bandwidth-gbps {
      type uint64;
      units gbps;
    }

    // grouping statements

    // data definition statements

    // augment statements

    // rpc statements

    // notification statements

  }
                   <CODE ENDS>
```

6.2. MPLS LSP submodules

```
                   <CODE BEGINS> file openconfig-mpls-te.yang
  submodule openconfig-mpls-te {

    yang-version "1";

    belongs-to "openconfig-mpls" {
      prefix "mpls";
    }
```

```
// import some basic types
import ietf-inet-types { prefix inet; }
import openconfig-mpls-rsvp { prefix rsvp; }
import openconfig-mpls-sr { prefix sr; }
import openconfig-mpls-types {prefix mplst; }
import openconfig-types { prefix oc-types; }
import ietf-yang-types { prefix yang; }


// meta
organization "OpenConfig working group";

contact
  "OpenConfig working group
  netopenconfig@googlegroups.com";

description
  "Configuration related to constrained-path LSPs and traffic
  engineering.  These definitions are not specific to a particular
  signaling protocol or mechanism (see related submodules for
  signaling protocol-specific configuration).";

revision "2015-10-04" {
  description
    "Work in progress";
  reference "TBD";
}

// extension statements

// feature statements

// identity statements

// using identities for path comp method, though enums may also
// be appropriate if we decided these are the primary computation
// mechanisms in future.
identity path-computation-method {
  description
   "base identity for supported path computation
    mechanisms";
}

identity locally-computed {
  base path-computation-method;
  description
    "indicates a constrained-path LSP in which the
    path is computed by the local LER";
```

```
     }

     identity externally-queried {
       base path-computation-method;
       description
        "constrained-path LSP in which the path is
         obtained by querying an external source, such as a PCE server";
     }

     identity explicitly-defined {
       base path-computation-method;
       description
        "constrained-path LSP in which the path is
         explicitly specified as a collection of strict or/and loose
         hops";
     }

     // typedef statements

     typedef mpls-hop-type {
       type enumeration {
         enum LOOSE {
           description "loose hop in an explicit path";
         }
         enum STRICT {
           description "strict hop in an explicit path";
         }
       }
       description
        "enumerated type for specifying loose or strict
         paths";
     }

     typedef te-metric-type {
       type union {
         type enumeration {
           enum IGP {
             description
              "set the LSP metric to track the underlying
               IGP metric";
           }
         }
         type uint32;
       }
       description
        "union type for setting the LSP TE metric to a
         static value, or to track the IGP metric";
     }
```

```
typedef cspf-tie-breaking {
  type enumeration {
    enum RANDOM {
      description
       "CSPF calculation selects a random path among
        multiple equal-cost paths to the destination";
    }
    enum LEAST_FILL {
      description
       "CSPF calculation selects the path with greatest
        available bandwidth";
    }
    enum MOST_FILL {
      description "CSPF calculation selects the path with the least
        available bandwidth";
    }
  }
  default RANDOM;
  description
   "type to indicate the CSPF selection policy when
    multiple equal cost paths are available";
}


// grouping statements

grouping te-tunnel-reoptimize_config {
  description "Definition for reoptimize timer configuration";
  leaf te-lsp-reoptimize-timer {
    type uint16;
    units seconds;
    description
     "frequency of reoptimization of
      a traffic engineered LSP";
  }
}

grouping path-placement-constraints {
  description
    "Top level grouping for path placement constraints";

    container admin-groups {
      description
        "Include/Exclude constraints for link affinities";
      uses te-lsp-exclude-admin-group_config;
      uses te-lsp-include-any-admin-group_config;
      uses te-lsp-include-all-admin-group_config;
    }
```

```
     }

     grouping te-tunnel-bandwidth_config {
       description "Bandwidth configuration for TE LSPs";
       choice bandwidth {
         default explicit;
         description
          "select how bandwidth for the LSP will be
           specified and managed";
         case explicit {
           leaf set-bandwidth {
             type uint32;
             description
              "set bandwidth explicitly, e.g., using
               offline calculation";
           }
         }
         case auto {
           uses te-lsp-auto-bandwidth_config;
         }
       }
     }

     grouping te-lsp-auto-bandwidth_config {
       description "Configuration parameters related to autobandwidth";
       container auto-bandwidth {
         description
          "configure auto-bandwidth operation in
           which devices automatically adjust bandwidth to meet
           requirements";

         leaf enabled {
           type boolean;
           default false;
           description
            "enables mpls auto-bandwidth on the
             lsp";
         }

         leaf min-bw {
           type uint32;
           description
            "set the minimum bandwidth in Mbps for an
             auto-bandwidth LSP";
         }

         leaf max-bw {
           type uint32;
```

```
          description
           "set the maximum bandwidth in Mbps for an
            auto-bandwidth LSP";
        }

        leaf adjust-interval {
          type uint32;
          description
           "time in seconds between adjustments to
            LSP bandwidth";
        }

        leaf adjust-threshold {
          type oc-types:percentage;
          description
           "percentage difference between the LSP’s
            specified bandwidth and its current bandwidth
            allocation -- if the difference is greater than the
            specified percentage, auto-bandwidth adjustment is
            triggered";
        }

        container overflow {
          description
           "configuration of MPLS overflow bandwidth
            adjustement for the LSP";
          uses te-lsp-overflow_config;
        }

        container underflow {
          description
           "configuration of MPLS underflow bandwidth
            adjustement for the LSP";
          uses te-lsp-underflow_config;
        }
      }
    }

    grouping  te-lsp-overflow_config {
      description
       "configuration for mpls lsp bandwidth
        overflow adjustment";

      leaf enabled {
        type boolean;
        default false;
        description
         "enables mpls lsp bandwidth overflow
```

```
            adjustment on the lsp";
        }

        leaf overflow-threshold {
          type oc-types:percentage;
          description
           "bandwidth percentage change to trigger
            an overflow event";

        }

        leaf trigger-event-count {
          type uint16;
          description
           "number of consecutive overflow sample
            events needed to trigger an overflow adjustment";
        }
      }

      grouping te-lsp-underflow_config {
        description
          "configuration for mpls lsp bandwidth
          underflow adjustment";

        leaf enabled {
          type boolean;
          default false;
          description
           "enables bandwidth underflow
            adjustment on the lsp";
        }

        leaf underflow-threshold {
          type oc-types:percentage;
          description
           "bandwidth percentage change to trigger
            and underflow event";
        }

        leaf trigger-event-count {
          type uint16;
          description
           "number of consecutive underflow sample
            events needed to trigger an underflow adjustment";
        }
      }

      grouping te-tunnel-metric_config {
```

```
        description "Configuration parameters related to LSP metric";
        leaf metric {
          type te-metric-type;
          description "LSP metric, either explicit or IGP";
        }
      }

      grouping te-lsp-exclude-admin-group_config {
        description
         "Configuration parameters related to admin-groups
          to exclude in path calculation";
        list exclude-groups {

          key exclude-admin-group-name;

          description
            "list of admin-groups to exclude in path calculation";

          leaf exclude-admin-group-name {
            type leafref {
              path "/mpls/te-global-attributes/mpls-admin-groups/" +
                "admin-group/admin-group-name";
            }
            description
              "name of the admin group -- references a defined admin
              group";
          }
        }
      }

      grouping te-lsp-include-all-admin-group_config {
        description
         "Configuration parameters related to admin-groups
          which all must be included in the path calculation";
        list include-all-groups {

          key all-admin-group-name;
          description
            "list of admin-groups of which all must be included";

          leaf all-admin-group-name {
            type leafref {
              path "/mpls/te-global-attributes/mpls-admin-groups/" +
                "admin-group/admin-group-name";
            }
            description
              "name of the admin group -- references a defined
              admin group";
```

```
        }
      }
    }

    grouping te-lsp-include-any-admin-group_config {
      description
       "Configuration parameters related to admin-groups
        of which one must be included in the path calculation";
      list include-any-groups {

        key any-admin-group-name;
        description
          "list of admin-groups of which one must be included";

        leaf any-admin-group-name {
          type leafref {
            path "/mpls/te-global-attributes/mpls-admin-groups/" +
              "admin-group/admin-group-name";
          }
          description
            "name of the admin group -- references a defined
            admin group";
        }
      }
    }

    grouping te-tunnel-protection_config {
      description
       "Configuration parameters related to LSP
        protection";
      leaf protection-style-requested {
        type identityref {
          base mplst:protection-type;
        }
        default mplst:unprotected;
        description
          "style of mpls frr protection desired: can be
          link, link-node or unprotected.";
      }
    }

    grouping te-lsp-comp-explicit {
      description
       "definitions for LSPs in which hops are explicitly
        specified";

      container explicit-path {
        description "LSP with explicit path specification";
```

```
         leaf path-name {
           type leafref {
             path "/mpls/lsps/constrained-path/"
               + "explicit-path/config/named-explicit-paths/name";
             require-instance true;
           }
           description "reference to a defined path";
         }
       }
     }

     grouping te-lsp-comp-queried {
       description "definitons for LSPs computed by querying a remote
         service, e.g., PCE server";

       container queried-path {
         description "LSP with path queried from an external server";

         leaf path-computation-server {
           type inet:ip-address;
           description
            "Address of the external path computation
             server";
         }
       }
     }

     grouping te-lsp-comp-local {
       description "definitons for locally-computed LSPs";

       container locally-computed {
         description "LSP with path computed by local ingress LSR";

         leaf use-cspf {
           type boolean;
           description "Flag to enable CSPF for locally computed LSPs";
         }
         leaf cspf-tiebreaker {
           type cspf-tie-breaking;
           description
             "Determine the tie-breaking method to choose between
             equally desirable paths during CSFP computation";
         }
       }
     }


     grouping explicit-route-subobject {
```

```
       description
         "The explicit route subobject grouping";
       choice type {
         description
           "The explicit route subobject type";
         case ipv4-address {
           description
             "IPv4 address explicit route subobject";
           leaf address {
             type inet:ip-address;
             description "router hop for the LSP path";
           }

           leaf hop-type {
             type mpls-hop-type;
             description "strict or loose hop";
           }
         }

         case label {
           leaf value {
             type uint32;
             description "the label value";
           }
           description
             "The Label ERO subobject";
         }
       }
     }


     // Explicit paths config somewhat following the IETF model
     grouping named-explicit-path_config {
       description
         "Global explicit path configuration
         grouping";
       list named-explicit-paths {
         key "name";
         description
           "A list of explicit paths";
         leaf name {
           type string;
           description
             "A string name that uniquely identifies
             an explicit path";
         }
         list explicit-route-objects {
           key "index";
```

```
          description
            "List of explicit route objects";
          leaf index {
            type uint8 {
              range "0..255";
            }
            description
              "Index of this explicit route object,
               to express the order of hops in path";
          }
          uses explicit-route-subobject;
        }
      }
    }

    grouping explicit-paths-top {
      description
        "common information for MPLS explicit path definition";
        list explicit-path {
          key name;
          description "Explicit path definition";

          leaf name {
            type leafref {
              path "/mpls/lsps/constrained-path/"
                + "explicit-path/config/named-explicit-paths/name";
              require-instance true;
            }
            description "definition for naming an explicit  path";

          }
          container config {
            description "configuration for an explicit path";
            uses named-explicit-path_config;
          }
          container state {
            config false;
            description "operational state for LSP path name";
            uses named-explicit-path_config;
          }
        }
    }

    grouping mpls-te-srlg_config {
      description
        "Configuration of various attributes associated
        with the SRLG";
```

```
      leaf srlg-name {
        type string;
        description "SRLG group identifier";
      }

      leaf srlg-value {
        type uint32;
        description "group ID for the SRLG";
      }

      leaf srlg-cost {
        type uint32;
        description
          "The cost of the SRLG to the computation
          algorithm";
      }
    }

    grouping mpls-te-srlg-members_config {
      description "Configuration of the membership of the SRLG";

      leaf from-address {
        type inet:ip-address;
        description "IP address of the a-side of the SRLG link";
      }

      leaf to-address {
        type inet:ip-address;
        description "IP address of the z-side of the SRLG link";
      }
    }

    grouping mpls-te-srlg-top {
      description
        "Top level grouping for MPLS shared
        risk link groups.";
      container mpls-te-srlg {
        description
          "Shared risk link groups attributes";
        list srlg {
          key srlg-name;
          description "List of shared risk link groups";

          leaf srlg-name {
            type leafref {
              path "../config/srlg-name";
              require-instance true;
            }
```

```
            description "The SRLG group identifier";
          }

          container config {
            description "Configuration parameters related to the SRLG";
            uses mpls-te-srlg_config;
          }

          container state {
            config false;
            description "State parameters related to the SRLG";
            uses mpls-te-srlg_config;
          }

          list members-list {
            key from-address;
            description
              "List of SRLG members, which are expressed
              as IP address endpoints of links contained in the SRLG";

            leaf from-address {
              type leafref {
                path "../config/from-address";
                require-instance true;
              }
              description "The from address of the link in the SRLG";
            }

            container config {
              description
                "Configuration parameters relating to the
                SRLG members";
              uses mpls-te-srlg-members_config;
            }

            container state {
              config false;
              description
                "State parameters relating to the SRLG
                members";
              uses mpls-te-srlg-members_config;
            }
          }
        }
      }
    }

    grouping tunnel-path_config {
```

```
       description
         "Tunnel path properties grouping";
       container path-computation-method {
         description
           "select and configure the way the LSP path is
           computed";

         leaf path-computation {
           type identityref {
             base path-computation-method;
           }
           description "path computation method to use with the LSP";
         }

         uses te-lsp-comp-explicit;
         uses te-lsp-comp-queried;
         uses te-lsp-comp-local;
       }

       uses path-placement-constraints;

       leaf no-cspf {
         type empty;
         description
           "Indicates no CSPF is to be attempted on this
           path.";
       }

       choice sigaling-specific-path-attributes {
         description "Signaling-protocol specific path attributes.";
         case RSVP {
           uses rsvp:rsvp-p2p-path-attributes_config;
         }
         case SR {
           uses sr:sr-path-attributes_config;
         }
       }
     }

     grouping te-tunnel_config {
       description
         "Configuration parameters relevant to a single
          traffic engineered tunnel.";

       leaf name {
         type string;
         description "The tunnel name";
       }
```

