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

TEAS Working Group
Internet Draft
Intended status: Informational

Expires September 1, 2018.

Young Lee (Editor)
Huawei

Daniele Ceccarelli
Ericsson

Takuya Miyasaka
KDDI

Jong Yoon Shin
SKT

Kwang-koog Lee
KT

March 1, 2018

Requirements for Abstraction and Control of TE Networks

draft-ietf-teas-actn-requirements-09

Abstract

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

Status of this Memo

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   H. Chen, Ed.
Internet-Draft   Huawei Technologies
Intended status: Experimental   R. Torvi, Ed.
Expires: September 19, 2018   Juniper Networks
                                                          March 18, 2018

Extensions to RSVP-TE for LSP Ingress FRR Protection
draft-ietf-teas-rsvp-ingress-protection-17.txt

Abstract

This document describes extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for locally protecting the ingress node of a Point-to-Point (P2P) or Point-to-Multipoint (P2MP) Traffic Engineered (TE) Label Switched Path (LSP). It extends the fast-reroute (FRR) protection for transit nodes of an LSP to the ingress node of the LSP. The procedures described in this document are experimental.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

For a MPLS TE LSP, protecting the failures of its transit nodes using fast-reroute (FRR) is covered in RFC 4090 for P2P LSP and RFC 4875 for P2MP LSP. However, protecting the failure of its ingress node using FRR is not covered in either RFC 4090 or RFC 4875. The MPLS Transport Profile (MPLS-TP) Linear Protection described in RFC 6378 can provide a protection against the failure of any transit node of a LSP between the ingress node and the egress node of the LSP, but cannot protect against the failure of the ingress node.

To protect against the failure of the (primary) ingress node of a primary end to end P2MP (or P2P) TE LSP, a typical existing solution is to set up a secondary backup end to end P2MP (or P2P) TE LSP. The backup LSP is from a backup ingress node to backup egress nodes (or node). The backup ingress node is different from the primary ingress node. The backup egress nodes (or node) are (or is) different from the primary egress nodes (or node) of the primary LSP. For a P2MP TE LSP, on each of the primary (and backup) egress nodes, a P2P LSP is created from the egress node to its primary (backup) ingress node and configured with BFD. This is used to detect the failure of the primary (backup) ingress node for the receiver to switch to the backup (or primary) egress node to receive the traffic after the primary (or backup) ingress node fails when both the primary LSP and the secondary LSP carry the traffic. In addition, FRR may be used to provide protections against the failures of the transit nodes and the links of the primary and secondary end to end TE LSPs.

There are a number of issues in this solution:

- o It consumes lots of network resources. Double states need to be maintained in the network since two end to end TE LSPs are created. Double link bandwidth is reserved and used when both the primary and the secondary end to end TE LSPs carry the traffic at the same time.

- o More operations are needed, which include the configuration of two end to end TE LSPs and BFDs from each of the egress nodes to its corresponding ingress node.

- o The detection of the failure of the ingress node may not be reliable. Any failure on the path of the BFD from an egress node to an ingress node may cause the BFD to indicate the failure of the ingress node.

o The speed of protection against the failure of the ingress node may be slow.

This specification defines a simple extension to RSVP-TE for local protection (FRR) of the ingress node of a P2MP or P2P LSP to resolve these issues. Ingress local protection and ingress FRR protection will be used exchangeably.

Note that this document is experimental. Two different approaches are proposed to transfer the information for ingress protection. They both use the same new INGRESS_PROTECTION object, which is sent in both PATH and RESV messages between a primary ingress and a backup ingress. One approach is Relay-Message Method (refer to section 5.1.1 and 5.2.1), the other is Proxy-Ingress Method (refer to section 5.1.2 and 5.2.2). Each of them has its advantages and disadvantages. It is hard to decide which one is used as a standard approach now. It is expected that the experiment on the ingress local protection with these two approaches provides quantities to help choose one. The quantities include the numbers on control traffic, states, codes and operations. After one approach is selected, the document will be revised to reflect that selection and any other items learned from the experiment. The revised document is expected to be submitted for publication on the standards track.

1.1. Ingress Local Protection Example

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 Ia to three egresses: L1, L2 and L3. The backup LSP is from backup ingress Ib to the next hops R2 and R4 of ingress Ia.

```
                       *******  *******              S Source
                    [R2]-----[R3]-----[L1]          Ix Ingress
                   */ &                             Rx Transit
                  */  &                             Lx Egress
                 */   &                            *** Primary LSP
                */    &                            &&& Backup LSP across
               */     &                                logical hop
              */      &
             */ ********    ********  *******
      [S]---[Ia]--------[R4]------[R5]-----[L2]
        \      |     &    &           *\
         \     |    &    &             *\
          \    |   &    &               *\
           \   |  &    &                 *\
            \  | &    &                   *\
             \ |&    &                     *\
              [Ib]&&&                       [L3]
```

Figure 1: Ingress Local Protection

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

When source S detects the failure of Ia, it switches the traffic to backup ingress Ib, which imports the traffic from S into the backup LSP to Ia’s next hops R2 and R4, where the traffic is merged into the primary LSP, and then sent to egresses L1, L2 and L3.

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.

1.2. Ingress Local Protection Overview

There are four parts in ingress local protection:

- o Setting up the necessary backup LSP forwarding state based on the information received for ingress local protection;
- o Detecting the primary ingress failure and providing the fast repair (as discussed in Sections 2 and 3);
- o Maintaining the RSVP-TE control plane state until a global repair is done; and
- o Performing the global repair(see Section 5.4.2).

The primary ingress of a primary LSP sends the backup ingress the information for ingress protection in a PATH message with a new INGRESS_PROTECTION object. The backup ingress sets up the backup LSP(s) and forwarding state after receiving the necessary information for ingress protection. And then it sends the primary ingress the status of ingress protection in a RESV message with a new INGRESS_PROTECTION object.

When the primary ingress fails, the backup ingress sends or refreshes the next hops of the primary ingress the PATH messages without any INGRESS_PROTECTION object after verifying the failure. Thus the RSVP-TE control plane state of the primary LSP is maintained.

2. 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.

2.1. 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. Fast detecting the failure means detecting the failure in a few or tens of milliseconds. The backup ingress is ready to import the traffic from the source into the backup LSP(s) after the backup LSP(s) 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(s), where the traffic is merged into the primary LSP.

For an LSP, after the primary ingress fails, the backup ingress MUST use a method to verify the failure of the primary ingress before the PATH message for the LSP expires at the next hop of the primary ingress. After verifying the failure, the backup ingress sends/refreshes the PATH message to the next hop through the backup LSP as needed. The method may verify the failure of the primary ingress slowly such as in seconds.

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.

When the previously failed primary ingress of a primary LSP becomes available again and the primary LSP has recovered from its primary ingress, the source may switch the traffic to the primary ingress from the backup ingress. A operator may control the traffic switch through using a command on the source node after seeing that the primary LSP has recovered.

2.2. 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.

Note that one of the differences between Source-Detect and Backup-Source-Detect is: in the former, the backup ingress verifies the failure of the primary ingress slowly such as in seconds; in the latter, the backup ingress detects the failure fast such as in a few or tens of milliseconds.

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.

3. 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, it is off-path. If a backup ingress is a next-hop of the primary ingress of the LSP, it is on-path. When a backup ingress for protecting the primary ingress is configured, the backup ingress MUST not be on the LSP except for it is the next hop of the primary ingress. If it is on-path, the primary forwarding state associated with the primary LSP SHOULD be clearly separated from the backup LSP(s) state.

3.1. 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.

4. Protocol Extensions

A new object INGRESS_PROTECTION is defined for signaling ingress local protection. The primary ingress of a primary LSP sends the backup ingress this object in a PATH message. In this case, the object contains the information needed to set up ingress protection. The information includes:

- Backup ingress IP address indicating the backup ingress,
- Traffic Descriptor describing the traffic that the primary LSP transports, this traffic is imported into the backup LSP(s) on the backup ingress when the primary ingress fails,
- Label and Routes indicating the first hops of the primary LSP, each of which is paired with its label, and
- Desire options on ingress protection such as P2MP option indicating a desire to use a backup P2MP LSP to protect the primary ingress of a primary P2MP LSP.

The backup ingress sends the primary ingress this object in a RESV message. In this case, the object contains the information about the status on the ingress protection.

4.1. INGRESS_PROTECTION Object

4.1.1. Class Number and Class Type

The Class Number for the INGRESS_PROTECTION object MUST be of the form 0bbbbbbb to enable implementations that do not recognize the object to reject the entire message and return an "Unknown Object Class" error [RFC2205]. It is suggested that a Class Number value from the Private Use range (124-127) [RFC3936] specified for the 0bbbbbbb octet be chosen for this experiment. It is also suggested that a Class Type value of 1 be used for this object in this experiment.

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.

4.1.2. Object Format

The INGRESS_PROTECTION object has the following format:

```
  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)    |   NUB   |     Flags     |   Options     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                          (Subobjects)                         ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    NUB      Number of Unprotected Branches
    Flags
     0x01    Ingress local protection available
     0x02    Ingress local protection in use
     0x04    Bandwidth protection

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

For protecting the ingress of a P2MP LSP, if the backup ingress doesn’t have a backup LSP to each of the next hops of the primary ingress, it SHOULD clear "Ingress local protection available" and set NUB to the number of the next hops to which there is no backup LSP.

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

- o Ingress local protection available: The backup ingress MUST set 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 MUST set this flag when it detects a failure in the primary ingress and actively redirects the traffic into the backup LSPs. The backup ingress records this flag and does not send any RESV message with this flag to the primary ingress since the primary ingress is down.

- o Bandwidth protection: The backup ingress MUST set 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 backup P2MP LSP to protect the primary ingress.

The INGRESS_PROTECTION object may contain some subobjects of following format:

```
 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           |Reserved (zero)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Contents/Body of subobject                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where Type is the type of a subobject, Length is the total size of the subobject in bytes, including Type, Length and Contents fields.