```
       leaf type {
         type identityref {
           base mplst:tunnel-type;
         }
         description "Tunnel type, p2p or p2mp";
       }

       leaf local-id {
         type union {
           type uint32;
           type string;
         }
         description
           "locally signficant optional identifier for the
            tunnel; may be a numerical or string value";
       }

       leaf description {
         type string;
         description "optional text description for the tunnel";
       }

       leaf admin-status {
         type identityref {
           base mplst:tunnel-admin-status;
         }
         default mplst:ADMIN_UP;
         description "TE tunnel administrative state.";
       }

       leaf preference {
         type uint8 {
           range "1..255";
         }
         description "Specifies a preference for this tunnel.
           A lower number signifies a better preference";
       }

       uses te-tunnel-metric_config;
       uses te-tunnel-bandwidth_config;
       uses te-tunnel-protection_config;
       uses te-tunnel-reoptimize_config;

       choice signaling-specific-tunnel-attributes {
         description "Signaling-protocol specific path attributes.";
         case RSVP {
           uses rsvp:rsvp-p2p-tunnel-attributes_config;
         }
```

```
       }

       choice tunnel-type {
         description
           "Describes tunnel by type type";
         case p2p {
           leaf destination {
             type inet:ip-address;
             description
               "P2P tunnel destination address";
           }
           /* P2P list of path(s) */
           list primary-paths {
             key "name";
             leaf name {
               type string;
               description "Path name";
             }
             description
               "List of primary paths for this
               tunnel.";
             leaf preference {
               type uint8 {
                 range "1..255";
               }
               description
                 "Specifies a preference for
                 this path. The lower the
                 number higher the
                 preference";
             }
             uses tunnel-path_config;
           }

           list secondary-paths {
             key "name";
             description
               "List of secondary paths for this
                 tunnel.";
             leaf name {
               type string;
               description "Path name";
             }
             leaf preference {
               type uint8 {
                 range "1..255";
               }
               description
```

```
                 "Specifies a preference for
                   this path. The lower the
                   number higher the
                   preference";
             }
             uses tunnel-path_config;
           }
         }
         case p2mp {
           // TODO - complete
         }
       }
     }

     grouping te-tunnel_state {
       description
         "Counters and statistical data relevent to a single
          tunnel.";


       leaf oper-status {
         type identityref {
           base mplst:lsp-oper-status;
         }
         description
           "The operational status of the TE tunnel";
       }

       leaf role {
         type identityref {
           base mplst:lsp-role;
         }
         description
           "The lsp role at the current node, whether it is headend,
           transit or tailend.";
       }

       container counters {
         description
           "State data for MPLS label switched paths. This state
           data is specific to a single label switched path.";

         leaf bytes {
           type yang:counter64;
           description
             "Number of bytes that have been forwarded over the
              label switched path.";
         }
```

```
         leaf packets {
           type yang:counter64;
           description
             "Number of pacets that have been forwarded over the
              label switched path.";
         }

         leaf path-changes {
           type yang:counter64;
           description
             "Number of path changes for the label switched path";
         }

         leaf state-changes {
           type yang:counter64;
           description
             "Number of state changes for the label switched path";
         }

         leaf online-time {
           type yang:date-and-time;
           description
             "Indication of the time the label switched path
              transitioned to an Oper Up or in-service state";
         }

         leaf current-path-time {
           type yang:date-and-time;
           description
             "Indicates the time the LSP switched onto its
              current path. This is reset upon a LSP path
              change.";
         }

         leaf next-reoptimization-time {
           type yang:date-and-time;
           description
             "Indicates the next scheduled time the LSP
              will be reoptimized.";
         }
       }
     }

     grouping te-tunnels-top {
       description
         "Top level grouping for TE tunnels";

       list tunnel {
```

```
          key "name type";
          description "List of TE tunnels";

          leaf name {
            type leafref {
              path "../config/name";
              require-instance true;
            }
            description "The tunnel name";
          }

          leaf type {
            type leafref {
              path "../config/type";
              require-instance true;
            }
            description "The tunnel type, p2p or p2mp.";
          }

          container config {
            description
              "Configuration parameters related to TE tunnels:";
            uses te-tunnel_config;
          }

          container state {
            config false;
            description "State parameters related to TE interfaces";
            uses te-tunnel_config;
            uses te-tunnel_state;
          }
        }
      }

    // data definition statements

    // augment statements

    // rpc statements

    // notification statements
    }
                     <CODE ENDS>
```

6.3. MPLS signaling protocol modules

```
                    <CODE BEGINS> file openconfig-mpls-rsvp.yang
   module openconfig-mpls-rsvp {

     yang-version "1";

     // namespace
     namespace "http://openconfig.net/yang/rsvp";

     prefix "rsvp";

     // import some basic types
     import ietf-inet-types { prefix inet; }
     import openconfig-mpls-types { prefix mplst; }
     import ietf-yang-types { prefix yang; }
     import openconfig-types { prefix oc-types; }


     // meta
     organization "OpenConfig working group";

     contact
       "OpenConfig working group
        netopenconfig@googlegroups.com";

     description
       "Configuration for RSVP-TE signaling, including global protocol
        parameters and LSP-specific configuration for constrained-path
        LSPs";

     revision "2015-09-18" {
       description
         "Initial revision";
       reference "TBD";
     }

     // extension statements

     // feature statements

     // identity statements

     // typedef statements

     // grouping statements
```

```
     grouping mpls-rsvp-soft-preemption_config {
       description "Configuration for MPLS soft preemption";
       leaf enable {
         type boolean;
         default false;
         description "Enables soft preemption on a node.";
       }

       leaf soft-preemption-timeout {
         type uint16 {
           range 0..max;
         }
         // The RFC actually recommends 30 seconds as default.
         default 0;
         description
           "Timeout value for soft preemption to revert
            to hard preemption";
         reference "RFC5712 MPLS-TE soft preemption";
       }
     }

     grouping mpls-rsvp-soft-preemption {
       description "Top level group for MPLS soft preemption";
       container soft-preemption {
         description
           "Protocol options relating to RSVP
            soft preemption";
         container config {
           description
             "Configuration parameters relating to RSVP
              soft preemption support";
           uses mpls-rsvp-soft-preemption_config;
         }
         container state {
           config false;
           description
             "State parameters relating to RSVP
              soft preemption support";
           uses mpls-rsvp-soft-preemption_config;
         }
       }
     }

     grouping mpls-rsvp-hellos_config {
       description "RSVP protocol options configuration.";

       leaf hello-interval {
         type uint16 {
```

```
          range 1000..60000;
        }
        units milliseconds;
        default 9000;
        description
          "set the interval in ms between RSVP hello
           messages";
        reference
          "RFC 3209: RSVP-TE: Extensions to RSVP for
           LSP Tunnels.
           RFC 5495: Description of the Resource
           Reservation Protocol - Traffic-Engineered
           (RSVP-TE) Graceful Restart Procedures";
      }

      leaf refresh-reduction {
        type boolean;
        default true;
        description
          "enables all RSVP refresh reduction message
           bundling, RSVP message ID, reliable message delivery
           and summary refresh";
        reference
          "RFC 2961 RSVP Refresh Overhead Reduction
           Extensions";
      }
    }

    grouping mpls-rsvp-hellos {
      description "Top level grouping for RSVP hellos parameters";
      // TODO: confirm that the described semantics are supported
      // on various implementations. Finer grain configuration
      // will be vendor-specific

      container rsvp-hellos {
        description "Top level container for RSVP hello parameters";
        container config {
          description
            "Configuration parameters relating to RSVP
             hellos";
          uses mpls-rsvp-hellos_config;
        }
        container state {
          config false;
          description "State information associated with RSVP hellos";
          uses mpls-rsvp-hellos_config;
        }
      }
```

```
    }

    grouping mpls-rsvp-subscription_config {
      description "RSVP subscription configuration";
      leaf subscription {
        type oc-types:percentage;
        description
          "percentage of the interface bandwidth that
           RSVP can reserve";
      }
    }
    grouping mpls-rsvp-subscription {
      description "Top level group for RSVP subscription options";
      container subscription {
        description
          "Bandwidth percentage reservable by RSVP
           on an interface";
        container config {
          description
            "Configuration parameters relating to RSVP
             subscription options";
          uses mpls-rsvp-subscription_config;
        }
        container state {
          config false;
          description
          "State parameters relating to RSVP
             subscription options";
          uses mpls-rsvp-subscription_config;
        }
      }
    }

    grouping mpls-rsvp-graceful-restart_config {
      description
        "Configuration parameters relating to RSVP Graceful-Restart";

      leaf enable {
        type boolean;
        default false;
        description "Enables graceful restart on the node.";
      }

      leaf restart-time {
        type uint32;
        description
          "Graceful restart time (seconds).";
        reference
```

```
            "RFC 5495: Description of the Resource
             Reservation Protocol - Traffic-Engineered
             (RSVP-TE) Graceful Restart Procedures";
        }
        leaf recovery-time {
          type uint32;
          description
            "RSVP state recovery time";
        }
      }

      grouping mpls-rsvp-graceful-restart {
        description
          "Top level group for RSVP graceful-restart
           parameters";
        container graceful-restart {
          description "TODO";
          container config {
            description
              "Configuration parameters relating to
               graceful-restart";
            uses mpls-rsvp-graceful-restart_config;
          }
          container state {
            config false;
            description
              "State information associated with
               RSVP graceful-restart";
            uses mpls-rsvp-graceful-restart_config;
          }
        }
      }

      grouping mpls-rsvp-authentication_config {
        description "RSVP authentication parameters container.";
        leaf enable {
          type boolean;
          default false;
          description "Enables RSVP authentication on the node.";
        }
        leaf authentication-key {
          type string {
            // Juniper supports 1..16 while
            // Cisco has a much bigger range, up to 60.
            length "1..32";
          }
          description
            "authenticate RSVP signaling
```

```
            messages";
         reference
           "RFC 2747: RSVP Cryptographic Authentication";
       }
     }

     grouping mpls-rsvp-authentication {
       description
         "Top level group for RSVP authentication,
          as per RFC2747";
       container authentication {
         description "TODO";
         container config {
           description
             "Configuration parameters relating
              to authentication";
           uses mpls-rsvp-authentication_config;
         }
         container state {
           config false;
           description
             "State information associated
              with authentication";
           uses mpls-rsvp-authentication_config;
         }
       }
     }

     grouping mpls-rsvp-protection_config {
       description "RSVP facility (link/node) protection configuration";

       leaf link-protection-style-requested {
         type identityref {
           base mplst:protection-type;
         }
         default mplst:link-node-protection-requested;
         description
           "style of mpls frr protection desired:
           link, link-node, or unprotected";
       }

       leaf bypass-optimize-interval {
         type uint16;
         units seconds;
         description
           "interval between periodic optimization
           of the bypass LSPs";
         // note: this is interface specific on juniper
```

```
        // on iox, this is global. need to resolve.
      }
      // to be completed, things like enabling link protection,
      // optimization times, etc.
    }

    grouping mpls-rsvp-link-protection {
      description "Top level group for RSVP protection";
      container protection {
        description "link-protection (NHOP) related configuration";
        container config {
          description "Configuration for link-protection";
          uses mpls-rsvp-protection_config;
        }
        container state {
          config false;
          description "State for link-protection";
          uses mpls-rsvp-protection_config;
        }
      }
    }

    grouping mpls-rsvp-error-statistics {
      description "RSVP-TE packet statistics";
      container error {
        description "RSVP-TE error statistics";
        leaf authentication-failure {
          type yang:counter32;
          description
            "Authentication failure count";
        }

        leaf path-error {
          type yang:counter32;
          description
            "Path error to client count";
        }

        leaf resv-error {
          type yang:counter32;
          description
            "Resv error to client count";
        }

        leaf path-timeout {
          type yang:counter32;
          description
            "Path timeout count";
```

```
          }

          leaf resv-timeout {
            type yang:counter32;
            description
              "Resv timeout count";
          }

          leaf rate-limit {
            type yang:counter32;
            description
              "Count of packets that were rate limited";
          }

          // TODO - complete the other error statistics
        }
      }

      grouping mpls-rsvp-protocol-statistics {
        description "RSVP protocol statistics";
        container protocol {
          description "RSVP-TE protocol statistics";
          leaf hello-sent {
            type yang:counter32;
            description
              "Hello sent count";
          }

          leaf hello-rcvd {
            type yang:counter32;
            description
              "Hello received count";
          }

          leaf path-sent {
            type yang:counter32;
            description
              "Path sent count";
          }

          leaf path-rcvd {
            type yang:counter32;
            description
              "Path received count";
          }

          // TODO - To be completed the other packet statistics
        }
```

```
     }

     grouping mpls-rsvp-statistics {
       description "Top level grouping for RSVP protocol state";
       uses mpls-rsvp-protocol-state;
     }

     grouping rsvp-global {
       description "Global RSVP protocol configuration";
       container rsvp-te {
         description "RSVP-TE global signaling protocol configuration";

         container rsvp-sessions {
           description "Configuration and state of RSVP sessions";

           container config {
             description
               "Configuration of RSVP sessions on the device";
           }

           container state {
             config false;
             description
               "State information relating to RSVP sessions
               on the device";
             uses mpls-rsvp-session-state;
           }
         }

         container rsvp-neighbors {
           description
             "Configuration and state for RSVP neighbors connecting
             to the device";

           container config {
             description "Configuration of RSVP neighbor information";
           }

           container state {
             config false;
             description
               "State information relating to RSVP neighbors";
             uses mpls-rsvp-neighbor-state;
           }
         }

         container global {
           description "Platform wide RSVP configuration and state";
```

```
          uses mpls-rsvp-graceful-restart;
          uses mpls-rsvp-soft-preemption;

          container statistics {
            config false;
            description "Platform wide RSVP state, including counters";
            // TODO - reconcile global and per-interface
            // protocol-related statistics

            container counters {
              config false;
              description
                "Platform wide RSVP statistics and counters";
              uses mpls-rsvp-global-protocol-state;
              uses mpls-rsvp-statistics;
            }
          }
        }

        container interface-attributes {
          // interfaces, bw percentages, hello timers, etc goes here";

          list interface {
            key interface-name;
            description "list of per-interface RSVP configurations";

            // TODO: update to interface ref -- move to separate
            // augmentation.
            leaf interface-name {
              type leafref {
                path "../config/interface-name";
                require-instance true;
              }
              description "references a configured IP interface";
            }


            container config {
              description
                "Configuration of per-interface RSVP parameters";

              leaf interface-name {
                type string;
                description "Name of configured IP interface";
              }
            }

            container state {
```

```
              config false;
              description
                "Per-interface RSVP protocol and state information";
              uses mpls-rsvp-interfaces-state;

              container counters {
                config false;
                description
                  "Interface specific RSVP statistics and counters";
                uses mpls-rsvp-protocol-state;
              }
            }

            uses mpls-rsvp-hellos;
            uses mpls-rsvp-authentication;
            uses mpls-rsvp-subscription;
            uses mpls-rsvp-link-protection;
          }
        }
      }
    }

    grouping rsvp-p2p-tunnel-attributes_config {
      description "properties of RSPP point-to-point paths";

     leaf source {
       type inet:ip-address;
       description
         "tunnel source address";
     }

     leaf soft-preemption {
        type boolean;
        default false;
        description "enables RSVP soft-preemption on this LSP";
      }
    }


    grouping rsvp-p2p-path-attributes_config {
      description "properties of RSPP point-to-point paths";
      leaf setup-priority {
        type uint8 {
          range 0..7;
        }
        default 7;
        description
          "preemption priority during LSP setup, lower is
```

```
            higher priority; default 7 indicates that LSP will not
            preempt established LSPs during setup";
         reference "RFC 3209 - RSVP-TE: Extensions to RSVP for
           LSP Tunnels";
       }

       leaf hold-priority {
         type uint8 {
           range 0..7;
         }
         default 0;
         description
           "preemption priority once the LSP is established,
            lower is higher priority; default 0 indicates other LSPs
            will not preempt the LSPs once established";
         reference "RFC 3209 - RSVP-TE: Extensions to RSVP for
           LSP Tunnels";
       }

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

     grouping mpls-rsvp-neighbor-state {
       description "State information for RSVP neighbors";

       list rsvp-neighbor {
         key "neighbor-address";
         description
           "List of RSVP neighbors connecting to the device,
           keyed by neighbor address";

         leaf neighbor-address {
           type inet:ip-address;
           description "Address of RSVP neighbor";
         }

         leaf detected-interface {
           type string;
           description "Interface where RSVP neighbor was detected";
         }
```

```
         leaf neighbor-status {
           type enumeration {
             enum UP {
               description
                 "RSVP hello messages are detected from the neighbor";
             }
             enum DOWN {
               description
                 "RSVP neighbor not detected as up, due to a
                  communication failure or IGP notification
                  the neighbor is unavailable";
             }
           }
           description "Enumuration of possible RSVP neighbor states";
         }

         leaf neighbor-refresh-reduction {
           type boolean;
           description
             "Suppport of neighbor for RSVP refresh reduction";
           reference
             "RFC 2961 RSVP Refresh Overhead Reduction
              Extensions";
         }
       }
     }

     grouping mpls-rsvp-session-state {
       description "State information for RSVP TE sessions";
       list rsvp-session {
         key "source-port destination-port
          source-address destination-address";
         description "List of RSVP sessions";

         leaf source-address {
           type inet:ip-address;
           description "Origin address of RSVP session";
         }

         leaf destination-address {
           type inet:ip-address;
           description "Destination address of RSVP session";
         }

         leaf source-port {
           type uint16;
           description "RSVP source port";
           reference "RFC 2205";
```

```
         }

         leaf destination-port {
           type uint16;
           description "RSVP source port";
           reference "RFC 2205";

         }

         leaf session-state {
           type enumeration {
             enum UP {
               description "RSVP session is up";
             }
             enum DOWN {
               description "RSVP session is down";
             }
           }
           description "Enumeration of RSVP session states";
         }

         leaf session-type {
           type enumeration {
             enum SOURCE {
               description "RSVP session originates on this device";
             }
             enum TRANSIT {
               description "RSVP session transits this device only";
             }
             enum DESTINATION {
               description "RSVP session terminates on this device";
             }
           }
           description "Enumeration of possible RSVP session types";
         }

         leaf tunnel-id {
           type uint16;
           description "Unique identifier of RSVP session";
         }

         leaf label-in {
           type mplst:mpls-label;
           description
             "Incoming MPLS label associated with this RSVP session";
         }

         leaf label-out {
```

```
          type mplst:mpls-label;
          description
            "Outgoing MPLS label associated with this RSVP session";
        }

        leaf-list associated-lsps {
          type leafref {
            path "/mpls/lsps/constrained-path/tunnel/" +
            "config/name";
          }
          description
            "List of label switched paths associated with this RSVP
             session";
        }
      }
    } //rsvp-session-state

    grouping mpls-rsvp-interfaces-state {
      description "RSVP state information relevant to an interface";

      list bandwidth {
        key priority;
        description
          "Available and reserved bandwidth by priority on
           the interface.";

        leaf priority {
          type uint8 {
            range 0..7;
          }
          description
            "RSVP priority level for LSPs traversing the interface";
        }

        leaf available-bandwidth {
          type mplst:bandwidth-mbps;
          description "Bandwidth currently available";
        }

        leaf reserved-bandwidth {
          type mplst:bandwidth-mbps;
          description "Bandwidth currently reserved";
        }
      }

      leaf highwater-mark {
        type mplst:bandwidth-mbps;
        description "Maximum bandwidth ever reserved";
```

```
       }

       leaf active-reservation-count {
         type yang:gauge64;
         description "Number of active RSVP reservations";
       }
     }

     grouping mpls-rsvp-global-protocol-state {
       description "RSVP protocol statistics which may not apply
         on an interface, but are significant globally.";

       leaf path-timeouts {
         type yang:counter64;
         description "TODO";
       }

       leaf reservation-timeouts {
         type yang:counter64;
         description "TODO";
       }

       leaf rate-limited-messages {
         type yang:counter64;
         description "RSVP messages dropped due to rate limiting";
       }
     }

     grouping mpls-rsvp-protocol-state {
       description "RSVP protocol statistics and message counters";
       leaf in-path-messages {
         type yang:counter64;
         description "Number of received RSVP Path messages";
       }

       leaf in-path-error-messages {
         type yang:counter64;
         description "Number of received RSVP Path Error messages";
       }

       leaf in-path-tear-messages {
         type yang:counter64;
         description "Number of received RSVP Path Tear messages";
       }

       leaf in-reservation-messages {
         type yang:counter64;
         description "Number of received RSVP Resv messages";
```

```
        }

        leaf in-reservation-error-messages {
          type yang:counter64;
          description "Number of received RSVP Resv Error messages";
        }

        leaf in-reservation-tear-messages {
          type yang:counter64;
          description "Number of received RSVP Resv Tear messages";
        }

        leaf in-rsvp-hello-messages {
          type yang:counter64;
          description "Number of received RSVP hello messages";
        }

        leaf in-rsvp-srefresh-messages {
          type yang:counter64;
          description "Number of received RSVP summary refresh messages";
        }

        leaf in-rsvp-ack-messages {
          type yang:counter64;
          description
            "Number of received RSVP refresh reduction ack
             messages";
        }

        leaf out-path-messages {
          type yang:counter64;
          description "Number of sent RSVP PATH messages";
        }

        leaf out-path-error-messages {
          type yang:counter64;
          description "Number of sent RSVP Path Error messages";
        }

        leaf out-path-tear-messages {
          type yang:counter64;
          description "Number of sent RSVP Path Tear messages";
        }

        leaf out-reservation-messages {
          type yang:counter64;
          description "Number of sent RSVP Resv messages";
        }
```

```
        leaf out-reservation-error-messages {
          type yang:counter64;
          description "Number of sent RSVP Resv Error messages";
        }

        leaf out-reservation-tear-messages {
          type yang:counter64;
          description "Number of sent RSVP Resv Tear messages";
        }

        leaf out-rsvp-hello-messages {
          type yang:counter64;
          description "Number of sent RSVP hello messages";
        }

        leaf out-rsvp-srefresh-messages {
          type yang:counter64;
          description "Number of sent RSVP summary refresh messages";
        }

        leaf out-rsvp-ack-messages {
          type yang:counter64;
          description
            "Number of sent RSVP refresh reduction ack messages";
        }
      }


      // data definition statements

      // augment statements

      // rpc statements

      // notification statements

    }
                     <CODE ENDS>



                     <CODE BEGINS> file openconfig-mpls-sr.yang
    module openconfig-mpls-sr {

      yang-version "1";

      // namespace
```

```
namespace "http://openconfig.net/yang/sr";

prefix "sr";


// import some basic types
import ietf-inet-types { prefix inet; }
import openconfig-mpls-types { prefix mplst; }

// meta
organization "OpenConfig working group";

contact
  "OpenConfig working group
  netopenconfig@googlegroups.com";

description
  "Configuration for MPLS with segment routing-based LSPs,
  including global parameters, and LSP-specific configuration for
  both constrained-path and IGP-congruent LSPs";

revision "2015-10-14" {
  description
    "Work in progress";
  reference "TBD";
}

// extension statements

// feature statements

// identity statements

// typedef statements

grouping srgb_config {

  // Matches the "global" configuration options in
  // draft-litkowski-spring-yang...
  // TODO: request to Stephane for this to be a separate
  // grouping such that it can be included.

  leaf lower-bound {
    type uint32;
    description
      "Lower value in the block.";
  }
  leaf upper-bound {
```

```
          type uint32;
          description
            "Upper value in the block.";
        }
        description
          "List of global blocks to be advertised.";
      }

      grouping srgb_state {
        description
          "State parameters relating to the SRGB";

        leaf size {
          type uint32;
          description
            "Number of indexes in the SRGB block";
        }
        leaf free {
          type uint32;
          description
            "Number of SRGB indexes that have not yet been allocated";
        }
        leaf used {
          type uint32;
          description
            "Number of SRGB indexes that are currently allocated";
        }

        // TODO: where do we put LFIB entries?