4.1.3. 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 MUST have a Backup Ingress IPv4 Address subobject containing an IPv4 address belonging to the backup ingress if IPv4 is used. The Type of the subobject is 1, and the body of the subobject 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

4.1.4. 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 MUST have a Backup Ingress IPv6 Address subobject containing an IPv6 address belonging to the backup ingress if IPv6 is used. The Type of the subobject is 2, the body of the subobject 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

4.1.5. Subobject: Ingress IPv4 Address

The INGRESS_PROTECTION object may have an Ingress IPv4 Address subobject containing an IPv4 address belonging to the primary ingress if IPv4 is used. The Type of the subobject is 3. The subobject 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

4.1.6. Subobject: Ingress IPv6 Address

The INGRESS_PROTECTION object may have an Ingress IPv6 Address subobject containing an IPv6 address belonging to the primary ingress if IPv6 is used. The Type of the subobject is 4. The subobject 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

4.1.7. Subobject: Traffic Descriptor

The INGRESS_PROTECTION object may have a Traffic Descriptor subobject describing the traffic to be mapped to the backup LSP on the backup ingress for locally protecting the primary ingress. The subobject types for Interface, IPv4 Prefix, IPv6 Prefix and Application Identifier are 5, 6, 7 and 8 respectively. The subobject 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 subobject may contain multiple Traffic Elements of same type as follows:

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

4.1.8. Subobject: Label-Routes

The INGRESS_PROTECTION object in a PATH message from the primary ingress to the backup ingress may have a Label-Routes subobject containing the labels and routes that the next hops of the ingress use. The Type of the subobject is 9. The subobject 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.

5. Behavior of Ingress Protection

5.1. Overview

There are two different proposed signaling approaches to transfer the information for ingress protection. They both use the same new INGRESS_PROTECTION object. The object is sent in both PATH and RESV messages.

5.1.1. Relay-Message Method

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. An INGRESS_PROTECTION object without any Traffic-Descriptor subobject is called 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.

The empty INGRESS_PROTECTION object is for efficient processing of ingress protection for a P2MP LSP. For a P2MP LSP, its primary ingress may have more than one PATH messages, each of which is sent to a next hop along a branch of the P2MP LSP. The PATH message along a branch will be selected and sent to the backup ingress with an INGRESS_PROTECTION object containing the Traffic-Descriptor subobject; all the PATH messages along the other branches will be sent to the backup ingress containing an INGRESS_PROTECTION object without any Traffic-Descriptor subobject (empty INGRESS_PROTECTION object). For a P2MP LSP, the backup ingress only needs one Traffic-Descriptor.

5.1.2. Proxy-Ingress Method

Conceptually, a proxy ingress is created that starts the RSVP signaling. The explicit path of the LSP goes from the proxy ingress to the backup ingress and then to the real ingress. The behavior and signaling for the proxy ingress is done by the real ingress; the use of a proxy ingress address avoids problems with loop detection. Note that the proxy ingress MUST reside within the same router as the real ingress.

```
                     [ traffic source ]       *** Primary LSP
                      $             $         --- Backup LSP
                      $             $          $$  Link
                      $             $
              [ proxy ingress ]  [ backup ]
              [ & ingress     ]     |
                     *              |
                     *****[ MP ]----|
```

Figure 2: Example Protected LSP with Proxy Ingress Node

The backup ingress MUST know the merge points or next-hops and their associated labels. This is accomplished by having the RSVP PATH and RESV messages go through the backup ingress, although the forwarding path need not go through the backup ingress. If the backup ingress fails, the ingress simply removes the INGRESS_PROTECTION object and forwards the PATH messages to the LSP's next-hop(s). If the ingress has its LSP configured for ingress protection, then the ingress can add the backup ingress and itself to the ERO and start forwarding the PATH messages to the backup ingress.

Slightly different behavior can apply for the on-path and off-path cases. In the on-path case, the backup ingress is a next hop node after the ingress for the LSP. In the off-path, the backup ingress is not any next-hop node after the ingress for all associated sub-LSPs.

The key advantage of this approach is that it minimizes the special handling code required. Because the backup ingress is on the signaling path, it can receive various notifications. It easily has access to all the PATH messages needed for modification to be sent to refresh control-plane state after a failure.

5.2. 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 the IP address of the backup ingress it wants to be used before it can use the INGRESS_PROTECTION object.

- o Proxy-Ingress-Id (only needed for Proxy-Ingress Method): The Proxy-Ingress-Id is only used in the Record Route Object for recording the proxy-ingress. If no proxy-ingress-id is specified, then a local interface address that will not otherwise be included in the Record Route Object can be used. A similar technique is used in [RFC4090 Sec 6.1.1].

- 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 subobject 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.

- o A connection between backup ingress and primary ingress: If there is not any direct link between the primary ingress and the backup ingress, a tunnel MUST be configured between them.

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

5.2.1. Relay-Message Method

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

1. Select a PATH message P0 for the LSP.

2. If the backup ingress is off-path (the backup ingress is not the next hop of the primary ingress for P0), then send it a PATH message P0' with the content from P0 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 subobject, the Backup Ingress Address subobject and the Label-Routes subobject. The options is set to indicate whether a Backup P2MP LSP is desired. The Label-Routes subobject contains the next-hops of the primary ingress and their labels. Note that for on-path case, there is an existing PATH message to the backup ingress (i.e., the next hop), and we just add an INGRESS_PROTECTION object into the existing PATH message to be sent to the backup ingress. We do not send a separate PATH message to the backup ingress for this existing PATH message.

3. For each Pi of the other PATH messages for the LSP, send the backup ingress a PATH message Pi' with the content copied from Pi and an empty INGRESS_PROTECTION object.

For every PATH message Pj' (i.e., P0'/Pi') to be sent to the backup ingress, it has the same SESSION as Pj (i.e., P0/Pi). If the backup ingress is off-path, the primary ingress updates Pj' according to the backup ingress as its next hop before sending it. It adds the backup ingress to the beginning of the ERO, and sets RSVP_HOP based on the interface to the backup ingress. The primary ingress MUST NOT set up any forwarding state to the backup ingress if the backup ingress is off-path.

5.2.2. Proxy-Ingress Method

The primary ingress is responsible for starting the RSVP signaling for the proxy-ingress node. To do this, the following MUST be done for the RSVP PATH message.

1. Compute the EROs for the LSP as normal for the ingress.

2. If the selected backup ingress node is not the first node on the path (for all sub-LSPs), then insert at the beginning of the ERO first the backup ingress node and then the ingress node.

3. In the PATH RRO, instead of recording the ingress node’s address, replace it with the Proxy-Ingress-Id.

4. Leave the HOP object populated as usual with information for the ingress-node.

5. Add the INGRESS_PROTECTION object to the PATH message. Include the Backup Ingress Address (IPv4 or IPv6) subobject and the Traffic-Descriptor subobject. Set or clear the options indicating that a Backup P2MP LSP is desired.

6. Optionally, add the FAST-REROUTE object [RFC4090] to the Path message. Indicate whether one-to-one backup is desired. Indicate whether facility backup is desired.

7. The RSVP PATH message is sent to the backup node as normal.

If the ingress detects that it can’t communicate with the backup ingress, then the ingress SHOULD instead send the PATH message to the next-hop indicated in the ERO computed in step 1. Once the ingress detects that it can communicate with the backup ingress, the ingress SHOULD follow the steps 1-7 to obtain ingress failure protection.

When the ingress node receives an RSVP PATH message with an INGRESS_PROTECTION object and the object specifies that node as the ingress node and the PHOP as the backup ingress node, the ingress node SHOULD remove the INGRESS_PROTECTION object from the PATH message before sending it out. Additionally, the ingress node MUST store that it will install ingress forwarding state for the LSP rather than midpoint forwarding.

When an RSVP RESV message is received by the ingress, it uses the NHOP to determine whether the message is received from the backup ingress or from a different node. The stored associated PATH message contains an INGRESS_PROTECTION object that identifies the backup ingress node. If the RESV message is not from the backup node, then ingress forwarding state SHOULD be set up, and the INGRESS_PROTECTION object MUST be added to the RESV before it is sent to the NHOP, which SHOULD be the backup node. If the RESV message is from the backup node, then the LSP SHOULD be considered available for use.

If the backup ingress node is on the forwarding path, then a RESV is received with an INGRESS_PROTECTION object and an NHOP that matches the backup ingress. In this case, the ingress node's address will not appear after the backup ingress in the RRO. The ingress node SHOULD set up ingress forwarding state, just as is done if the LSP weren't ingress-node protected.

5.3. 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 subobject 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. The LER determines whether it uses Relay-Message Method or Proxy-Ingress Method according to configurations.

5.3.1. 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 groups 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" and set NUB to the number of the merge points to which there is no backup LSP.

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.

5.3.1.1. Relay-Message Method

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 subobject 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 message to the PATH message received from the primary ingress. If the backup ingress is off-path, the LABEL object in the RESV message contains IMPLICIT-NULL. 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.

5.3.1.2. Proxy-Ingress Method

The backup ingress determines the next-hops to be merged to by collecting the set of the pair of (IPv4/IPv6 subobject, Label subobject) from the Record Route Object of each RESV that are closest to the top and not the Ingress router; this should be the second to the top pair. If a Label-Routes subobject is included in the INGRESS_PROTECTION object, the included IPv4/IPv6 subobjects are used to filter the set down to the specific next-hops where protection is desired. A RESV message MUST have been received before the Backup Ingress can create or select the appropriate backup LSP.

When the backup ingress receives a PATH message with the INGRESS_PROTECTION object, the backup ingress examines the object to learn what traffic associated with the LSP. The backup ingress forwards the PATH message to the ingress node with the normal RSVP changes.

When the backup ingress receives a RESV message with the INGRESS_PROTECTION object, the backup ingress records an IMPLICIT-NULL label in the RRO. Then the backup ingress forwards the RESV message to the ingress node, which is acting for the proxy ingress.

5.3.2. 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 (and for Proxy-Ingress Method the primary ingress is not its next hop via checking the PATH message with the INGRESS_PROTECTION object received from 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 subobject.

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.

5.3.3. 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 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.

5.4. 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.

5.4.1. 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.

5.4.2. 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. 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.

6. Security Considerations

In principle this document does not introduce new security issues. The security considerations pertaining to RFC 4090, RFC 4875, RFC 2205 and RFC 3209 remain relevant.

7. Compatibility

This extension reuses and extends semantics and procedures defined in RFC 2205, RFC 3209, RFC 4090 and RFC 4875 to support ingress protection. The new object defined to indicate ingress protection has a class number of the form 0bbbbbbb. Per RFC 2205, a node not supporting this extension will not recognize the new class number and should respond with an "Unknown Object Class" error. The error message will propagate to the ingress, which can then take action to avoid the incompatible node as a backup ingress or may simply terminate the session.