      }

      grouping adjacency-sid_config {
        description
          "Configuration related to an Adjacency Segment Identifier
          (SID)";

        // tuned from draft-litkowski-spring-yang
        // TODO: need to send a patch to Stephane

        leaf-list advertise {
          type enumeration {
            enum "PROTECTED" {
              description
                "Advertise an Adjacency-SID for this interface, which is
                eligible to be protected using a local protection
                mechanism on the local LSR. The local protection
                mechanism selected is dependent upon the configuration
```

```
              of RSVP-TE FRR or LFA elsewhere on the system";
          }
          enum UNPROTECTED {
            description
              "Advertise an Adajcency-SID for this interface, which is
              explicitly excluded from being protected by any local
              protection mechanism";
          }
        }
        description
          "Specifies the type of adjacency SID which should be
          advertised for the specified entity.";
      }

      leaf-list groups {
        type uint32;
        description
          "Specifies the groups to which this interface belongs.
          Setting a value in this list results in an additional AdjSID
          being advertised, with the S-bit set to 1. The AdjSID is
          assumed to be protected";
      }
    }

    grouping interface_config {
      description
        "Configuration parameters relating to a Segment Routing
        enabled interface";

      leaf interface {
        type string;
        // TODO: this should be changed to a leafref.
        description
          "Reference to the interface for which segment routing
          configuration is to be applied.";
      }
    }

    // grouping statements

    grouping sr-global {
      description "global segment routing signaling configuration";

      container segment-routing {
        description "SR global signaling config";

        list srgb {
          key "lower-bound upper-bound";
```

```
          uses srgb_config;
          container config {
            description
              "Configuration parameters relating to the Segment Routing
              Global Block (SRGB)";
            uses srgb_config;
          }
          container state {
            config false;
            description
              "State parameters relating to the Segment Routing Global
              Block (SRGB)";
            uses srgb_config;
            uses srgb_state;
          }
          description
            "List of Segment Routing Global Block (SRGB) entries. These
            label blocks are reserved to be allocated as domain-wide
            entries.";
        }

        list interfaces {
          key "interface";
          uses interface_config;
          container config {
            description
              "Interface configuration parameters for Segment Routing
              relating to the specified interface";
            uses interface_config;
          }
          container state {
            config false;
            description
              "State parameters for Segment Routing features relating
              to the specified interface";
            uses interface_config;
          }
          container adjacency-sid {
            description
              "Configuration for Adjacency SIDs that are related to
              the specified interface";
            container config {
              description
                "Configuration parameters for the Adjacency-SIDs
                that are related to this interface";
              uses adjacency-sid_config;
            }
            container state {
```

```
              config false;
              description
                "State parameters for the Adjacency-SIDs that are
                related to this interface";
              uses adjacency-sid_config;
            }
          }
          description
            "List of interfaces with associated segment routing
            configuration";
        }
      }
    }

    grouping sr-path-attributes_config {
      description
        "Configuration parameters relating to SR-TE LSPs";

      leaf sid-selection-mode {
        type enumeration {
          enum "ADJ-SID-ONLY" {
            description
              "The SR-TE tunnel should only use adjacency SIDs
              to build the SID stack to be pushed for the LSP";
          }
          enum "MIXED-MODE" {
            description
              "The SR-TE tunnel can use a mix of adjacency
              and prefix SIDs to build the SID stack to be pushed
              to the LSP";
          }
        }
        default "MIXED-MODE";
        description
          "The restrictions placed on the SIDs to be selected by the
          calculation method for the SR-TE LSP";
      }

      leaf sid-protection-required {
        type boolean;
        default "false";
        description
          "When this value is set to true, only SIDs that are
          protected are to be selected by the calculating method
          for the SR-TE LSP.";
      }
    }
```

```
     grouping sr_fec-address_config {
       description
         "Configuration parameters relating to a FEC that is to be
         advertised by Segment Routing";

       leaf fec-address {
         type inet:ip-prefix;
         description
           "FEC that is to be advertised as part of the Prefix-SID";
       }
     }

     grouping sr_fec-prefix-sid_config {
       description
         "Configuration parameters relating to the nature of the
         Prefix-SID that is to be advertised for a particular FEC";

       leaf type {
         type enumeration {
           enum "INDEX" {
             description
               "Set when the value of the prefix SID should be specified
               as an off-set from the SRGB’s zero-value. When multiple
               SRGBs are specified, the zero-value is the minimum
               of their lower bounds";
           }
           enum "ABSOLUTE" {
             description
               "Set when the value of a prefix SID is specified as the
               absolute value within an SRGB. It is an error to specify
               an absolute value outside of a specified SRGB";
           }
         }
         default "INDEX";
         description
           "Specifies how the value of the Prefix-SID should be
           interpreted - whether as an offset to the SRGB, or as an
           absolute value";
       }

       leaf node-flag {
         type boolean;
         description
           "Specifies that the Prefix-SID is to be treated as a Node-SID
           by setting the N-flag in the advertised Prefix-SID TLV in the
           IGP";
       }
```

```
        leaf last-hop-behavior {
          type enumeration {
            enum "EXPLICIT-NULL" {
              description
                "Specifies that the explicit null label is to be used
                when the penultimate hop forwards a labelled packet to
                this Prefix-SID";
            }
            enum "UNCHANGED" {
              description
                "Specicies that the Prefix-SID's label value is to be
                left in place when the penultimate hop forwards to this
                Prefix-SID";
            }
            enum "PHP" {
              description
                "Specicies that the penultimate hop should pop the
                Prefix-SID label before forwarding to the eLER";
            }
          }
          description
            "Configuration relating to the LFIB actions for the
            Prefix-SID to be used by the penultimate-hop";
        }
      }


      grouping igp-tunnel-sr {
        description "defintiions for SR-signaled, IGP-based LSP tunnel
        types";

        container tunnel {
          description "contains configuration stanzas for different LSP
          tunnel types (P2P, P2MP, etc.)";

          leaf tunnel-type {
            type mplst:tunnel-type;
            description "specifies the type of LSP, e.g., P2P or P2MP";
          }

          container p2p-lsp {
            when "tunnel-type = 'P2P'" {
              description "container active when LSP tunnel type is
              point to point";
            }
            description "properties of point-to-point tunnels";

            list fec {
```

```
            key "fec-address";
            uses sr_fec-address_config;

            description
              "List of FECs that are to be originated as SR LSPs";

            container config {
              description
                "Configuration parameters relating to the FEC to be
                advertised by SR";
              uses sr_fec-address_config;
            }
            container state {
              config false;
              description
                "Operational state relating to a FEC advertised by SR";
              uses sr_fec-address_config;
            }
            container prefix-sid {
              description
                "Parameters relating to the Prefix-SID
                used for the originated FEC";

              container config {
                description
                  "Configuration parameters relating to the Prefix-SID
                  used for the originated FEC";
                uses sr_fec-prefix-sid_config;
              }
              container state {
                config false;
                description
                  "Operational state parameters relating to the
                  Prefix-SID used for the originated FEC";
                uses sr_fec-prefix-sid_config;
              }
            }
          }
        }
      }
    }

    grouping igp-lsp-sr-setup {
      description "grouping for SR-IGP path setup for IGP-congruent
      LSPs";

      container segment-routing {
```

```
          presence "Presence of this container sets the LSP to use
          SR signaling";

          description "segment routing signaling extensions for
          IGP-confgruent LSPs";

          uses igp-tunnel-sr;

        }
      }

      // data definition statements

      // augment statements

      // rpc statements

      // notification statements

   }
                    <CODE ENDS>



                    <CODE BEGINS> file openconfig-mpls-ldp.yang
   module openconfig-mpls-ldp {

     yang-version "1";

     // namespace
     namespace "http://openconfig.net/yang/ldp";

     prefix "ldp";

     // import some basic types
     import ietf-inet-types { prefix inet; }
     import openconfig-mpls-types { prefix mplst; }


     // meta
     organization "OpenConfig working group";

     contact
       "OpenConfig working group
       netopenconfig@googlegroups.com";

     description
       "Configuration of Label Distribution Protocol global and LSP-
```

```
    specific parameters for IGP-congruent LSPs";

  revision "2015-07-04" {
    description
      "Initial revision";
    reference "TBD";
  }

  // extension statements

  // feature statements

  // identity statements

  // typedef statements

  // grouping statements

   grouping ldp-global {
    description "global LDP signaling configuration";

    container ldp {
      description "LDP global signaling configuration";

      container timers {
        description "LDP timers";
      }
    }
  }


  grouping igp-tunnel-ldp {
    description "common defintiions for LDP-signaled LSP tunnel
    types";

    container tunnel {
      description "contains configuration stanzas for different LSP
      tunnel types (P2P, P2MP, etc.)";

      leaf tunnel-type {
        type mplst:tunnel-type;
          description "specifies the type of LSP, e.g., P2P or P2MP";
      }

      leaf ldp-type {
        type enumeration {
          enum BASIC {
            description "basic hop-by-hop LSP";
```

```
            }
            enum TARGETED {
              description "tLDP LSP";
            }
          }
          description "specify basic or targeted LDP LSP";
        }

        container p2p-lsp {
          when "tunnel-type = 'P2P'" {
            description "container active when LSP tunnel type is
            point to point";
          }

          description "properties of point-to-point tunnels";

          leaf-list fec-address {
            type inet:ip-prefix;
            description "Address prefix for packets sharing the same
            forwarding equivalence class for the IGP-based LSP";
          }
        }

        container p2mp-lsp {
          when "tunnel-type = 'P2MP'" {
            description "container is active when LSP tunnel type is
            point to multipoint";
          }

          description "properties of point-to-multipoint tunnels";

          // TODO: specify group/source, etc.

        }

        container mp2mp-lsp {
          when "tunnel-type = 'MP2MP'" {
            description "container is active when LSP tunnel type is
            multipoint to multipoint";
          }

          description "properties of multipoint-to-multipoint tunnels";

          // TODO: specify group/source, etc.

        }
      }
    }
```

```
    grouping igp-lsp-ldp-setup {
      description "grouping for LDP setup attributes";

      container ldp {

        presence "Presence of this container sets the LSP to use
        LDP signaling";

        description "LDP signaling setup for IGP-congruent LSPs";

        // include tunnel (p2p, p2mp, ...)

        uses igp-tunnel-ldp;

      }
    }

    // data definition statements

    // augment statements

    // rpc statements

    // notification statements

  }
                   <CODE ENDS>



                   <CODE BEGINS> file openconfig-mpls-igp.yang
  submodule openconfig-mpls-igp {

    yang-version "1";

    belongs-to "openconfig-mpls" {
      prefix "mpls";
    }


    // import some basic types
    import openconfig-mpls-ldp { prefix ldp; }
    import openconfig-mpls-sr { prefix sr; }



    // meta
    organization "OpenConfig working group";
```

```
   contact
     "OpenConfig working group
     netopenconfig@googlegroups.com";

   description
     "Configuration generic configuration parameters for IGP-congruent
     LSPs";

   revision "2015-07-04" {
     description
       "Initial revision";
     reference "TBD";
   }

   // extension statements

   // feature statements

   // identity statements

   // typedef statements

   // grouping statements


   grouping igp-lsp-common {
     description "common definitions for IGP-congruent LSPs";

     // container path-attributes {
     //  description "general path attribute settings for IGP-based
     //  LSPs";

     //}

   }


   grouping igp-lsp-setup {
     description "signaling protocol definitions for IGP-based LSPs";

     container path-setup-protocol {
       description "select and configure the signaling method for
           the LSP";

       // uses path-setup-common;
       uses ldp:igp-lsp-ldp-setup;
       uses sr:igp-lsp-sr-setup;
     }
```

```
     }


     // data definition statements

     // augment statements

     // rpc statements

     // notification statements

   }
                    <CODE ENDS>



                    <CODE BEGINS> file openconfig-mpls-static.yang
   submodule openconfig-mpls-static {

     yang-version "1";

     belongs-to "openconfig-mpls" {
       prefix "mpls";
     }

     // import some basic types
     import openconfig-mpls-types {prefix mplst; }
     import ietf-inet-types { prefix inet; }


     // meta
     organization "OpenConfig working group";

     contact
       "OpenConfig working group
       netopenconfig@googlegroups.com";

     description
       "Defines static LSP configuration";

     revision "2015-07-04" {
       description
         "Initial revision";
       reference "TBD";
     }

     // extension statements
```

```
   // feature statements

   // identity statements

   // typedef statements

   // grouping statements

   grouping static-lsp-common {
     description "common definitions for static LSPs";

     leaf next-hop {
       type inet:ip-address;
       description "next hop IP address for the LSP";
     }

     leaf incoming-label {
       type mplst:mpls-label;
       description "label value on the incoming packet";
     }

     leaf push-label {
       type mplst:mpls-label;
       description "label value to push at the current hop for the
       LSP";
     }
   }

   grouping static-lsp-main {
     description "grouping for top level list of static LSPs";


     list label-switched-path {
       key name;
       description "list of defined static LSPs";

       leaf name {
         type string;
         description "name to identify the LSP";
       }

       // TODO: separation into ingress, transit, egress may help
       // to figure out what exactly is configured, but need to
       // consider whether implementations can support the
       // separation
       container ingress {
         description "Static LSPs for which the router is an
           ingress node";
```