8. IANA Considerations

This document does not request any IANA actions.

9. Co-authors and Contributors

1. Co-authors

Autumn Liu
Ciena
USA
Email: hliu@ciena.com

Zhenbin Li
Huawei Technologies
Email: zhenbin.li@huawei.com

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

2. Contributors

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

Mehmet Toy
Verizon
USA
Email: mehmet.toy@verizon.com

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

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

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, Alia Atlas, Olufemi Komolafe, Rob Rennison, Neil Harrison, Kannan Sampath, Gregory Mirsky, and Ronhazli Adam for their valuable comments and suggestions on this draft.

11. References

11.1. Normative References

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

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

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

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090,

DOI 10.17487/RFC4090, May 2005,
<https://www.rfc-editor.org/info/rfc4090>.

[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>.

11.2. Informative References

[RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <https://www.rfc-editor.org/info/rfc6378>.

Authors' Addresses

Huaimo Chen (editor)
Huawei Technologies
Boston, MA
USA

Email: huaimo.chen@huawei.com

Raveendra Torvi (editor)
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA

Email: rtorvi@juniper.net

TEAS Working Group T. Saad
Internet-Draft Juniper Networks
Intended status: Standards Track R. Gandhi
Expires: August 26, 2021 Cisco Systems Inc
X. Liu
Volta Networks
V. Beeram
Juniper Networks
I. Bryskin
Individual
O. Gonzalez de Dios
Telefonica
February 22, 2021

A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces
draft-ietf-teas-yang-te-26

Abstract

This document defines a YANG data model for the configuration and management of Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs). and interfaces. The model is divided into YANG modules that classify data into generic, device-specific, technology agnostic, and technology-specific elements.

This model covers data for configuration, operational state, remote procedural calls, and event notifications.

Status of This Memo

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 Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model covers data applicable to generic or device-independent, device-specific, and Multiprotocol Label Switching (MPLS) technology specific.

The document describes a high-level relationship between the modules defined in this document, as well as other external protocol YANG modules. The TE generic YANG data model does not include any data specific to a signaling protocol. It is expected other data plane technology model(s) will augment the TE generic YANG data model.

Also, it is expected other YANG module(s) that model TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) will augment the generic TE YANG module.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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

2.1. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te-types	ietf-te-types	[RFC8776]
te-packet-types	ietf-te-packet-types	[RFC8776]
te	ietf-te	this document
te-dev	ietf-te-device	this document

Table 1: Prefixes and corresponding YANG modules

2.2. Model Tree Diagrams

The tree diagrams extracted from the module(s) defined in this document are given in subsequent sections as per the syntax defined in [RFC8340].

3. Design Considerations

This document describes a generic TE YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technology models to reuse the TE generic data model and possibly augment it with technology specific data.

The elements of the generic TE YANG data model, including TE tunnels, LSPs, and interfaces have leaf(s) that identify the technology layer where they reside. For example, the LSP encoding type can identify the technology associated with a TE tunnel or LSP.

Also, the generic TE YANG data model does not cover signaling protocol data. The signaling protocol used to instantiate TE LSPs are outside the scope of this document and expected to be covered by augmentations defined in other document(s).

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

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

- 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 Suitable defaults are specified for all configurable elements.

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

3.1. State Data Organization

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

4. Model Overview

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

4.1. Module Relationship

The generic TE YANG data model that is defined in "ietf-te.yang" covers the building blocks that are device independent and agnostic of any specific technology or control plane instances. The TE device model defined in "ietf-te-device.yang" augments the generic TE YANG data model and covers data that is specific to a device - for example, attributes of TE interfaces, or TE timers that are local to a TE node.

The TE data model for specific instances of data plane technology exist in a separate YANG module(s) that augment the generic TE YANG data model. For example, the MPLS-TE module "ietf-te-mpls.yang" is defined in another document and augments the TE generic model as shown in Figure 1.

The TE data model for specific instances of signaling protocol are outside the scope of this document and are defined in other documents. For example, the RSVP-TE YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp].

```
     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

5. TE YANG Model

The generic TE YANG module ('ietf-te') is meant to manage and operate a TE network. This includes creating, modifying and retrieving TE tunnels, LSPs, and interfaces and their associated attributes (e.g. Administrative-Groups, SRLGs, etc.).

The detailed tree structure is provided in Figure 2.

5.1. Module Structure

The 'ietf-te' uses three main containers grouped under the main 'te' container (see Figure 2).

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

The 'globals' container maintains the set of global TE attributes that can be applicable to TE tunnel(s) and interface(s).

The 'tunnels' container includes the list of TE tunnels that are instantiated.

The 'lsps' container includes the list of TE LSP(s) that are instantiated.

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

      +--rw tunnels
         .
         .

      +-- lsps

rpcs:
   +---x tunnels-path-compute
   +---x tunnels-action
```

Figure 2: TE generic high level model view

5.1.1. TE Globals

The 'globals' container covers properties that control TE features behavior system-wide, and its respective state (see Figure 3). The TE globals configuration include:

- o Table of named (extended) administrative groups mappings
- o Table of named SRLG 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

```
+--rw globals
|  +--rw named-admin-groups
|  |  +--rw named-admin-group* [name]
..
|  +--rw named-srlgs
|  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
..
|  +--rw named-path-constraints
|  |  +--rw named-path-constraint* [name]
..
```

Figure 3: TE globals tree structure

5.1.2. TE Tunnels

The set of TE tunnels are provisioned under the 'tunnels' container (see Figure 4). A TE tunnel in the list is uniquely identified by a name. When the model is used to manage a specific device, the 'tunnels' list contains the TE tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnels' list contains all TE tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

The TE tunnel has a number of attributes that are set directly under the tunnel. The 'encoding' and 'switching-type' nodes define the specific technology that the tunnel operates in.

```
       +--rw tunnels
       |  +--rw tunnel* [name]
       |     +--ro operational-state?           identityref
       |     +--rw name                         string
       |     +--rw identifier?                  uint32
       |     +--rw description?                 string
       |     +--rw encoding?                    identityref
       |     +--rw switching-type?              identityref
       |     +--rw admin-state?                 identityref
       |     +--rw reoptimize-timer?            uint16
       |     +--rw source?                      te-types:te-node-id
       |     +--rw destination?                 te-types:te-node-id
       |     +--rw src-tunnel-tp-id?            binary
       |     +--rw dst-tunnel-tp-id?            binary
       |     +--rw bidirectional?               boolean
```

Figure 4: TE tunnel list structure

The TE tunnel has the following main attributes:

association-objects:

A container that includes the set of associations of the TE tunnel with other TE tunnels.

protection:

A container that includes the TE tunnel protection properties.

restoration:

A container that includes the TE tunnel restoration properties.

primary-paths:

A container that includes the set of primary paths (see Figure 5). A primary path is identified by 'name'. A primary path is selected from the list to instantiate an LSP for the tunnel. The list of primary paths is visited by order of preference. A primary path has the following attributes:

* primary-reverse-path: A container that includes properties of the primary reverse path. The reverse path is applicable to bidirectional TE tunnels.

* candidate-secondary-paths: A container that includes a list of candidate secondary paths which may be used for the primary path. The candidate secondary path(s) reference path(s)m from the secondary paths list. The preference of the secondary paths is specified within the list and dictates the order of visiting the secondary path from the list.

* compute-only: A primary path of TE tunnel is, by default, provisioned so that it can is instantiated in forwarding to carry traffic as soon as a valid path is compute. In some cases, 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 path is configured in 'compute-only' mode to distinguish it from the default behavior. A 'compute-only' path 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. A client may query the 'compute-only' computed path properties 'on-demand', or alternatively, can subscribe to be notified of computed path(s) and whenever the path properties change.

```
+--rw primary-paths
|  +--rw primary-path* [name]
|     +--rw name                              string
|     +--rw path-computation-method?          identityref
|     +--rw path-computation-server
|     |  +--rw id?     te-generic-node-id
|     |  +--rw type?   enumeration
|     +--rw compute-only?                     empty
|     +--rw use-path-computation?             boolean
|     +--rw lockdown?                         empty
|     +--ro path-scope?                       identityref
|     +--rw preference?                       uint8
|     +--rw k-requested-paths?                uint8
```

Figure 5: TE tunnel primary paths.

secondary-paths:

A container that includes the set of secondary paths. The secondary paths are identified by 'name'. A secondary path can be reference by a TE tunnel's 'candidate-secondary-path'. A secondary path contains attributes similar to a primary path.

hierarchy:

A container that includes hierarchy related properties of the tunnel (see Figure 6. A TE LSP can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used as a TE links to carry traffic in other (client) networks [RFC6107]. In this case, the model introduces the TE tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the underlying TE tunnel is associated with. The hierarchy container includes the following:

* dependency-tunnels: a hierarchical TE tunnel provisioned or to be provisioned in an immediately adjacent server layer a given client layer TE tunnel depends on for multi-layer path computation. A dependency TE tunnel is provisioned if and only if it is used (selected by path computation) at least by one client layer TE tunnel. The TE link in the client layer network topology supported by a dependent TE tunnel is dynamically created only when the dependency TE tunnel is actually provisioned.

* hierarchical-link: A container that includes the identity of a hierarchical link (in client layer) that this tunnel is associated with.

* te-topology-identifier: A container that includes the topology identifier associated with the tunnel.

```
       +--rw hierarchy
       |  +--rw dependency-tunnels
       |  |  +--rw dependency-tunnel* [name]
       |  |     +--rw name
       |  |     |       -> ../../../../../../tunnels/tunnel/name
       |  |     +--rw encoding?         identityref
       |  |     +--rw switching-type?   identityref
       |  +--rw hierarchical-link
       |     +--rw local-te-node-id?         te-types:te-node-id
       |     +--rw local-te-link-tp-id?      te-types:te-tp-id
       |     +--rw remote-te-node-id?        te-types:te-node-id
       |     +--rw te-topology-identifier
       |        +--rw provider-id?   te-global-id
       |        +--rw client-id?     te-global-id
       |        +--rw topology-id?   te-topology-id
```

Figure 6: TE tunnel primary paths.

5.1.3. TE LSPs

The 'lsps' container includes the set of TE LSP(s) that are instantiated. A TE LSP is identified by a 3-tuple ('tunnel-name', 'node', 'lsp-id'). When the model is used to manage a specific device, the 'lsps' list contains all TE LSP(s) that traverse the device (including ingressing, transiting and egressing the device).

5.2. Tree Diagram

Figure 7 shows the tree diagram of the generic TE YANG model defined in modules 'ietf-te.yang'.

```
module: ietf-te
  +--rw te!
     +--rw globals
     |  +--rw named-admin-groups
     |  |  +--rw named-admin-group* [name]
     |  |          {te-types:extended-admin-groups,te-types:named-extend
ed-admin-groups}?
     |  |     +--rw name            string
     |  |     +--rw bit-position?   uint32
     |  +--rw named-srlgs
     |  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
     |  |     +--rw name     string
     |  |     +--rw value?   te-types:srlg
     |  |     +--rw cost?    uint32
     |  +--rw named-path-constraints
     |     +--rw named-path-constraint* [name]
     |             {te-types:named-path-constraints}?
```

```
   |         +--rw name                                 string
   |         +--rw te-bandwidth
   |         |  +--rw (technology)?
   |         |     +--:(generic)
   |         |        +--rw generic?   te-bandwidth
   |         +--rw link-protection?                     identityref
   |         +--rw setup-priority?                      uint8
   |         +--rw hold-priority?                       uint8
   |         +--rw signaling-type?                      identityref
   |         +--rw path-metric-bounds
   |         |  +--rw path-metric-bound* [metric-type]
   |         |     +--rw metric-type    identityref
   |         |     +--rw upper-bound?   uint64
   |         +--rw path-affinities-values
   |         |  +--rw path-affinities-value* [usage]
   |         |     +--rw usage    identityref
   |         |     +--rw value?   admin-groups
   |         +--rw path-affinity-names
   |         |  +--rw path-affinity-name* [usage]
   |         |     +--rw usage            identityref
   |         |     +--rw affinity-name* [name]
   |         |        +--rw name    string
   |         +--rw path-srlgs-lists
   |         |  +--rw path-srlgs-list* [usage]
   |         |     +--rw usage     identityref
   |         |     +--rw values*   srlg
   |         +--rw path-srlgs-names
   |         |  +--rw path-srlgs-name* [usage]
   |         |     +--rw usage    identityref
   |         |     +--rw names*   string
   |         +--rw disjointness?
   |         |       te-path-disjointness
   |         +--rw explicit-route-objects-always
   |         |  +--rw route-object-exclude-always* [index]
   |         |  |  +--rw index                          uint32
   |         |  |  +--rw (type)?
   |         |  |     +--:(numbered-node-hop)
   |         |  |     |  +--rw numbered-node-hop
   |         |  |     |     +--rw node-id     te-node-id
   |         |  |     |     +--rw hop-type?   te-hop-type
   |         |  |     +--:(numbered-link-hop)
   |         |  |     |  +--rw numbered-link-hop
   |         |  |     |     +--rw link-tp-id    te-tp-id
   |         |  |     |     +--rw hop-type?     te-hop-type
   |         |  |     |     +--rw direction?    te-link-direction
   |         |  |     +--:(unnumbered-link-hop)
   |         |  |     |  +--rw unnumbered-link-hop
   |         |  |     |     +--rw link-tp-id    te-tp-id
```

```
     |        |  |     |     +--rw node-id       te-node-id
     |        |  |     |     +--rw hop-type?     te-hop-type
     |        |  |     |     +--rw direction?    te-link-direction
     |        |  |     +--:(as-number)
     |        |  |     |  +--rw as-number-hop
     |        |  |     |     +--rw as-number    inet:as-number
     |        |  |     |     +--rw hop-type?    te-hop-type
     |        |  |     +--:(label)
     |        |  |        +--rw label-hop
     |        |  |           +--rw te-label
     |        |  |              +--rw (technology)?
     |        |  |              |  +--:(generic)
     |        |  |              |     +--rw generic?
     |        |  |              |             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 path-in-segment!
     |        |  +--rw label-restrictions
     |        |     +--rw label-restriction* [index]
     |        |        +--rw restriction?    enumeration
     |        |        +--rw index           uint32
     |        |        +--rw label-start
     |        |        |  +--rw te-label
     |        |        |     +--rw (technology)?
     |        |        |     |  +--:(generic)
     |        |        |     |     +--rw generic?
     |        |        |     |             rt-types:generalized-label
     |        |        |     +--rw direction?
     |        |        |             te-label-direction
     |        |        +--rw label-end
     |        |        |  +--rw te-label
     |        |        |     +--rw (technology)?
     |        |        |     |  +--:(generic)
     |        |        |     |     +--rw generic?
     |        |        |     |             rt-types:generalized-label
     |        |        |     +--rw direction?
     |        |        |             te-label-direction
     |        |        +--rw label-step
     |        |        |  +--rw (technology)?
     |        |        |     +--:(generic)
     |        |        |        +--rw generic?   int32
     |        |        +--rw range-bitmap?   yang:hex-string
     |        +--rw path-out-segment!
     |           +--rw label-restrictions
     |              +--rw label-restriction* [index]
     |                 +--rw restriction?    enumeration
     |                 +--rw index           uint32
     |                 +--rw label-start
     |                 |  +--rw te-label
     |                 |     +--rw (technology)?
     |                 |     |  +--:(generic)
     |                 |     |     +--rw generic?
     |                 |     |             rt-types:generalized-label
     |                 |     +--rw direction?
     |                 |             te-label-direction
     |                 +--rw label-end
     |                 |  +--rw te-label
     |                 |     +--rw (technology)?
     |                 |     |  +--:(generic)
     |                 |     |     +--rw generic?
     |                 |     |             rt-types:generalized-label
```

```
     |                   |     +--rw direction?
     |                   |             te-label-direction
     |                   +--rw label-step
     |                   |  +--rw (technology)?
     |                   |     +--:(generic)
     |                   |        +--rw generic?   int32
     |                   +--rw range-bitmap?   yang:hex-string
     +--rw tunnels
     |  +--rw tunnel* [name]
     |     +--ro operational-state?             identityref
     |     +--rw name                           string
     |     +--rw identifier?                    uint32
     |     +--rw description?                   string
     |     +--rw encoding?                      identityref
     |     +--rw switching-type?                identityref
     |     +--rw admin-state?                   identityref
     |     +--rw reoptimize-timer?              uint16
     |     +--rw source?                        te-types:te-node-id
     |     +--rw destination?                   te-types:te-node-id
     |     +--rw controller
     |     |  +--rw protocol-origin?        identityref
     |     |  +--rw controller-entity-id?   string
     |     +--rw src-tunnel-tp-id?              binary
     |     +--rw dst-tunnel-tp-id?              binary
     |     +--rw bidirectional?                 boolean
     |     +--rw association-objects
     |     |  +--rw association-object* [association-key]
     |     |  |  +--rw association-key    string
     |     |  |  +--rw type?              identityref
     |     |  |  +--rw id?                uint16
     |     |  |  +--rw source
     |     |  |     +--rw id?     te-gen-node-id
     |     |  |     +--rw type?   enumeration
     |     |  +--rw association-object-extended* [association-key]
     |     |     +--rw association-key    string
     |     |     +--rw type?              identityref
     |     |     +--rw id?                uint16
     |     |     +--rw source
     |     |     |  +--rw id?     te-gen-node-id
     |     |     |  +--rw type?   enumeration
     |     |     +--rw global-source?     uint32
     |     |     +--rw extended-id?       yang:hex-string
     |     +--rw protection
     |     |  +--rw enable?                         boolean
     |     |  +--rw protection-type?                identityref
     |     |  +--rw protection-reversion-disable?   boolean
     |     |  +--rw hold-off-time?                  uint32
     |     |  +--rw wait-to-revert?                 uint16
```

```
     |     |  +--rw aps-signal-id?                 uint8
     |     +--rw restoration
     |     |  +--rw enable?                         boolean
     |     |  +--rw restoration-type?               identityref
     |     |  +--rw restoration-scheme?             identityref
     |     |  +--rw restoration-reversion-disable?  boolean
     |     |  +--rw hold-off-time?                  uint32
     |     |  +--rw wait-to-restore?                uint16
     |     |  +--rw wait-to-revert?                 uint16
     |     +--rw te-topology-identifier
     |     |  +--rw provider-id?   te-global-id
     |     |  +--rw client-id?     te-global-id
     |     |  +--rw topology-id?   te-topology-id
     |     +--rw te-bandwidth
     |     |  +--rw (technology)?
     |     |     +--:(generic)
     |     |        +--rw generic?   te-bandwidth
     |     +--rw link-protection?              identityref
     |     +--rw setup-priority?               uint8
     |     +--rw hold-priority?                uint8
     |     +--rw signaling-type?               identityref
     |     +--rw hierarchy
     |     |  +--rw dependency-tunnels
     |     |  |  +--rw dependency-tunnel* [name]
     |     |  |     +--rw name
     |     |  |     |       -> /te/tunnels/tunnel/name
     |     |  |     +--rw encoding?         identityref
     |     |  |     +--rw switching-type?   identityref
     |     |  +--rw hierarchical-link
     |     |     +--rw local-te-node-id?         te-types:te-node-id
     |     |     +--rw local-te-link-tp-id?      te-types:te-tp-id
     |     |     +--rw remote-te-node-id?        te-types:te-node-id
     |     |     +--rw te-topology-identifier
     |     |        +--rw provider-id?   te-global-id
     |     |        +--rw client-id?     te-global-id
     |     |        +--rw topology-id?   te-topology-id
     |     +--rw primary-paths
     |     |  +--rw primary-path* [name]
     |     |     +--rw name                             string
     |     |     +--rw path-computation-method?         identityref
     |     |     +--rw path-computation-server
     |     |     |  +--rw id?     te-gen-node-id
     |     |     |  +--rw type?   enumeration
     |     |     +--rw compute-only?                    empty
     |     |     +--rw use-path-computation?            boolean
     |     |     +--rw lockdown?                        empty
     |     |     +--ro path-scope?                      identityref
     |     |     +--rw preference?                      uint8
```

```
     |     |     +--rw k-requested-paths?               uint8
     |     |     +--rw association-objects
     |     |     |  +--rw association-object* [association-key]
     |     |     |  |  +--rw association-key    string