```
          uses static-lsp-common;
        }

        container transit {
          description "static LSPs for which the router is a
            transit node";

          uses static-lsp-common;
        }

        container egress {
          description "static LSPs for which the router is a
            egress  node";

          uses static-lsp-common;
        }
      }
    }

    // data definition statements

    // augment statements

    // rpc statements

    // notification statements

  }
                  <CODE ENDS>
```

7. Contributing Authors

The following people contributed significantly to this document and are listed below:

Ina Minei
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US
Email: inaminei@google.com

Anees Shaikh
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: aashaikh@google.com

Phil Bedard
Cox Communications
Atlanta, GA 30319
US
Email: phil.bedard@cox.com

8. Acknowledgements

The authors are grateful for valuable contributions to this document and the associated models from: Ebben Aires, Deepak Bansal, Nabil Bitar, Feihong Chen, Mazen Khaddam.

9. References

[I-D.ietf-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Alex, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-00 (work in progress), July 2015.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., rjs@rob.sh, r., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-02 (work in progress), October 2015.

[I-D.openconfig-netmod-model-structure]
Shaikh, A., Shakir, R., D'Souza, K., and L. Fang, "Operational Structure and Organization of YANG Models", draft-openconfig-netmod-model-structure-00 (work in progress), March 2015.

[I-D.openconfig-netmod-opstate]
Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling of Operational State Data in YANG", draft-openconfig-netmod-opstate-01 (work in progress), July 2015.

[I-D.shaikh-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Alex, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-shaikh-idr-bgp-model-02 (work in progress), June 2015.

[I-D.shakir-rtgwg-sr-performance-engineered-lsps]
Shakir, R., Vernals, D., and A. Capello, "Performance Engineered LSPs using the Segment Routing Data-Plane", draft-shakir-rtgwg-sr-performance-engineered-lsps-00 (work in progress), July 2013.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC5443] Jork, M., Atlas, A., and L. Fang, "LDP IGP Synchronization", RFC 5443, DOI 10.17487/RFC5443, March 2009, <http://www.rfc-editor.org/info/rfc5443>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

Authors' Addresses

Joshua George
Google
1600 Amphitheatre Pkwy
Mountain View, CA 94043
US

Email: jgeorge@google.com

Luyuan Fang
Microsoft
15590 NE 31st St
Redmond, WA 98052
US

Email: lufang@microsoft.com

Eric Osborne
Level 3

Email: eric.osborne@level3.com

Rob Shakir
Jive Communications, Inc.
1275 West 1600 North, Suite 100
Orem, UT 84057

Email: rjs@rob.sh

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

Quintin Zhao
Robin Li
Huawei Technologies
King Ke
Tencent Holdings Ltd.
Luyuan Fang
Microsoft
Chao Zhou
Cisco Systems
Boris Zhang
Telus Communications
July 6, 2015

The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs
draft-zhao-pce-central-controller-user-cases-02

Abstract

In certain networks deployment scenarios, service providers would like to keep all the existing MPLS functionalities in both MPLS and GMPLS network while removing the complexity of existing signaling protocols such as LDP and RSVP-TE. In this document, we propose to use the PCE as a central controller so that LSP can be calculated/signaled/initiated/downloaded/managed through a centralized PCE server to each network devices along the LSP path while leveraging the existing PCE technologies as much as possible.

This draft describes the use cases for using the PCE as the central controller where LSPs are calculated/setup/initiated/downloaded/maintained through extending the current PCE architectures and extending the PCEP.

Status of This Memo

This Internet-Draft will expire on January 7, 2016.

Copyright Notice

Table of Contents

1. Introduction

1.1. Background

In certain network deployment scenarios, service providers would like to have the ability to dynamically adapt to a wide range of customer's requests for the sake of flexible network service delivery, SDN has provides additional flexibility in how the network is operated comparing the traditional network.

The existing networking ecosystem has become awfully complex and highly demanding in terms of robustness, performance, scalability, flexibility, agility, etc. By migrating to the SDN enabled network from the existing network, service providers and network operators must have a solution which they can evolve easily from the existing network into the SDN enabled network while keeping the network services remain scalable, guarantee robustness and availability etc.

Taking the smooth transition between traditional network and the new SDN enabled network into account, especially from a cost impact assessment perspective, using the existing PCE components from the current network to function as the central controller of the SDN network is one choice, which not only achieves the goal of having a centralized controller to provide the functionalities needed for the central controller, but also leverages the existing PCE network components.

The Path Computation Element communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform route computations in response to Path Computation Clients (PCCs) requests. PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model draft [I-D. draft-ietf-pce- stateful-pce] describes a set of extensions to PCEP to enable active control of MPLS-TE and GMPLS tunnels.

[I-D.crabbe-pce-pce-initiated-lsp] describes the setup and teardown of PCE-initiated LSPs under the active stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic MPLS network that is centrally controlled and deployed.

[I-D.ali-pce-remote-initiated-gmpls-lsp] complements [I-D. draft-crabbe-pce-pce-initiated-lsp] by addressing the requirements for remote-initiated GMPLS LSPs.

SR technology leverages the source routing and tunneling paradigms. A source node can choose a path without relying on hop-by-hop signaling protocols such as LDP or RSVP-TE. Each path is specified as a set of "segments" advertised by link-state routing protocols

(IS-IS or OSPF). [I-D.filsfils-spring-segment-routing] provides an introduction to SR technology. The corresponding IS-IS and OSPF extensions are specified in [I-D.ietf-isis-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-extensions], respectively.

A Segment Routed path (SR path) can be derived from an IGP Shortest Path Tree (SPT). Segment Routed Traffic Engineering paths (SR-TE paths) may not follow IGP SPT. Such paths may be chosen by a suitable network planning tool and provisioned on the source node of the SR-TE path.

It is possible to use a stateful PCE for computing one or more SR-TE paths taking into account various constraints and objective functions. Once a path is chosen, the stateful PCE can instantiate an SR-TE path on a PCC using PCEP extensions specified in [I-D.crabbe-pce-pce-initiated-lsp] using the SR specific PCEP extensions described in [I-D.sivabalan-pce-segment-routing].

By using the solutions provided from above drafts, LSP in both MPLS and GMPLS network can be setup/delete/maintained/synchronized through a centrally controlled dynamic MPLS network. Since in these solutions, the LSP is need to be signaled through the head end LER to the tail end LER, there are either RSVP-TE signaling protocol need to be deployed in the MPLS/GMPLS network, or extend TGP protocol with node/adjacency segment identifiers signaling capability to be deployed.

The PCECC solution proposed in this document allow for a dynamic MPLS network that is eventually controlled and deployed without the deployment of RSVP-TE protocol or extended IGP protocol with node/adjacency segment identifiers signaling capability while providing all the key MPLS functionalities needed by the service providers. These key MPLS features include MPLS P2P LSP, P2MP/MP2MP LSP, MPLS protection mechanism etc. In the case that one LSP path consists legacy network nodes and the new network nodes which are centrally controlled, the PCECC solution provides a smooth transition step for users.

1.2. Using the PCE as the Central Controller (PCECC) Approach

With PCECC, it not only removes the existing MPLS signaling totally from the control plane without losing any existing MPLS functionalities, but also PCECC achieves this goal through utilizing the existing PCEP without introducing a new protocol into the network.

The following diagram illustrates the PCECC architecture.

```
+------------------------------------------------------------------+
|                         PCECC                                    |
|    +--------------------------------------------------------+    |
|    |  LSP-Database    RSVP-TE Signal Control Module          |    |
|    |  TE-Database     LDP signaling Control Module           |    |
|    |  Label-Database  LSP/label/TE MGRs                      |    |
|    +--------------------------------------------------------+    |
|    ^              ^            ^             ^        ^          |
| IGP|LDP/RSVP-TE   |PCEP        |PCEP     PCEP|     IGP|LDP/      |
|    |PCEP          |            |             |        |RSVP-TE/  |
|    V              V            V             V        V PCE      |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
| |NODE 1  |   | NODE 2 |   | NODE 3 |   | NODE 4 |   | NODE 5 |   |
| |        |...|        |...|        |...|        |...|        |   |
| | Legacy |IGP|        |IGP|        |IGP|  PCC4  |IGP| Legacy |   |
| |  Node  |   |        |   |        |   |        |   |  Node  |   |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
|                                                                  |
+------------------------------------------------------------------+
```

Figure 1: PCECC Architecture

Through the draft, we call the combination of the functionality for global label range signaling and the functionality of LSP setup/ download/cleanup using the combination of global labels and local labels as PCECC functionality.

Current MPLS label has local meaning. That is, MPLS label allocated locally and signaled through the LDP/RSVP-TE/BGP etc dynamic signaling protocol.

As the SDN(Service-Driven Network) technology develops, MPLS global label has been proposed again for new solutions. [I-D.li-mpls-global-label-usecases] proposes possible usecases of MPLS global label. MPLS global label can be used for identification of the location, the service and the network in different application scenarios. From these usecases we can see that no matter SDN or traditional application scenarios, the new solutions based on MPLS global label can gain advantage over the existing solutions to facilitate service provisions. The solution choices are described in [I-D.li-mpls-global-label-framework].

To ease the label allocation and signaling mechanism, also with the new applications such as concentrated LSP controller is introduced, PCE can be conveniently used as a central controller and MPLS global label range negotiator.

The later section of this draft describes the user cases for PCE server and PCE clients to have the global label range negotiation and local label range negotiation functionality.

To empower networking with centralized controllable modules, there are many choices for downloading the forwarding entries to the data plane, one way is the use of the OpenFlow protocol, which helps devices populate their forwarding tables according to a set of instructions to the data plane. There are other candidate protocols to convey specific configuration information towards devices also. Since the PCEP protocol is already deployed in some of the service network, to leverage the PCEP to populated the MPLS forwarding table is a possible good choice.

For the centralized network, the performance achieved through distributed system can not be easy matched if all of the forwarding path is computed, downloaded and maintained by the centralized controller. The performance can be improved by supporting part of the forwarding path in the PCECC network through the segment routing mechanism except that the adjacency IDs for all the network nodes and links are propagated through the centralized controller instead of using the IGP extension.

The node and link adjacency IDs can be negotiated through the PCECC with each PCECC clients and these IDs can be just taken from the global label range which has been negotiated already.

With the capability of supporting SR within the PCECC architecture, all the p2p forwarding path protection use cases described in the draft [I-D.ietf-spring-resiliency-use-cases] will be supported too within the PCECC network. These protection alternatives include end-to-end path protection, local protection without operator management and local protection with operator management.

With the capability of global label and local label existing at the same time in the PCECC network, PCECC will use compute, setup and maintain the P2MP and MP2MP lsp using the local label range for each network nodes.

With the capability of setting up/maintaining the P2MP/MP2MP LSP within the PCECC network, it is easy to provide the end-end managed path protection service and the local protection with the operation management in the PCECC network for the P2MP/MP2MP LSP, which includes both the RSVP-TE P2MP based LSP and also the mLDP based LSP.

2. Terminology

The following terminology is used in this document.

IGP: Interior Gateway Protocol. Either of the two routing protocols, Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS).

PCC: Path Computation Client: any client application requesting a path computation to be performed by a Path Computation Element.

PCE: Path Computation Element. An entity (component, application, or network node) that is capable of computing a network path or route based on a network graph and applying computational constraints.

TE: Traffic Engineering.

3. PCEP Requirements

Following key requirements associated PCECC should be considered when designing the PCECC based solution:

1. Path Computation Element (PCE) clients supporting this draft MUST have the capability to advertise its PCECC capability to the PCECC.

2. Path Computation Element (PCE) supporting this draft MUST have the capability to negotiate a global label range for a group of clients.

3. Path Computation Client (PCC) MUST be able ask for global label range assigned in path request message .

4. PCE are not required to support label reserve service. Therefore, it MUST be possible for a PCE to reject a Path Computation Request message with a reason code that indicates no support for label reserve service.

5. PCEP SHOULD provide a means to return global label range and LSP label assignments of the computed path in the reply message.

6. PCEP SHOULD provide a means to download the MPLS forwarding entry to the PCECC’s clients.

4. Use Cases of PCECC for Label Resource Reservations

Example 1 to 2 are based on network configurations illustrated using the following figure:

```
   +------------------------------+    +------------------------------+
   |         PCE DOMAIN 1         |    |         PCE DOMAIN 2         |
   |          +--------+          |    |          +--------+          |
   |          |        |          |    |          |        |          |
   |          | PCECC1 |  -----------------------  | PCECC2 |          |
   |          |        |          |    |          |        |          |
   |          |        |          |    |          |        |          |
   |          +--------+          |    |          +--------+          |
   |         ^          ^         |    |         ^          ^         |
   |        /            \        |    |        /            \        |
   |       V              V       |    |       V              V       |
   | +--------+        +--------+ |    | +--------+        +--------+ |
   | |NODE 11 |        | NODE 1n| |    | |NODE 21 |        | NODE 2n| |
   | |        | ...... |        | |    | |        | ...... |        | |
   | | PCECC  |        |  PCECC | |    | | PCECC  |        |PCECC   | |
   | |Enabled |        | Enabled|      | |Enabled |        |Enabled | |
   | +--------+        +--------+ |    | +--------+        +--------+ |
   |                              |    |                              |
   +------------------------------+    +------------------------------+
```

Figure 2: Using PCECC for Global Label Allocation

Example 1: Shared Global Label Range Reservation

- o PCECC Clients nodes report MPLS label capability to the central controller PCECC.
- o The central controller PCECC collects MPLS label capability of all nodes. Then PCECC can calculate the shared MPLS global label range for all the PCECC client nodes.
- o In the case that the shared global label range need to be negotiated across multiple domains, the central controllers of these domains need to be communicate to negotiate a common global label range.
- o The central controller PCECC notifies the shared global label range to all PCECC client nodes.

Example 2: Global Label Allocation

- o PCECC Client node1 send global label allocation request to the central controller PCECC1.

- o The central controller PCECC1 allocates the global label for FEC1 from the shared global label range and sends the reply to the client node1.

- o The central controller PCECC1 notifies the allocated label for FEC1 to all PCECC client nodes within domain 1.

5. Using PCECC for SR without the IGP Extension

For the centralized network, the performance achieved through distributed system can not be easy matched if all of the forwarding path is computed, downloaded and maintained by the centralized controller. The performance can be improved by supporting part of the forwarding path in the PCECC network through the segment routing mechanism except that node segment ids and adjacency segment IDs for all the network are allocated dynamically and propagated through the centralized controller instead of using the IGP extension.

When the PCECC is used for the distribution of the node segment ID and adjacency segment ID, the node segment ID is allocated from the global label pool. For the allocation of adjacency segment ID, there are two choices, the first choice is that it is allocated from the local label pool, the second choice is that it is allocated from the global label pool. The advantage for the second choice is that the depth of the label stack for the forwarding path encoding will be reduced since adjacency segment ID can signal the forwarding path without adding the node segment ID in front of it. In this version of the draft, we use the fist choice for now. We may update the draft to reflect the use of the second choice.

Same as the SR solutions, when PCECC is used as the central controller, the support of FRR on any topology can be pre-computated and setup without any additional signaling (other than the regular IGP/BGP protocols) including the support of shared risk constraints, support of node and link protection and support of microloop avoidance.

The following example illustrate the use case where the node segment ID and adjacency segment ID are allocated from the global label allocated for SR path.