     |     |     |  |  +--rw type?              identityref
     |     |     |  |  +--rw id?                uint16
     |     |     |  |  +--rw source
     |     |     |  |     +--rw id?     te-gen-node-id
     |     |     |  |     +--rw type?   enumeration
     |     |     |  +--rw association-object-extended*
     |     |     |          [association-key]
     |     |     |     +--rw association-key    string
     |     |     |     +--rw type?              identityref
     |     |     |     +--rw id?                uint16
     |     |     |     +--rw source
     |     |     |     |  +--rw id?     te-gen-node-id
     |     |     |     |  +--rw type?   enumeration
     |     |     |     +--rw global-source?     uint32
     |     |     |     +--rw extended-id?       yang:hex-string
     |     |     +--rw optimizations
     |     |     |  +--rw (algorithm)?
     |     |     |     +--:(metric) {path-optimization-metric}?
     |     |     |     |  +--rw optimization-metric* [metric-type]
     |     |     |     |  |  +--rw metric-type
     |     |     |     |  |  |       identityref
     |     |     |     |  |  +--rw weight?
     |     |     |     |  |  |       uint8
     |     |     |     |  |  +--rw explicit-route-exclude-objects
     |     |     |     |  |  |  +--rw route-object-exclude-object*
     |     |     |     |  |  |          [index]
     |     |     |     |  |  |     +--rw index
     |     |     |     |  |  |     |       uint32
     |     |     |     |  |  |     +--rw (type)?
     |     |     |     |  |  |        +--:(numbered-node-hop)
     |     |     |     |  |  |        |  +--rw numbered-node-hop
     |     |     |     |  |  |        |     +--rw node-id
     |     |     |     |  |  |        |     |       te-node-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |             te-hop-type
     |     |     |     |  |  |        +--:(numbered-link-hop)
     |     |     |     |  |  |        |  +--rw numbered-link-hop
     |     |     |     |  |  |        |     +--rw link-tp-id
     |     |     |     |  |  |        |     |       te-tp-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |     |       te-hop-type
     |     |     |     |  |  |        |     +--rw direction?
     |     |     |     |  |  |        |             te-link-direction
     |     |     |     |  |  |        +--:(unnumbered-link-hop)
```

```
     |        |        |        |    |    |           |  +--rw unnumbered-link-hop
     |        |        |        |    |    |           |     +--rw link-tp-id
     |        |        |        |    |    |           |     |       te-tp-id
     |        |        |        |    |    |           |     +--rw node-id
     |        |        |        |    |    |           |     |       te-node-id
     |        |        |        |    |    |           |     +--rw hop-type?
     |        |        |        |    |    |           |     |       te-hop-type
     |        |        |        |    |    |           |     +--rw direction?
     |        |        |        |    |    |           |             te-link-direction
     |        |        |        |    |    |           +--:(as-number)
     |        |        |        |    |    |           |  +--rw as-number-hop
     |        |        |        |    |    |           |     +--rw as-number
     |        |        |        |    |    |           |     |       inet:as-number
     |        |        |        |    |    |           |     +--rw hop-type?
     |        |        |        |    |    |           |             te-hop-type
     |        |        |        |    |    |           +--:(label)
     |        |        |        |    |    |           |  +--rw label-hop
     |        |        |        |    |    |           |     +--rw te-label
     |        |        |        |    |    |           |        +--rw (technology)?
     |        |        |        |    |    |           |        |  +--:(generic)
     |        |        |        |    |    |           |        |     +--rw generic?
     |        |        |        |    |    |           |        |             rt-types:ge
neralized-label
     |        |        |        |    |    |           |        +--rw direction?
     |        |        |        |    |    |           |                te-label-directio
n
     |        |        |        |    |    |           +--:(srlg)
     |        |        |        |    |    |              +--rw srlg
     |        |        |        |    |    |                 +--rw srlg?   uint32
     |        |        |        |    |    +--rw explicit-route-include-objects
     |        |        |        |    |       +--rw route-object-include-object*
     |        |        |        |    |               [index]
     |        |        |        |    |          +--rw index
     |        |        |        |    |          |       uint32
     |        |        |        |    |          +--rw (type)?
     |        |        |        |    |             +--:(numbered-node-hop)
     |        |        |        |    |             |  +--rw numbered-node-hop
     |        |        |        |    |             |     +--rw node-id
     |        |        |        |    |             |     |       te-node-id
     |        |        |        |    |             |     +--rw hop-type?
     |        |        |        |    |             |             te-hop-type
     |        |        |        |    |             +--:(numbered-link-hop)
     |        |        |        |    |             |  +--rw numbered-link-hop
     |        |        |        |    |             |     +--rw link-tp-id
     |        |        |        |    |             |     |       te-tp-id
     |        |        |        |    |             |     +--rw hop-type?
     |        |        |        |    |             |     |       te-hop-type
     |        |        |        |    |             |     +--rw direction?
```

```
     |     |     |     |  |           |             te-link-direction
     |     |     |     |  |           +--:(unnumbered-link-hop)
     |     |     |     |  |           |  +--rw unnumbered-link-hop
     |     |     |     |  |           |     +--rw link-tp-id
     |     |     |     |  |           |     |       te-tp-id
     |     |     |     |  |           |     +--rw node-id
     |     |     |     |  |           |     |       te-node-id
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |     |       te-hop-type
     |     |     |     |  |           |     +--rw direction?
     |     |     |     |  |           |             te-link-direction
     |     |     |     |  |           +--:(as-number)
     |     |     |     |  |           |  +--rw as-number-hop
     |     |     |     |  |           |     +--rw as-number
     |     |     |     |  |           |     |       inet:as-number
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |             te-hop-type
     |     |     |     |  |           +--:(label)
     |     |     |     |  |              +--rw label-hop
     |     |     |     |  |                 +--rw te-label
     |     |     |     |  |                    +--rw (technology)?
     |     |     |     |  |                    |  +--:(generic)
     |     |     |     |  |                    |     +--rw generic?
     |     |     |     |  |                    |             rt-types:ge
neralized-label
     |     |     |     |  |                    +--rw direction?
     |     |     |     |  |                            te-label-directio
n
     |     |     |     |  +--rw tiebreakers
     |     |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |     |     |     |        +--rw tiebreaker-type    identityref
     |     |     |     +--:(objective-function)
     |     |     |              {path-optimization-objective-function}?
     |     |     |        +--rw objective-function
     |     |     |           +--rw objective-function-type?
     |     |     |                   identityref
     |     |     +--rw named-path-constraint?           leafref
     |     |     |       {te-types:named-path-constraints}?
     |     |     +--rw te-bandwidth
     |     |     |  +--rw (technology)?
     |     |     |     +--:(generic)
     |     |     |        +--rw generic?   te-bandwidth
     |     |     +--rw link-protection?                 identityref
     |     |     +--rw setup-priority?                  uint8
     |     |     +--rw hold-priority?                   uint8
     |     |     +--rw signaling-type?                  identityref
     |     |     +--rw path-metric-bounds
     |     |     |  +--rw path-metric-bound* [metric-type]
```

```
     |     |     |     +--rw metric-type    identityref
     |     |     |     +--rw upper-bound?   uint64
     |     |     +--rw path-affinities-values
     |     |     |  +--rw path-affinities-value* [usage]
     |     |     |     +--rw usage    identityref
     |     |     |     +--rw value?   admin-groups
     |     |     +--rw path-affinity-names
     |     |     |  +--rw path-affinity-name* [usage]
     |     |     |     +--rw usage            identityref
     |     |     |     +--rw affinity-name* [name]
     |     |     |        +--rw name    string
     |     |     +--rw path-srlgs-lists
     |     |     |  +--rw path-srlgs-list* [usage]
     |     |     |     +--rw usage     identityref
     |     |     |     +--rw values*   srlg
     |     |     +--rw path-srlgs-names
     |     |     |  +--rw path-srlgs-name* [usage]
     |     |     |     +--rw usage    identityref
     |     |     |     +--rw names*   string
     |     |     +--rw disjointness?
     |     |     |       te-path-disjointness
     |     |     +--rw explicit-route-objects-always
     |     |     |  +--rw route-object-exclude-always* [index]
     |     |     |  |  +--rw index                        uint32
     |     |     |  |  +--rw (type)?
     |     |     |  |     +--:(numbered-node-hop)
     |     |     |  |     |  +--rw numbered-node-hop
     |     |     |  |     |     +--rw node-id     te-node-id
     |     |     |  |     |     +--rw hop-type?   te-hop-type
     |     |     |  |     +--:(numbered-link-hop)
     |     |     |  |     |  +--rw numbered-link-hop
     |     |     |  |     |     +--rw link-tp-id    te-tp-id
     |     |     |  |     |     +--rw hop-type?     te-hop-type
     |     |     |  |     |     +--rw direction?    te-link-direction
     |     |     |  |     +--:(unnumbered-link-hop)
     |     |     |  |     |  +--rw unnumbered-link-hop
     |     |     |  |     |     +--rw link-tp-id    te-tp-id
     |     |     |  |     |     +--rw node-id       te-node-id
     |     |     |  |     |     +--rw hop-type?     te-hop-type
     |     |     |  |     |     +--rw direction?    te-link-direction
     |     |     |  |     +--:(as-number)
     |     |     |  |     |  +--rw as-number-hop
     |     |     |  |     |     +--rw as-number    inet:as-number
     |     |     |  |     |     +--rw hop-type?    te-hop-type
     |     |     |  |     +--:(label)
     |     |     |  |        +--rw label-hop
     |     |     |  |           +--rw te-label
     |     |     |  |              +--rw (technology)?
```

```
     |        |        |  |              |  +--:(generic)
     |        |        |  |              |     +--rw generic?
     |        |        |  |              |             rt-types:generalized-la
bel
     |        |        |  |              +--rw direction?
     |        |        |  |                      te-label-direction
     |        |        |  +--rw route-object-include-exclude* [index]
     |        |        |     +--rw explicit-route-usage?        identityref
     |        |        |     +--rw index                        uint32
     |        |        |     +--rw (type)?
     |        |        |        +--:(numbered-node-hop)
     |        |        |        |  +--rw numbered-node-hop
     |        |        |        |     +--rw node-id     te-node-id
     |        |        |        |     +--rw hop-type?   te-hop-type
     |        |        |        +--:(numbered-link-hop)
     |        |        |        |  +--rw numbered-link-hop
     |        |        |        |     +--rw link-tp-id    te-tp-id
     |        |        |        |     +--rw hop-type?     te-hop-type
     |        |        |        |     +--rw direction?    te-link-direction
     |        |        |        +--:(unnumbered-link-hop)
     |        |        |        |  +--rw unnumbered-link-hop
     |        |        |        |     +--rw link-tp-id    te-tp-id
     |        |        |        |     +--rw node-id       te-node-id
     |        |        |        |     +--rw hop-type?     te-hop-type
     |        |        |        |     +--rw direction?    te-link-direction
     |        |        |        +--:(as-number)
     |        |        |        |  +--rw as-number-hop
     |        |        |        |     +--rw as-number    inet:as-number
     |        |        |        |     +--rw hop-type?    te-hop-type
     |        |        |        +--:(label)
     |        |        |        |  +--rw label-hop
     |        |        |        |     +--rw te-label
     |        |        |        |        +--rw (technology)?
     |        |        |        |        |  +--:(generic)
     |        |        |        |        |     +--rw generic?
     |        |        |        |        |             rt-types:generalized-la
bel
     |        |        |        |        +--rw direction?
     |        |        |        |                te-label-direction
     |        |        |        +--:(srlg)
     |        |        |           +--rw srlg
     |        |        |              +--rw srlg?   uint32
     |        |        +--rw path-in-segment!
     |        |        |  +--rw label-restrictions
     |        |        |     +--rw label-restriction* [index]
     |        |        |        +--rw restriction?    enumeration
     |        |        |        +--rw index           uint32
     |        |        |        +--rw label-start
```

```
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-end
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-step
     |     |     |        |  +--rw (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--rw generic?   int32
     |     |     |        +--rw range-bitmap?   yang:hex-string
     |     |     +--rw path-out-segment!
     |     |     |  +--rw label-restrictions
     |     |     |     +--rw label-restriction* [index]
     |     |     |        +--rw restriction?    enumeration
     |     |     |        +--rw index           uint32
     |     |     |        +--rw label-start
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-end
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-step
     |     |     |        |  +--rw (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--rw generic?   int32
     |     |     |        +--rw range-bitmap?   yang:hex-string
     |     |     +--ro computed-paths-properties
     |     |     |  +--ro computed-path-properties* [k-index]
```

```
     |        |        |     +--ro k-index             uint8
     |        |        |     +--ro path-properties
     |        |        |        +--ro path-metric* [metric-type]
     |        |        |        |  +--ro metric-type           identityref
     |        |        |        |  +--ro accumulative-value?   uint64
     |        |        |        +--ro path-affinities-values
     |        |        |        |  +--ro path-affinities-value* [usage]
     |        |        |        |     +--ro usage    identityref
     |        |        |        |     +--ro value?   admin-groups
     |        |        |        +--ro path-affinity-names
     |        |        |        |  +--ro path-affinity-name* [usage]
     |        |        |        |     +--ro usage            identityref
     |        |        |        |     +--ro affinity-name* [name]
     |        |        |        |        +--ro name    string
     |        |        |        +--ro path-srlgs-lists
     |        |        |        |  +--ro path-srlgs-list* [usage]
     |        |        |        |     +--ro usage     identityref
     |        |        |        |     +--ro values*   srlg
     |        |        |        +--ro path-srlgs-names
     |        |        |        |  +--ro path-srlgs-name* [usage]
     |        |        |        |     +--ro usage    identityref
     |        |        |        |     +--ro names*   string
     |        |        |        +--ro path-route-objects
     |        |        |        |  +--ro path-route-object* [index]
     |        |        |        |     +--ro index
     |        |        |        |     |       uint32
     |        |        |        |     +--ro (type)?
     |        |        |        |        +--:(numbered-node-hop)
     |        |        |        |        |  +--ro numbered-node-hop
     |        |        |        |        |     +--ro node-id     te-node-id
     |        |        |        |        |     +--ro hop-type?
     |        |        |        |        |             te-hop-type
     |        |        |        |        +--:(numbered-link-hop)
     |        |        |        |        |  +--ro numbered-link-hop
     |        |        |        |        |     +--ro link-tp-id    te-tp-id
     |        |        |        |        |     +--ro hop-type?
     |        |        |        |        |     |       te-hop-type
     |        |        |        |        |     +--ro direction?
     |        |        |        |        |             te-link-direction
     |        |        |        |        +--:(unnumbered-link-hop)
     |        |        |        |        |  +--ro unnumbered-link-hop
     |        |        |        |        |     +--ro link-tp-id    te-tp-id
     |        |        |        |        |     +--ro node-id
     |        |        |        |        |     |       te-node-id
     |        |        |        |        |     +--ro hop-type?
     |        |        |        |        |     |       te-hop-type
     |        |        |        |        |     +--ro direction?
     |        |        |        |        |             te-link-direction
```

```
     |     |     |        |        +--:(as-number)
     |     |     |        |        |  +--ro as-number-hop
     |     |     |        |        |     +--ro as-number
     |     |     |        |        |     |       inet:as-number
     |     |     |        |        |     +--ro hop-type?
     |     |     |        |        |             te-hop-type
     |     |     |        |        +--:(label)
     |     |     |        |           +--ro label-hop
     |     |     |        |              +--ro te-label
     |     |     |        |                 +--ro (technology)?
     |     |     |        |                 |  +--:(generic)
     |     |     |        |                 |     +--ro generic?
     |     |     |        |                 |             rt-types:gener
alized-label
     |     |     |        |                 +--ro direction?
     |     |     |        |                         te-label-direction
     |     |     |        +--ro te-bandwidth
     |     |     |        |  +--ro (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--ro generic?   te-bandwidth
     |     |     |        +--ro disjointness-type?
     |     |     |                te-types:te-path-disjointness
     |     |     +--ro computed-path-error-infos
     |     |     |  +--ro computed-path-error-info* []
     |     |     |     +--ro error-description?   string
     |     |     |     +--ro error-timestamp?     yang:date-and-time
     |     |     |     +--ro error-reason?        identityref
     |     |     +--ro lsp-provisioning-error-infos
     |     |     |  +--ro lsp-provisioning-error-info* []
     |     |     |     +--ro error-description?   string
     |     |     |     +--ro error-timestamp?     yang:date-and-time
     |     |     |     +--ro error-node-id?       te-types:te-node-id
     |     |     |     +--ro error-link-id?       te-types:te-tp-id
     |     |     |     +--ro lsp-id?              uint16
     |     |     +--ro lsps
     |     |     |  +--ro lsp* [node lsp-id]
     |     |     |     +--ro tunnel-name?
     |     |     |     |       -> /te/lsps/lsp/tunnel-name
     |     |     |     +--ro node           -> /te/lsps/lsp/node
     |     |     |     +--ro lsp-id         -> /te/lsps/lsp/lsp-id
     |     |     +--rw primary-reverse-path
     |     |     |  +--rw name?                             string
     |     |     |  +--rw path-computation-method?
     |     |     |  |       identityref
     |     |     |  +--rw path-computation-server
     |     |     |  |  +--rw id?     te-gen-node-id
     |     |     |  |  +--rw type?   enumeration
     |     |     |  +--rw compute-only?                     empty
```

```
     |        |        |  +--rw use-path-computation?
     |        |        |  |       boolean
     |        |        |  +--rw lockdown?                            empty
     |        |        |  +--ro path-scope?
     |        |        |  |       identityref
     |        |        |  +--rw association-objects
     |        |        |  |  +--rw association-object* [association-key]
     |        |        |  |  |  +--rw association-key    string
     |        |        |  |  |  +--rw type?              identityref
     |        |        |  |  |  +--rw id?                uint16
     |        |        |  |  |  +--rw source
     |        |        |  |  |     +--rw id?     te-gen-node-id
     |        |        |  |  |     +--rw type?   enumeration
     |        |        |  |  +--rw association-object-extended*
     |        |        |  |          [association-key]
     |        |        |  |     +--rw association-key    string
     |        |        |  |     +--rw type?              identityref
     |        |        |  |     +--rw id?                uint16
     |        |        |  |     +--rw source
     |        |        |  |     |  +--rw id?     te-gen-node-id
     |        |        |  |     |  +--rw type?   enumeration
     |        |        |  |     +--rw global-source?     uint32
     |        |        |  |     +--rw extended-id?       yang:hex-string
     |        |        |  +--rw optimizations
     |        |        |  |  +--rw (algorithm)?
     |        |        |  |     +--:(metric) {path-optimization-metric}?
     |        |        |  |     |  +--rw optimization-metric* [metric-type]
     |        |        |  |     |  |  +--rw metric-type
     |        |        |  |     |  |  |       identityref
     |        |        |  |     |  |  +--rw weight?
     |        |        |  |     |  |  |       uint8
     |        |        |  |     |  |  +--rw explicit-route-exclude-objects
     |        |        |  |     |  |  |  +--rw route-object-exclude-object*
     |        |        |  |     |  |  |          [index]
     |        |        |  |     |  |  |     +--rw index
     |        |        |  |     |  |  |     |       uint32
     |        |        |  |     |  |  |     +--rw (type)?
     |        |        |  |     |  |  |        +--:(numbered-node-hop)
     |        |        |  |     |  |  |        |  +--rw numbered-node-hop
     |        |        |  |     |  |  |        |     +--rw node-id
     |        |        |  |     |  |  |        |     |       te-node-id
     |        |        |  |     |  |  |        |     +--rw hop-type?