```
                          192.0.2.1/32
                          +----------+
                          | R1(1001) |
                          +----------+
                               |
                          +----------+
                          | R2(1002) |  192.0.2.2/32
                          +----------+
                         *   |   *    *
                        *    |   *     *
                       *link1|   *      *
        192.0.2.4/32  *      |   *link2  *  192.0.2.5/32
           +-----------+ 9001|   *     +-----------+
           | R4(1004)  |     |   *     | R5(1005)  |
           +-----------+     |   *     +-----------+
                      *      |   *9003  *   +
                       *     |   *     *    +
                        *    |   *    *     +
                        +-----------+   +-----------+
           192.0.2.3/32 | R3(1003)  |   |R6(1006)   |192.0.2.6/32
                        +-----------+   +-----------+
                             |
                        +-----------+
                        | R8(1008)  |  192.0.2.8/32
                        +-----------+
```

Figure 3: Using PCECC for SR Path

5.1. Use Cases of PCECC for SR Best Effort(BE) Path

In this mode of the solution, the PCECC just need to allocate the node segment ID and adjacency ID without calculating the explicit path for the SR path. The ingress of the forwarding path just need to encapsulate the destination node segment ID on top of the packet. All the intermediate nodes will forward the packet based on the final destination node segment id. It is similar to the LDP LSP forwarding except that label swapping is using the same global label both for the in segment and out segment in each hop.

The p2p SR BE path examples are explained as bellow:

Note that the node segment id for each node from the shared global labels ranges negotiated already.

Example 1:

R1 may send a packet to R8 simply by pushing an SR header with segment list {1008}. The path can be: R1-R2-R3-R8 or R1-R2-R5-R8 depending on the route calculation on node R2.

Example 2: local link/node protection:

For the packet which has destination of R3 and after that, R2 may preinstalled the backup forwarding entry to protect the R4 node, the pre-installed the backup path can go through either node5 or link1 or link2 between R2 and R3. The backup path calculation is locally decided by R2 and any existing IP FRR algorithms can be used here.

5.2. Use Cases of PCECC for SR Traffic Engineering (TE) Path

In the case of traffic engineering path is needed, the PCECC need to allocate the node segment ID and adjacency ID, and at the same time PCECC calculates the explicit path for the SR path and pass this explicit path represented with a sequence of node segment id and adjacency id. The ingress of the forwarding path need to encapsulate the stack of node segment id and adjacency id on top of the packet. For the case where strict traffic engineering path is needed, all the intermediate nodes and links will be specified through the stack of labels so that the packet is forwarded exactly as it is wanted.

Even though it is similar to TE LSP forwarding where forwarding path is engineered, but the Qos is only guaranteed through the enforce of the bandwidth admission control. As for the RSVP-TE LSP case, Qos is guaranteed through the link bandwidth reservation in each hop of the forwarding path.

The p2p SR traffic engineering path examples are explained as bellow:

Note that the node segment id for each node is allocated from the shared global labels ranges negotiated already and adjacency segment ids for each link are allocated from the local label pool for each node.

Example 1:

R1 may send a packet P1 to R8 simply by pushing an SR header with segment list {1008}. The path should be: R1-R2-R3-R8.

Example 2:

R1 may send a packet P2 to R8 by pushing an SR header with segment list {1002, 9001, 1008}. The path should be: R1-R2-(1)link-R3-R8.

Example 3:

R1 may send a packet P3 to R8 while avoiding the links between R2 and

R3 by pushing an SR header with segment list {1004, 1008}. The path should be : R1-R2-R4-R3-R8

The p2p local protection examples for SR TE path are explained as below:

Example 4: local link protection:

- R1 may send a packet P4 to R8 by pushing an SR header with segment list {1002, 9001, 1008}. The path should be: R1-R2-(1)link-R3-R8.

- When node R2 receives the packet from R1 which has the header of R2- (1)link-R3-R8, and also find out there is a link failure of link1, then it will send out the packet with header of R3-R8 through link2.

Example 5: local node protection:

- R1 may send a packet P5 to R8 by pushing an SR header with segment list {1004, 1008}. The path should be : R1-R2-R4-R3-R8.

- When node R2 receives the packet from R1 which has the header of {1004, 1008}, and also find out there is a node failure for node4, then it will send out the packet with header of {1005, 1008} to node5 instead of node4.

6. Use Cases of PCECC for TE LSP

In the previous sections, we have discussed the cases where the SR path is setup through the PCECC. Although those cases give the simplicity and scalability, but there are existing functionalities for the traffic engineering path such as the bandwidth guarantee through the full forwarding path and the multicast forwarding path which SR based solution cannot solve. Also there are cases where the depth of the label stack may have been an issue for existing deployment and certain vendors.

So to address these issues, PCECC architecture should also support the TE LSP and multicast LSP functionalities. To achieve this, the existing PCEP can be used to communicate between the PCE server and PCE’s client PCC for exchanging the path request and reply information regarding to the TE LSP info. In this case, the TE LSP info is not only the path info itself, but it includes the full forwarding info. Instead of letting the ingress of LSP to initiate the LSP setup through the RSVP-TE signaling protocol, with minor extensions, we can use the PCEP to download the complete TE LSP forwarding entries for each node in the network.

```
                          192.0.2.1/32
                         +----------+
                         | R1(1001) |
                         +----------+
                           |       |
                       6001|link1  |
                           |   6002|link2
                          +----------+
                          | R2(1002) |  192.0.2.2/32
                          +----------+
                   link3 *   |   *    * link4
                   7002 *    |   *     *7001
                       *link1|   *      *
        192.0.2.4/32  *      |   *link2  *  192.0.2.5/32
           +-----------+ 5001|   *     +-----------+
           | R4(1004)  |     |   *     | R5(1005)  |
           +-----------+     |   *     +-----------+
                      *      |   *5003  *       +
                   9001*     |   *     *link1   +
                        *    |   *    *9002     +
                        +-----------+   +-----------+
           192.0.2.3/32 | R3(1003)  |   |R6(1006)   |192.0.2.6/32
                        +-----------+   +-----------+
                         |         |
                     3001|link1    |
                         |     3002|link2
                        +-----------+
                        | R8(1008)  |  192.0.2.8/32
                        +-----------+
```

Figure 4: Using PCECC for TE LSP

TE LSP Setup Example

- o Node1 sends a path request message for the setup of TE LSP from R1 to R8.

- o PCECC program each node along the path from R1 to R8 with the primary path: {R1, link1, 6001}, {R2, link3, 7002], {R4, link0, 9001}, {R3, link1, 3001}, {R8}.

- o For the end to end protection, PCECC program each node along the path from R1 to R8 with the secondary path: {R1, link2, 6002}, {R2, link4, 7001], {R5, link1, 9002}, {R3, link2, 3002}, {R8}.

- o It is also possible to have a secondary backup path for the local node protection setup by PCECC. For exampleGBP[not] the primary path is still same as what we have setup so far, then to protect the node R4 locally, PCECC can program the secondary path like this: {R1, link1, 6001}, {R2, link1, 5001}, {R3, link1, 3001}, {R8}. By doing this, the node R4 is locally protected.

7.   Use Cases of PCECC for Multicast LSPs

The current multicast LSPs are setup either using the RSVP-TE P2MP or mLDP protocols. The setup of these LSPs not only need a lot of manual configurations, but also it is also complex when the protection is considered. By using the PCECC solution, the multicast LSP can be computed and setup through centralized controller which has the full picture of the topology and bandwidth usage for each link. It not only reduces the complex configurations comparing the distributed RSVP-TE P2MP or mLDP signal lings, but also it can compute the disjoint primary path and secondary path efficiently.

7.1.   Using PCECC for P2MP/MP2MP LSPs' Setup

With the capability of global label and local label existing at the same time in the PCECC network, PCECC will use compute, setup and maintain the P2MP and MP2MP lsp using the local label range for each network nodes.

```
                          +----------+
                          |    R1    | Root node of the multicast LSP
                          +----------+
                              |6000
                          +----------+
           Transit Node   |    R2    |
                          +----------+
                          *  |   *  *
                     9001*   |   *   *9002
                        *    |   *    *
           +-----------+     |   *     +-----------+
           |    R4     |     |   *     |    R5     | Transit Nodes
           +-----------+     |   *     +-----------+
                      *      |   *      *     +
                   9003*     |   *     *      +9004
                        *    |   *    *       +
                        +-----------+  +-----------+
                        |    R3     |  |    R5     | Leaf Node
                        +-----------+  +-----------+
                         9005|
                        +-----------+
                        |    R8     | Leaf Node
                        +-----------+
```

Figure 5: Using PCECC for P2MP TE LSP

The P2MP examples are explained here:

Step1: R1 may send a packet P1 to R2 simply by pushing an label of 6000 to the packet.

Step2: After R2 receives the packet with label 6000, it will forwarding to R4 by pushing header of 9001 and R5 by pusing header of 9002.

Step3: After R4 receives the packet with label 9001, it will forwarding to R3 by pushing header of 9003. After R5 receives the packet with label 9002, it will forwarding to R5 by pushing header of 9004.

Step3: After R3 receives the packet with label 9003, it will forwarding to R8 by pushing header of 9005

7.2. Use Cases of PCECC for the Resiliency of P2MP/MP2MP LSPs

7.2.1. PCECC for the End-to-End Protection of the P2MP/MP2MP LSPs

In this section we describe the end-end managed path protection service and the local protection with the operation management in the PCECC network for the P2MP/MP2MP LSP, which includes both the RSVP-TE P2MP based LSP and also the mLDP based LSP.

An end-to-end protection (for nodes and links) principle can be applied for computing backup P2MP or MP2MP LSPs. During computation of the primarily multicast trees, PCECC server may also be taken into consideration to compute a secondary tree. A PCE may compute the primary and backup P2MP or MP2Mp LSP together or sequentially.

```
                               +----+  +----+
              Root node of LSP | R1 |--| R11|
                               +----+  +----+
                                 /         +
                              10/           +20
                               /             +
                       +----------+        +-----------+
        Transit Node   |    R2    |        |     R3    |
                       +----------+        +-----------+
                         |        \       +         +
                         |         \     +          +
                       10|        10\   +20       20+
                         |           \ +            +
                         |            \             +
                         |           + \            +
                       +-----------+      +-----------+ Leaf Nodes
                       |    R4     |      |    R5     | (Downstream LSR)
                       +-----------+      +-----------+
```

Figure 6: Using PCECC for P2MP TE End-to-End Protection

In the example above, when the PCECC setup the primary multicast tree from the root node R1 to the leafs, which is R1->R2->{R4, R5}, at same time, it can setup the backup tree, which is R11->R3->{R4, R5}. Both the these two primary forwarding tree and secondary forwarding tree will be downloaded to each routers along the primary path and the secondary path. The traffic will be forwarded through the R1->R2->{R4, R5} path normally, and when there is a node in the primary tree, then the root node R1 will switch the flow to the backup tree, which is R11->R3->{R4, R5}. By using the PCECC, the path computation and forwarding path downloading can all be done without the complex signaling used in the P2MP RSVP-TE or mLDP.

7.2.2. PCECC for the Local Protection of the P2MP/MP2MP LSPs

In this section we describe the local protection service in the PCECC network for the P2MP/MP2MP LSP.

While the PCECC sets up the primary multicast tree, it can also build the back LSP among PLR, the protected node, and MPs (the downstream nodes of the protected node). In the cases where the amount of downstream nodes are huge, this mechanism can avoid unnecessary packet duplication on PLR, so that protect the network from traffic congestion risk.