     |        |        |  |     |  |  |        |             te-hop-type
     |        |        |  |     |  |  |        +--:(numbered-link-hop)
     |        |        |  |     |  |  |        |  +--rw numbered-link-hop
     |        |        |  |     |  |  |        |     +--rw link-tp-id
     |        |        |  |     |  |  |        |     |       te-tp-id
     |        |        |  |     |  |  |        |     +--rw hop-type?
```

```
     |     |     |  |     |  |  |        |     |       te-hop-type
     |     |     |  |     |  |  |        |     +--rw direction?
     |     |     |  |     |  |  |        |             te-link-direction
     |     |     |  |     |  |  |        +--:(unnumbered-link-hop)
     |     |     |  |     |  |  |        |  +--rw unnumbered-link-hop
     |     |     |  |     |  |  |        |     +--rw link-tp-id
     |     |     |  |     |  |  |        |     |       te-tp-id
     |     |     |  |     |  |  |        |     +--rw node-id
     |     |     |  |     |  |  |        |     |       te-node-id
     |     |     |  |     |  |  |        |     +--rw hop-type?
     |     |     |  |     |  |  |        |     |       te-hop-type
     |     |     |  |     |  |  |        |     +--rw direction?
     |     |     |  |     |  |  |        |             te-link-direction
     |     |     |  |     |  |  |        +--:(as-number)
     |     |     |  |     |  |  |        |  +--rw as-number-hop
     |     |     |  |     |  |  |        |     +--rw as-number
     |     |     |  |     |  |  |        |     |       inet:as-number
     |     |     |  |     |  |  |        |     +--rw hop-type?
     |     |     |  |     |  |  |        |             te-hop-type
     |     |     |  |     |  |  |        +--:(label)
     |     |     |  |     |  |  |        |  +--rw label-hop
     |     |     |  |     |  |  |        |     +--rw te-label
     |     |     |  |     |  |  |        |        +--rw (technology)?
     |     |     |  |     |  |  |        |        |  +--:(generic)
     |     |     |  |     |  |  |        |        |     +--rw generic?
     |     |     |  |     |  |  |        |        |             rt-types
:generalized-label
     |     |     |  |     |  |  |        |        +--rw direction?
     |     |     |  |     |  |  |        |                te-label-direc
tion
     |     |     |  |     |  |  |        +--:(srlg)
     |     |     |  |     |  |  |           +--rw srlg
     |     |     |  |     |  |  |              +--rw srlg?   uint32
     |     |     |  |     |  |  +--rw explicit-route-include-objects
     |     |     |  |     |  |     +--rw route-object-include-object*
     |     |     |  |     |  |             [index]
     |     |     |  |     |  |        +--rw index
     |     |     |  |     |  |        |       uint32
     |     |     |  |     |  |        +--rw (type)?
     |     |     |  |     |  |           +--:(numbered-node-hop)
     |     |     |  |     |  |           |  +--rw numbered-node-hop
     |     |     |  |     |  |           |     +--rw node-id
     |     |     |  |     |  |           |     |       te-node-id
     |     |     |  |     |  |           |     +--rw hop-type?
     |     |     |  |     |  |           |             te-hop-type
     |     |     |  |     |  |           +--:(numbered-link-hop)
     |     |     |  |     |  |           |  +--rw numbered-link-hop
     |     |     |  |     |  |           |     +--rw link-tp-id
```

```
     |     |     |  |     |  |           |     |       te-tp-id
     |     |     |  |     |  |           |     +--rw hop-type?
     |     |     |  |     |  |           |     |       te-hop-type
     |     |     |  |     |  |           |     +--rw direction?
     |     |     |  |     |  |           |             te-link-direction
     |     |     |  |     |  |           +--:(unnumbered-link-hop)
     |     |     |  |     |  |           |  +--rw unnumbered-link-hop
     |     |     |  |     |  |           |     +--rw link-tp-id
     |     |     |  |     |  |           |     |       te-tp-id
     |     |     |  |     |  |           |     +--rw node-id
     |     |     |  |     |  |           |     |       te-node-id
     |     |     |  |     |  |           |     +--rw hop-type?
     |     |     |  |     |  |           |     |       te-hop-type
     |     |     |  |     |  |           |     +--rw direction?
     |     |     |  |     |  |           |             te-link-direction
     |     |     |  |     |  |           +--:(as-number)
     |     |     |  |     |  |           |  +--rw as-number-hop
     |     |     |  |     |  |           |     +--rw as-number
     |     |     |  |     |  |           |     |       inet:as-number
     |     |     |  |     |  |           |     +--rw hop-type?
     |     |     |  |     |  |           |             te-hop-type
     |     |     |  |     |  |           +--:(label)
     |     |     |  |     |  |              +--rw label-hop
     |     |     |  |     |  |                 +--rw te-label
     |     |     |  |     |  |                    +--rw (technology)?
     |     |     |  |     |  |                    |  +--:(generic)
     |     |     |  |     |  |                    |     +--rw generic?
     |     |     |  |     |  |                    |             rt-types
:generalized-label
     |     |     |  |     |  |                    +--rw direction?
     |     |     |  |     |  |                            te-label-direc
tion
     |     |     |  |     |  +--rw tiebreakers
     |     |     |  |     |     +--rw tiebreaker* [tiebreaker-type]
     |     |     |  |     |        +--rw tiebreaker-type
     |     |     |  |     |                identityref
     |     |     |  |     +--:(objective-function)
     |     |     |  |              {path-optimization-objective-function
}?
     |     |     |  |        +--rw objective-function
     |     |     |  |           +--rw objective-function-type?
     |     |     |  |                   identityref
     |     |     |  +--rw named-path-constraint?               leafref
     |     |     |  |       {te-types:named-path-constraints}?
     |     |     |  +--rw te-bandwidth
     |     |     |  |  +--rw (technology)?
     |     |     |  |     +--:(generic)
     |     |     |  |        +--rw generic?   te-bandwidth
```

```
     |     |     |  +--rw link-protection?
     |     |     |  |       identityref
     |     |     |  +--rw setup-priority?                    uint8
     |     |     |  +--rw hold-priority?                     uint8
     |     |     |  +--rw signaling-type?
     |     |     |  |       identityref
     |     |     |  +--rw path-metric-bounds
     |     |     |  |  +--rw path-metric-bound* [metric-type]
     |     |     |  |     +--rw metric-type    identityref
     |     |     |  |     +--rw upper-bound?   uint64
     |     |     |  +--rw path-affinities-values
     |     |     |  |  +--rw path-affinities-value* [usage]
     |     |     |  |     +--rw usage    identityref
     |     |     |  |     +--rw value?   admin-groups
     |     |     |  +--rw path-affinity-names
     |     |     |  |  +--rw path-affinity-name* [usage]
     |     |     |  |     +--rw usage            identityref
     |     |     |  |     +--rw affinity-name* [name]
     |     |     |  |        +--rw name    string
     |     |     |  +--rw path-srlgs-lists
     |     |     |  |  +--rw path-srlgs-list* [usage]
     |     |     |  |     +--rw usage     identityref
     |     |     |  |     +--rw values*   srlg
     |     |     |  +--rw path-srlgs-names
     |     |     |  |  +--rw path-srlgs-name* [usage]
     |     |     |  |     +--rw usage    identityref
     |     |     |  |     +--rw names*   string
     |     |     |  +--rw disjointness?
     |     |     |  |       te-path-disjointness
     |     |     |  +--rw explicit-route-objects-always
     |     |     |  |  +--rw route-object-exclude-always* [index]
     |     |     |  |  |  +--rw index                         uint32
     |     |     |  |  |  +--rw (type)?
     |     |     |  |  |     +--:(numbered-node-hop)
     |     |     |  |  |     |  +--rw numbered-node-hop
     |     |     |  |  |     |     +--rw node-id     te-node-id
     |     |     |  |  |     |     +--rw hop-type?   te-hop-type
     |     |     |  |  |     +--:(numbered-link-hop)
     |     |     |  |  |     |  +--rw numbered-link-hop
     |     |     |  |  |     |     +--rw link-tp-id    te-tp-id
     |     |     |  |  |     |     +--rw hop-type?     te-hop-type
     |     |     |  |  |     |     +--rw direction?
     |     |     |  |  |     |             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 path-in-segment!
     |     |     |  |  +--rw label-restrictions
     |     |     |  |     +--rw label-restriction* [index]
     |     |     |  |        +--rw restriction?    enumeration
     |     |     |  |        +--rw index           uint32
     |     |     |  |        +--rw label-start
     |     |     |  |        |  +--rw te-label
     |     |     |  |        |     +--rw (technology)?
     |     |     |  |        |     |  +--:(generic)
     |     |     |  |        |     |     +--rw generic?
     |     |     |  |        |     |             rt-types:generalized-la
bel
     |     |     |  |        |     +--rw direction?
     |     |     |  |        |             te-label-direction
     |     |     |  |        +--rw label-end
     |     |     |  |        |  +--rw te-label
     |     |     |  |        |     +--rw (technology)?
     |     |     |  |        |     |  +--:(generic)
     |     |     |  |        |     |     +--rw generic?
     |     |     |  |        |     |             rt-types:generalized-la
bel
     |     |     |  |        |     +--rw direction?
     |     |     |  |        |             te-label-direction
     |     |     |  |        +--rw label-step
     |     |     |  |        |  +--rw (technology)?
     |     |     |  |        |     +--:(generic)
     |     |     |  |        |        +--rw generic?   int32
     |     |     |  |        +--rw range-bitmap?   yang:hex-string
     |     |     |  +--rw path-out-segment!
     |     |     |  |  +--rw label-restrictions
     |     |     |  |     +--rw label-restriction* [index]
     |     |     |  |        +--rw restriction?    enumeration
     |     |     |  |        +--rw index           uint32
     |     |     |  |        +--rw label-start
     |     |     |  |        |  +--rw te-label
     |     |     |  |        |     +--rw (technology)?
     |     |     |  |        |     |  +--:(generic)
     |     |     |  |        |     |     +--rw generic?
     |     |     |  |        |     |             rt-types:generalized-la
bel
     |     |     |  |        |     +--rw direction?
```

```
     |     |     |  |        |             te-label-direction
     |     |     |  |        +--rw label-end
     |     |     |  |        |  +--rw te-label
     |     |     |  |        |     +--rw (technology)?
     |     |     |  |        |     |  +--:(generic)
     |     |     |  |        |     |     +--rw generic?
     |     |     |  |        |     |             rt-types:generalized-la
bel
     |     |     |  |        |     +--rw direction?
     |     |     |  |        |             te-label-direction
     |     |     |  |        +--rw label-step
     |     |     |  |        |  +--rw (technology)?
     |     |     |  |        |     +--:(generic)
     |     |     |  |        |        +--rw generic?   int32
     |     |     |  |        +--rw range-bitmap?   yang:hex-string
     |     |     |  +--ro computed-paths-properties
     |     |     |  |  +--ro computed-path-properties* [k-index]
     |     |     |  |     +--ro k-index            uint8