```
                               +------------+
                               |     R1     | Root Node
                               +------------+
                                      .
                                      .
                                      .
                               +------------+ Point of Local Repair/
                               |     R10     | Switchover Point
                               +------------+ (Upstream LSR)
                                 /         +
                              10/           +20
                               /             +
                       +----------+        +-----------+
        Protected Node |    R20   |        |     R30   |
                       +----------+        +-----------+
                         |        \       +         +
                         |         \     +          +
                       10|        10\   +20       20+
                         |           \ +            +
                         |            \             +
                         |           + \            +
                       +-----------+      +-----------+ Merge Point
                       |    R40    |      |    R50    | (Downstream LSR)
                       +-----------+      +-----------+
                             .                  .
                             .                  .
```

Figure 7: Using PCECC for P2MP TE LocalProtection

In the example above, when the PCECC setup the primary multicast path around the PLR node R10 to protect node R20, which is R10->R20->{R40, R50}, at same time, it can setup the backup path R10->R30->{R40, R50}. Both the these two primary forwarding path and secondary forwarding path will be downloaded to each routers along the primary path and the secondary path. The traffic will be forwarded through the R10->R20->{R40, R50} path normally, and when there is a node failure for node R20, then the PLR node R10 will switch the flow to the backup path, which is R10->R30->{R40, R50}. By using the PCECC, the path computation and forwarding path downloading can all be done without the complex signaling used in the P2MP RSVP-TE or mLDP.

8. Use Cases of PCECC for LSP in the Network Migration

One of the main advantages for PCECC solution is that it has backward compatibility naturally since the PCE server itself can function as a proxy node of MPLS network for all the new nodes which don’t support the existing MPLS signaling protocol anymore.

As it is illustrated in the following example, the current network will migrate to a total PCECC controlled network gradually by replacing the legacy nodes. During the migration, the legacy nodes still need to signal using the existing MPLS protocol such as LDP and RSVP-TE, and the new nodes setup their portion of the forwarding path through PCECC directly. With the PCECC function as the proxy of these new nodes, MPLS signaling can populate through network as normal.

Example described in this section is based on network configurations illustrated using the following figure:

```
+------------------------------------------------------------------+
|                         PCE DOMAIN                               |
|    +-----------------------------------------------------+       |
|    |                       PCECC                         |       |
|    +-----------------------------------------------------+       |
|     ^              ^                       ^            ^       |
|     |      RSVP-TE | if22              if44|RSVP-TE     |       |
|     V              V                       V            V       |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
| | NODE 1 |   | NODE 2 |   | NODE x |   | NODE 4 |   | NODE 5 |   |
| |        |...|        |...|        |...|        |...|        |   |
| | Legacy |if1| Legacy |if2|PCCEC   |if3| PCECC  |if4| Legacy |   |
| |  Node  |   |  Node  |   |Enabled |   |Enabled |   |  Node  |   |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
|                                                                  |
+------------------------------------------------------------------+
```

Figure 8: Using PCECC During Migration

Example: PCECC Initiated LSP Setup In the Network Migration

In this example, there are five nodes for the TE LSP from head end (node1) to the tail end (node5). Where the NodeX is central controlled and other nodes are legacy nodes.

o Node1 sends a path request message for the setup of LSP destinating to Node5.

o PCECC sends a reply message for LSP setup with path (node1, if1), (node2, if22), (node-PCECC, if44), (node4, if4), Nnode5.

o Node1, Node2, Node-PCECC, Node 5 will setup the LSP to Node5 normally using the local label as normal.

o Then the PCECC will program the outsegment of Node2, the insegment of Node4, and the insegment/outsegment for NodeX.

9. Use Cases of PCECC for L3VPN and PWE3

The existing services using MPLS LSP tunnels based on MPLS signalling mechanism such L3VPN, PWE3 and IPv6 can be simplified by using the PCECC to negoitate the label assignments for the L3VPN, PWE3 and Ipv6.

In the case of L3VPN, VPN labels can be negotiated and distributed through the PCECC PCEP among the PE router instead of using the BGP protocols.

```
                +--------------------------------------------+
                |                  PCE DOMAIN                 |
                |    +-----------------------------------+   |
                |    |                PCECC               |   |
                |    +-----------------------------------+   |
                |           ^          ^              ^       |
                |PWE3/L3VPN | PCEP PCEP|LSP PWE3/L3VPN|PCEP   |
                |           V          V              V       |
   +--------+   |     +---------+  +---------+  +---------+   |  +--------+
   |  CE    |   |     | PE1     |  | NODE x  |  | PE2     |   |  |   CE   |
   |        |......   |         |...|         |...|         |.....|        |
   | Legacy |   |if1  | PCECC   |if2|PCCEC    |if3| PCECC   |if4  | Legacy |
   |  Node  |   |     | Enabled|   |Enabled  |   |Enabled  |   |  |  Node  |
   +--------+   |     +---------+  +---------+  +---------+   |  +--------+
                |                                             |
                +--------------------------------------------+
```

Figure 9: Using PCECC for L3VPN and PWE3

In the cast PWE3, instead of using the LDP signalling protocols, the lable and port pairs assigned to each pseudowire can be negotiated through PCECC among the PE rotuers and the corresponding forwarding entries will be distributed into each PE routers through the extended PCEP protocols.

10. The Considerations for PCECC Procedure and PCEP extensions

The PCECC’s procedures and PCEP extensions is defined in [I-D.zhao-pce-pcep-extension-for-pce-controller].

11. IANA Considerations

This document does not require any action from IANA.

12. Security Considerations

TBD.

13. Acknowledgments

We would like to thank Robert Tao, Changjiang Yan, Tieying Huang for their useful comments and suggestions.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

14.2. Informative References

[RFC5441] Vasseur, JP., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, April 2009.

[RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

[I-D.filsfils-spring-segment-routing] Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-spring-segment-routing-04 (work in progress), July 2014.

[I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-11 (work in progress), April 2015.

[I-D.crabbe-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE

Model", draft-crabbe-pce-pce-initiated-lsp-03 (work in progress), October 2013.

[I-D.ali-pce-remote-initiated-gmpls-lsp] Ali, Z., Sivabalan, S., Filsfils, C., Varga, R., Lopez, V., Dios, O., and X. Zhang, "Path Computation Element Communication Protocol (PCEP) Extensions for remote-initiated GMPLS LSP Setup", draft-ali-pce-remote-initiated-gmpls-lsp-03 (work in progress), February 2014.

[I-D.ietf-isis-segment-routing-extensions] Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.

[I-D.psenak-ospf-segment-routing-extensions] Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", draft-psenak-ospf-segment-routing-extensions-05 (work in progress), June 2014.

[I-D.sivabalan-pce-segment-routing] Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Raszuk, R., Lopez, V., and J. Tantsura, "PCEP Extensions for Segment Routing", draft-sivabalan-pce-segment-routing-03 (work in progress), July 2014.

[I-D.li-mpls-global-label-usecases] Li, Z., Zhao, Q., Yang, T., and R. Raszuk, "Use Cases of MPLS Global Label", draft-li-mpls-global-label-usecases-02 (work in progress), July 2014.

[I-D.li-mpls-global-label-framework] Li, Z., Zhao, Q., Chen, X., Yang, T., and R. Raszuk, "A Framework of MPLS Global Label", draft-li-mpls-global-label-framework-02 (work in progress), July 2014.

[I-D.zhao-pce-pcep-extension-for-pce-controller] Zhao, Q., Zhao, K., Li, Z., Dhody, D., Palle, U., and T. Communications, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", d raft-zhao-pce-pcep-extension-for-pce-controller-01 (work in progress), March 2015.

[I-D.ietf-spring-resiliency-use-cases] Francois, P., Filsfils, C., Decraene, B., and R. Shakir, "Use-cases for Resiliency in SPRING", draft-ietf-spring-resiliency-use-cases-01 (work in progress), March 2015.

Authors' Addresses

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
US

EMail: quintin.zhao@huawei.com

Robin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

EMail: lizhenbin@huawei.com

King Ke
Tencent Holdings Ltd.
Shenzhen
China

EMail: kinghe@tencent.com

Luyuan Fang
Microsoft

EMail: lufang@microsoft.com

Chao Zhou
Cisco Systems

EMail: chao.zhou@cisco.com

Boris Zhang
Telus Communications
200 Consilium Pl Floor 15
Toronto, ON M1H 3J3
Canada

EMail: Boris.Zhang@telus.com

Katherine Zhao
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: Katherine.zhao@huawei.com

Richard Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

EMail: renwei.li@huawei.com

TEAS Working Group Y. Zhuang, Ed.
Internet-Draft Q. Wu
Intended status: Standards Track H. Chen
Expires: February 9, 2017 Huawei
A. Farrel
Juniper Networks
August 8, 2016

Architecture for Scheduled Use of Resources
draft-zhuang-teas-scheduled-resources-03

Abstract

Time-Scheduled reservation of traffic engineering (TE) resources can be used to provide resource booking for TE Label Switched Paths so as to better guarantee services for customers and to improve the efficiency of network resource usage into the future. This document provides a framework that describes and discusses the architecture for the scheduled reservation of TE resources. This document does not describe specific protocols or protocol extensions needed to realize this service.

Table of Contents

1. Introduction

Traffic Engineering Label Switched Paths (TE-LSPs) are connection oriented tunnels in packet and non-packet networks [RFC3209], [RFC3945]. TE-LSPs may reserve network resources for use by the traffic they carry, thus providing some guarantees of service delivery and allowing a network operator to plan the use of the resources across the whole network.

In some technologies (such as wavelength switched optical networks) the resource is synonymous with the label that is switched on the path of the LSP so that it is not possible to establish an LSP that can carry traffic without assigning a concrete resource to the LSP. In other technologies (such as packet switched networks) the resources assigned to an LSP are a measure of the capacity of a link that is dedicated for use by the traffic on the LSP. In all cases, network planning consists of selecting paths for LSPs through the network so that there will be no contention for resources; LSP establishment is the act of setting up an LSP and reserving resources within the network; and network optimization or re-optimization is the process of re-positioning LSPs in the network to make the unreserved network resources more useful for potential future LSPs while ensuring that the established LSPs continue to fulfill their objectives.

It is often the case that it is known that an LSP will be needed at some time in the future. While a path for that LSP could be computed using knowledge of the currently established LSPs and the currently available resources, this does not give any degree of certainty that the necessary resources will be available when it is time to set up the new LSP. Yet setting up the LSP ahead of the time when it is needed (which would guarantee the availability of the resources) is wasteful since the network resources could be used for some other purpose in the meantime.

Similarly, it may be known that an LSP will no longer be needed after some future time and that it will be torn down releasing the network resources that were assigned to it. This information can be helpful in planning how a future LSP is placed in the network.

Time-Scheduled (TS) reservation of TE resources can be used to provide resource booking for TE-LSPs so as to better guarantee services for customers and to improve the efficiency of network resource usage into the future. This document provides a framework that describes and discusses the architecture for the scheduled reservation of TE resources. This document does not describe specific protocols or protocol extensions needed to realize this service.

2. Problem statement

2.1. Provisioning TE-LSPs and TE Resources

TE-LSPs in existing networks are provisioned using RSVP-TE as a signaling protocol [RFC3209] [RFC3473], by direct control of network elements such as in the Software Defined Networking (SDN) paradigm, and using the PCE Communication Protocol (PCEP) [RFC5440] as a control protocol.

TE resources are reserved at the point of use. That is, the resources (wavelengths, timeslots, bandwidth, etc.) are reserved for use on a specific link and are tracked by the Label Switching Routers (LSRs) at the end points of the link. Those LSRs learn which resources to reserve during the LSP setup process.

The use of TE resources can be varied by changing the parameters of the LSP that uses them, and the resources can be released by tearing down the LSP.

2.2. Selecting the Path of an LSP

Although TE-LSPs can determine their paths hop-by-hop using the shortest path toward the destination to route the signaling protocol messages [RFC3209], in practice this option is not applied because it does not look far enough ahead into the network to verify that the desired resources are available. Instead, the full length of the path of an LSP is computed ahead of time either by the head-end LSR of a signaled LSP, or by Path Computation Element (PCE) functionality in a dedicated server or built into network management software [RFC4655].

Such full-path computation is applied in order that an end-to-end view of the available resources in the network can be used to determine the best likelihood of establishing a viable LSP that meets the service requirements. Even in this situation, however, it is possible that two LSPs being set up at the same time will compete for scarce network resources meaning that one or both of them will fail to be established. This situation is avoided by using a centralized PCE that is aware of the LSP setup requests that are in progress.

2.3. Planning Future LSPs

LSPs may be established "on demand" when the requester determines that a new LSP is needed. In this case, the path of the LSP is computed as described in Section 2.2.

However, in many situations, the requester knows in advance that an LSP will be needed at a particular time in the future. For example, the requester may be aware of a large traffic flow that will start at a well-known time, perhaps for a database synchronzation or for the exchange of content between streamng sites. Furthermore, the requester may also know for how long the LSP is required before it can be torn down.

The set of requests for future LSPs could be collected and held in a central database (such as at a Network Management System - NMS): when the time comes for each LSP to be set up the NMS can ask the PCE to compute a path and can then requst the LSP to be provisioned. This approach has a number of drawbacks because it is not possible to determine in advance whether it will be possible to deliver the LSP since the resources it needs might be used by other LSPs in the network. Thus, at the time the requester asks for the future LSP, the NMS can only make a best-effort guarantee that the LSP will be set up at the desired time.

A better solution, therefore, is for the requests for future LSPs to be serviced at once. The paths of the LSPs can be computed ahead of time and converted into reservations of network resources during specific windows in the future.

2.4. Looking at Future Demands on TE Resources

While path computation as described in Section 2.2 takes account of the currently available network resources, and can act to place LSPs in the network so that there is the best possibility of future LSPs being accommodated, it cannot handle all eventualities. It is simple to construct scenarios where LSPs that are placed one at a time lead to future LSPs being blocked, but where foreknowledge of all of the LSPs would have made it possible for them all to be set up.

If, therefore, we were able to know in advance what LSPs were going to be requested we could plan for them and ensure resources were available. Furthermore, such an approach enables a commitment to be made to a service user that an LSP will be set up and available at a specific time.

This service can be achieved by tracking the current use of network resources and also a future view of the resource usage. We call this time-scheduled TE (TS-TE) resource reservation.

2.5. Requisite State Information

In order to achieve the TS-TE resource reservation, the use of resources on the path needs to be scheduled. Scheduling state is used to indicate when resources are reserved and when they are available for use.

A simple information model for one piece of scheduling state is as follows:

```
{ link id;
  resource id or reserved capacity;
  reservation start time;
  reservation end time
}
```

The resource that is scheduled can be link capacity, physical resources on a link, CPU utilization, memory, buffers on an interfaces, etc. The resource might also be the maximal unreserved bandwidth of the link over a time intervals. For any one resource there could be multiple pieces of scheduling state, and for any one link, the timing windows might overlap.

There are multiple ways to realize this information model and different ways to store the data. The resource state could be expressed as a start time and and end time as shown above, or could be expressed as a start time and a duration. Multiple periods, possibly of different lengths, may be associated with one reservation request, and a reservation might repeat on a regular cycle. Furthermore, the current state of network reservation could be kept separate from the scheduled usage, or everything could be merged into a single TS databasae. This document does not spend any more time on discussion of encoding of state information except to discuss the location of storage of the state information and the recovery of the information after failure events.

This scheduling state information can be used by applications to book resources for future or now, so as to maximize chance of services being delivered. Also, it can avoid contention for resources of LSPs.

Note that it is also to store the information about future LSPs. This information is held to allow the LSPs to be instantiated when they are due and using the paths/resources that have been computed for them, but also to provide correlation with the TS-TE resource reservations so that it is clear why resources were reserved allowing pre-emption and handling release of reserved resources in the event of cancelation of future LSPs.

3. Architectural Concepts

This section examines several important architectural concepts that lead to design decisions that will influence how networks can achieve TS-TE in a scalable and robust manner.

3.1. Where is Scheduling State Held?

The scheduling state information described in Section 2.5 has to be held somewhere. There are two places where this makes sense:

- o In the network nodes where the resources exist;
- o In a central scheduling controller where decisions about resource allocation are made.

The first of these makes policing of resource allocation easier. It means that many points in the network can request immediate or scheduled LSPs with the associated resource reservation and that all such requests can be correlated at the point where the resources are allocated. However, this approach has some scaling and technical problems:

- o The most obvious issue is that each network node must retain the full time-based state for all of its resources. In a busy network with a high arrival rate of new LSPs and a low hold time for each LSP, this could be a lot of state. Yet network nodes are normally implemented with minimal spare memory.

- o In order that path computation can be performed, the computing entity normally known as a Path Computation Element (PCE) [RFC4655] needs access to a database of available links and nodes in the network, and of the TE properties of the links. This database is known as the Traffic Engineering Database (TED) and is usually populated from information advertised in the IGP by each of the network nodes or exported using BGP-LS [I-D.ietf-idr-ls-distribution]. To be able to compute a path for a future LSP the PCE needs to populate the TED with all of the future resource availability: if this information is held on the network nodes it must also be advertised in the IGP. This could be a significant scaling issue for the IGP and the network nodes as all of the advertised information is held at every network node and must be periodically refreshed by the IGP.

- o When a normal node restarts it can recover resource reservation state from the forwarding hardware, from Non-volatile random-access memory (NVRAM), or from adjacent nodes through the signaling protocol [RFC5063]. If scheduling state is held at the network nodes it must also be recovered after the restart of a network node. This cannot be achieved from the forwarding hardware because the reservation will not have been made, could require additional expensive NVRAM, or might require that all adjacent nodes also have the scheduling state in order to reinstall it on the restarting node. This is potentially complex processing with scaling and cost implications.

Conversely, if the scheduling state is held centrally it is easily available at the point of use. That is, the PCE can utilize the state to plan future LSPs and can update that stored information with the scheduled reservation of resources for those future LSPs. This approach also has several issues:

- o If there are multiple controllers then they must synchronise their stored scheduling state as they each plan future LSPs, and must have a mechanism to resolve resource contention. This is relatively simple and is mitigated by the fact that there is ample processing time to replan future LSPs in the case of resource contention.

- o If other sources of immediate LSPs are allowed (for example, other controllers or autonomous action by head-end LSRs) then the changes in resource availability caused by the setup or teardown of these LSPs must be reflected in the TED (by use of the IGP as currently) and may have an impact of planned future LSPs. This impact can be mitigated by replanning future LSPs or through LSP preemption.

- o If other sources of planned LSPs are allowed, they can request path computation and resource reservation from the centralized PCE using PCEP [RFC5440].

- o If the scheduling state is held centrally at a PCE, the state must be held and restored after a system restart. This is relatively easy to achieve on a central server that can have access to non-volatile storage. The PCE could also synchronize the scheduling state with other PCEs after restart. See Section 4.2 for details.

- o Of course, a centralized system must store informaton about all of the resources in the network. In a busy network with a high arrival rate of new LSPs and a low hold time for each LSP, this could be a lot of state. This is multiplied by the size of the network measured both by the number of links and nodes, and by the number of trackable resources on each link or at each node. The challenge may be mitigated by the centralized server being dedicated hardware, but the problem of collecting the information from the network is only solved if the central server has full control of the booking of resources and the estblshment of new LSPs.

Thus the architectural conclusion is that scheduling state should be held centrally at the point of use and not in the network devices.

3.2. What State is Held?

As already described, the PCE needs access to an enhanced, time-based TED. It stores the traffic engineering (TE) information such as bandwidth for every link for a series of time intervals. There are a few ways to store the TE information in the TED. For example, suppose that the amount of the unreserved bandwidth at a priority level for a link is Bj in a time interval from time Tj to Tk (k = j+1), where j = 0, 1, 2, ...

```
        Bandwidth
         ^
         |                                    B3
         |          B1                        ___________
         |          __________
         |B0                                             B4
         |__________          B2                         _________
         |                    ________________
         |
        -+-------------------------------------------------------> Time
         |T0        T1        T2              T3         T4
```

Figure 1: A Plot of Bandwidth Usage against Time

The unreserved bandwidth for the link can be represented and stored in the TED as [T0, B0], [T1, B1], [T2, B2], [T3, B3], ... as shown in Figure 1.

But it must be noted that service requests for future LSPs are known in terms of the LSPs whose paths are computed and for which resources are scheduled. For example, if the requester of a future LSP decides to cancel the request or to modify the request, the PCE must be able to map this to the resources that were reserved. When the LSP or the request for the LSP with a number of time intervals is cancelled, the PCE must release the resources that were reserved on each of the links along the path of the LSP in every time intervals from the TED. If the bandwidth reserved on a link for the LSP is B from time T2 to T3 and the unreserved bandwidth on the link is B2 from T2 to T3, B is added to the link for the time interval from T2 to T3 and the unreserved bandwidth on the link from T2 to T3 will be B2 + B.

This suggests that the PCE needs an LSP Database (LSP-DB) [I-D.ietf-pce-stateful-pce] that contains information not only about LSPs that are active in the network, but also those that are planned. The information for an LSP stored in the LSP-DB includes for each time interval that applies to the LSP: the time interval, the paths computed for the LSP satisfying the constraints in the time interval, and the resources such as bandwidth reserved for the LSP in the time interval. See also Section 2.3

It is an implementation choice how the TED and LSP-DB are stored both for dynamic use and for recovery after failure or restart, but it may be noted that all of the information in the scheduled TED can be recovered from the active network state and from the scheduled LSP-DB.

4. Architecture Overview

The architectural considerations and conclusions described in the previous section lead to the architecture described in this section.

```
          -------------------
         | Service Requester |
          -------------------
                   ^
                  a|
                   v
                -------   b   --------
               |       |<--->| LSP-DB |
               |       |      --------
               |  PCE  |
               |       |  c    -----
               |       |<---->| TED |
                -------        -----
                ^     ^
                |     |
               d|     |e
                |     |
          ------+-----+--------------------
                |     |          Network
                |    --------
                |   | Router |
                v    --------
              -----          -----
             | LSR |<------>| LSR |
              -----     f    -----
```

Figure 2: Reference Architecture for Scheduled Use of Resources

4.1. Service Request

As shown in Figure 2, some component in the network requests a service. This may be an application, an NMS, an LSR, or any component that qualifies as a Path Computation Client (PCC). We show this on the figure as the "Service Requester" and it sends a request to the PCE for an LSP to be set up at some time (either now or in the future). The request, indicated on Figure 2 by the arrow (a) includes all of the parameters of the LSP that the requester wishes to supply such as bandwidth, start time, and end time. Note that the requester in this case may be the same LSR shown in the figure or may be a distinct system.

The PCE enters the LSP request in its LSP-DB (b), and uses information from its TED (c) to compute a path that satisfies constraints such as bandwidth constraint for the LSP in the time interval from a start time to an end time. It updates the future resource availability in the TED so that further path computations can take account of the scheduled resource usage. It stores the path for the LSP into the LSP-DB (b).

When it is time such as at a start time for the LSP to be set up, the PCE sends a PCEP Initiate request to the head end LSR (d) providing the path to be signaled as well as other parameters such as the bandwidth of the LSP.

As the LSP is signaled between LSRs (f) the use of resources in the network is updated and distributed using the IGP. This information is shared with the PCE either through the IGP or using BGP-LS (e), and the PCE updates the information stored in its TED (c).

After the LSP is set up, the head end LSR sends a PCEP LSP State Report (PCRpt message) to the PCE (d). The report contains the resources such as bandwidth usage for the LSP. The PCE updates the status of the LSP in the LSPDB according to the report.

When an LSP is no longer required (either because the Service Requester has cancelled the request, or because the LSP’s scheduled lifetime has expired) the PCE can remove it. If the LSP is currently active, the PCE instructs the head-end LSR to tear it down (d), and the network resource usage will be updated by the IGP and advertised back to the PCE through the IGP or BGP-LS (e). Once the LSP is no longer active, the PCE can remove it from the LSP-DB (b).

4.2. Initialization and Recovery

When a PCE in the architecture shown in Figure 2 is initialized, it must learn state from the network, from its stored databases, and potentially from other PCEs in the network.

The first step is to get an accurate view of the topology and resource availability in the network. This would normally involve reading the state direct from the network via the IGP or BGP-LS (e), but might include receiving a copy of the TED from another PCE. Note that a TED stored from a previous instantiation of the PCE is unlikely to be valid.

Next, the PCE must construct a time-based TED to show scheduled resource usage. How it does this is implementation specific and this document does not dictate any particular mechanism: it may recover a time-based TED previously saved to non-volatile storage, or it may reconstruct the time-based TED from information retrieved from the LSP-DB previously saved to non-volatile storage. If there is more than one PCE active in the network, the recovering PCE will need to synchronize the LSP-DB and time-based TED with other PCEs (see Section 4.3).

4.3. Synchronization Between PCEs

If there is more than one PCE active in the network which supports scheduling, it is important to achieve some consistency between the scheduled TED and scheduled LSP-DB between the PCEs.

[RFC7399] answers various questions around synchronization between the PCEs. It should be noted that the time-based "scheduled" information adds another dimension to it. It should be noted that the deployment may use a primary PCE and the other PCEs as backup, where the backup PCE can take over only in the event of a failure of the primary PCE. Or the PCEs may share the load at all times. The choice of the synchronization technique is largely dependent on the deployment of PCEs in the network.

One option for ensuring that multiple PCEs use the same scheduled information is simply to have the PCEs driven from the same shared database, but it is likely to be inefficient and inter-operation between multiple implementation harder.

Or the PCEs might be responsible for its own scheduled database and utilize some distributed database synchronization mechanism to have a consistent database. Based on the implementation, this could be efficient but the inter-operation between heterogeneous implementation is still hard.

Another approach would be to utilize PCEP messages to synchronize the scheduled state between PCEs. This approach would work well if the number of PCEs which support scheduling are less, but as the number increases considerable message exchange needs to happen to keep the scheduled database in sync. Future solution could also utilize some synchronization optimization techniques for efficiency. Another variation would be to request information from other PCEs for a particular time slice but this might have impact on the optimization algorithm.

5. Security Consideration

TBD

6. Acknowledgements

This work has benefited from the discussions of resource scheduling over the years. In particular the DRAGON project [DRAGON] and [I-D.yong-ccamp-ason-gmpls-autobw-service] both of which provide approaches to auto-bandwidth services in GMPLS networks.

Mehmet Toy, Lei Liu and Khuzema Pithewan contributed the earlier version of [I-D.chen-teas-frmwk-tts]. We would like to thank authors of that draft on Temporal Tunnel Services andfor help inspire discussion in the TEAS WG and get this work solid.

Thanks to Michael Scharf and Daniele Ceccarelli for useful comments on this work.

7. Contributors

The following people contributed to discussions that led to the development of this document:

Dhruv Dhody
Email: dhruv.dhody@huawei.com

8. Informative References

[DRAGON] National Science Foundation, "http://www.maxgigapop.net/wp-content/uploads/The-DRAGON-Project.pdf".

[I-D.chen-teas-frmwk-tts]
Chen, H., Toy, M., Liu, L., and K. Pithewan, "Framework for Temporal Tunnel Services", draft-chen-teas-frmwk-tts-01 (work in progress), March 2016.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-13 (work in progress), October 2015.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-15 (work in progress), July 2016.

[I-D.yong-ccamp-ason-gmpls-autobw-service]
Yong, L. and Y. Lee, "ASON/GMPLS Extension for Reservation and Time Based Automatic Bandwidth Service", draft-yong-ccamp-ason-gmpls-autobw-service-00 (work in progress), October 2006.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <http://www.rfc-editor.org/info/rfc3209>.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <http://www.rfc-editor.org/info/rfc3473>.

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <http://www.rfc-editor.org/info/rfc3945>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <http://www.rfc-editor.org/info/rfc4655>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <http://www.rfc-editor.org/info/rfc5063>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <http://www.rfc-editor.org/info/rfc5440>.

[RFC7399] Farrel, A. and D. King, "Unanswered Questions in the Path Computation Element Architecture", RFC 7399, DOI 10.17487/RFC7399, October 2014, <http://www.rfc-editor.org/info/rfc7399>.

Authors’ Addresses

Yan Zhuang (editor)
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhuangyan.zhuang@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Huaimo Chen
Huawei
Boston, MA
US

Email: huaimo.chen@huawei.com

Adrian Farrel
Juniper Networks

Email: adrian@olddog.co.uk