     |     |     |  |     +--ro path-properties
     |     |     |  |        +--ro path-metric* [metric-type]
     |     |     |  |        |  +--ro metric-type
     |     |     |  |        |  |       identityref
     |     |     |  |        |  +--ro accumulative-value?   uint64
     |     |     |  |        +--ro path-affinities-values
     |     |     |  |        |  +--ro path-affinities-value* [usage]
     |     |     |  |        |     +--ro usage    identityref
     |     |     |  |        |     +--ro value?   admin-groups
     |     |     |  |        +--ro path-affinity-names
     |     |     |  |        |  +--ro path-affinity-name* [usage]
     |     |     |  |        |     +--ro usage            identityref
     |     |     |  |        |     +--ro affinity-name* [name]
     |     |     |  |        |        +--ro name    string
     |     |     |  |        +--ro path-srlgs-lists
     |     |     |  |        |  +--ro path-srlgs-list* [usage]
     |     |     |  |        |     +--ro usage     identityref
     |     |     |  |        |     +--ro values*   srlg
     |     |     |  |        +--ro path-srlgs-names
     |     |     |  |        |  +--ro path-srlgs-name* [usage]
     |     |     |  |        |     +--ro usage    identityref
     |     |     |  |        |     +--ro names*   string
     |     |     |  |        +--ro path-route-objects
     |     |     |  |        |  +--ro path-route-object* [index]
     |     |     |  |        |     +--ro index
     |     |     |  |        |     |       uint32
     |     |     |  |        |     +--ro (type)?
     |     |     |  |        |        +--:(numbered-node-hop)
     |     |     |  |        |        |  +--ro numbered-node-hop
     |     |     |  |        |        |     +--ro node-id
```

```
     |     |     |  |        |        |     |       te-node-id
     |     |     |  |        |        |     +--ro hop-type?
     |     |     |  |        |        |             te-hop-type
     |     |     |  |        |        +--:(numbered-link-hop)
     |     |     |  |        |        |  +--ro numbered-link-hop
     |     |     |  |        |        |     +--ro link-tp-id
     |     |     |  |        |        |     |       te-tp-id
     |     |     |  |        |        |     +--ro hop-type?
     |     |     |  |        |        |     |       te-hop-type
     |     |     |  |        |        |     +--ro direction?
     |     |     |  |        |        |             te-link-direction
     |     |     |  |        |        +--:(unnumbered-link-hop)
     |     |     |  |        |        |  +--ro unnumbered-link-hop
     |     |     |  |        |        |     +--ro link-tp-id
     |     |     |  |        |        |     |       te-tp-id
     |     |     |  |        |        |     +--ro node-id
     |     |     |  |        |        |     |       te-node-id
     |     |     |  |        |        |     +--ro hop-type?
     |     |     |  |        |        |     |       te-hop-type
     |     |     |  |        |        |     +--ro direction?
     |     |     |  |        |        |             te-link-direction
     |     |     |  |        |        +--:(as-number)
     |     |     |  |        |        |  +--ro as-number-hop
     |     |     |  |        |        |     +--ro as-number
     |     |     |  |        |        |     |       inet:as-number
     |     |     |  |        |        |     +--ro hop-type?
     |     |     |  |        |        |             te-hop-type
     |     |     |  |        |        +--:(label)
     |     |     |  |        |           +--ro label-hop
     |     |     |  |        |              +--ro te-label
     |     |     |  |        |                 +--ro (technology)?
     |     |     |  |        |                 |  +--:(generic)
     |     |     |  |        |                 |     +--ro generic?
     |     |     |  |        |                 |             rt-types:ge
neralized-label
     |     |     |  |        |                 +--ro direction?
     |     |     |  |        |                         te-label-directio
n
     |     |     |  |        +--ro te-bandwidth
     |     |     |  |        |  +--ro (technology)?
     |     |     |  |        |     +--:(generic)
     |     |     |  |        |        +--ro generic?   te-bandwidth
     |     |     |  |        +--ro disjointness-type?
     |     |     |  |                te-types:te-path-disjointness
     |     |     |  +--ro computed-path-error-infos
     |     |     |  |  +--ro computed-path-error-info* []
     |     |     |  |     +--ro error-description?   string
     |     |     |  |     +--ro error-timestamp?
```

```
     |        |        |  |     |       yang:date-and-time
     |        |        |  |     +--ro error-reason?       identityref
     |        |        |  +--ro lsp-provisioning-error-infos
     |        |        |  |  +--ro lsp-provisioning-error-info* []
     |        |        |  |     +--ro error-description?   string
     |        |        |  |     +--ro error-timestamp?
     |        |        |  |     |       yang:date-and-time
     |        |        |  |     +--ro error-node-id?
     |        |        |  |     |       te-types:te-node-id
     |        |        |  |     +--ro error-link-id?
     |        |        |  |     |       te-types:te-tp-id
     |        |        |  |     +--ro lsp-id?              uint16
     |        |        |  +--ro lsps
     |        |        |  |  +--ro lsp* [node lsp-id]
     |        |        |  |     +--ro tunnel-name?
     |        |        |  |     |       -> /te/lsps/lsp/tunnel-name
     |        |        |  |     +--ro node           -> /te/lsps/lsp/node
     |        |        |  |     +--ro lsp-id         -> /te/lsps/lsp/lsp-id
     |        |        |  +--rw candidate-secondary-reverse-paths
     |        |        |     +--rw candidate-secondary-reverse-path*
     |        |        |             [secondary-path]
     |        |        |        +--rw secondary-path    leafref
     |        |        +--rw candidate-secondary-paths
     |        |           +--rw candidate-secondary-path* [secondary-path]
     |        |              +--rw secondary-path    leafref
     |        |              +--ro active?           boolean
     |        +--rw secondary-paths
     |        |  +--rw secondary-path* [name]
     |        |     +--rw name                                  string
     |        |     +--rw path-computation-method?         identityref
     |        |     +--rw path-computation-server
     |        |     |  +--rw id?     te-gen-node-id
     |        |     |  +--rw type?   enumeration
     |        |     +--rw compute-only?                    empty
     |        |     +--rw use-path-computation?            boolean
     |        |     +--rw lockdown?                        empty
     |        |     +--ro path-scope?                      identityref
     |        |     +--rw preference?                      uint8
     |        |     +--rw association-objects
     |        |     |  +--rw association-object* [association-key]
     |        |     |  |  +--rw association-key    string
     |        |     |  |  +--rw type?              identityref
     |        |     |  |  +--rw id?                uint16
     |        |     |  |  +--rw source
     |        |     |  |     +--rw id?     te-gen-node-id
     |        |     |  |     +--rw type?   enumeration
     |        |     |  +--rw association-object-extended*
     |        |     |          [association-key]
```

```
     |     |     |     +--rw association-key    string
     |     |     |     +--rw type?              identityref
     |     |     |     +--rw id?                uint16
     |     |     |     +--rw source
     |     |     |     |  +--rw id?     te-gen-node-id
     |     |     |     |  +--rw type?   enumeration
     |     |     |     +--rw global-source?     uint32
     |     |     |     +--rw extended-id?       yang:hex-string
     |     |     +--rw optimizations
     |     |     |  +--rw (algorithm)?
     |     |     |     +--:(metric) {path-optimization-metric}?
     |     |     |     |  +--rw optimization-metric* [metric-type]
     |     |     |     |  |  +--rw metric-type
     |     |     |     |  |  |       identityref
     |     |     |     |  |  +--rw weight?
     |     |     |     |  |  |       uint8
     |     |     |     |  |  +--rw explicit-route-exclude-objects
     |     |     |     |  |  |  +--rw route-object-exclude-object*
     |     |     |     |  |  |          [index]
     |     |     |     |  |  |     +--rw index
     |     |     |     |  |  |     |       uint32
     |     |     |     |  |  |     +--rw (type)?
     |     |     |     |  |  |        +--:(numbered-node-hop)
     |     |     |     |  |  |        |  +--rw numbered-node-hop
     |     |     |     |  |  |        |     +--rw node-id
     |     |     |     |  |  |        |     |       te-node-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |             te-hop-type
     |     |     |     |  |  |        +--:(numbered-link-hop)
     |     |     |     |  |  |        |  +--rw numbered-link-hop
     |     |     |     |  |  |        |     +--rw link-tp-id
     |     |     |     |  |  |        |     |       te-tp-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |     |       te-hop-type
     |     |     |     |  |  |        |     +--rw direction?
     |     |     |     |  |  |        |             te-link-direction
     |     |     |     |  |  |        +--:(unnumbered-link-hop)
     |     |     |     |  |  |        |  +--rw unnumbered-link-hop
     |     |     |     |  |  |        |     +--rw link-tp-id
     |     |     |     |  |  |        |     |       te-tp-id
     |     |     |     |  |  |        |     +--rw node-id
     |     |     |     |  |  |        |     |       te-node-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |     |       te-hop-type
     |     |     |     |  |  |        |     +--rw direction?
     |     |     |     |  |  |        |             te-link-direction
     |     |     |     |  |  |        +--:(as-number)
     |     |     |     |  |  |        |  +--rw as-number-hop
```

```
     |        |        |        |    |    |        |     +--rw as-number
     |        |        |        |    |    |        |     |       inet:as-number
     |        |        |        |    |    |        |     +--rw hop-type?
     |        |        |        |    |    |        |             te-hop-type
     |        |        |        |    |    |        +--:(label)
     |        |        |        |    |    |        |  +--rw label-hop
     |        |        |        |    |    |        |     +--rw te-label
     |        |        |        |    |    |        |        +--rw (technology)?
     |        |        |        |    |    |        |        |  +--:(generic)
     |        |        |        |    |    |        |        |     +--rw generic?
     |        |        |        |    |    |        |        |             rt-types:ge
neralized-label
     |        |        |        |    |    |        |        +--rw direction?
     |        |        |        |    |    |        |                te-label-directio
n
     |        |        |        |    |    |        +--:(srlg)
     |        |        |        |    |    |           +--rw srlg
     |        |        |        |    |    |              +--rw srlg?   uint32
     |        |        |        |    |    +--rw explicit-route-include-objects
     |        |        |        |    |       +--rw route-object-include-object*
     |        |        |        |    |               [index]
     |        |        |        |    |          +--rw index
     |        |        |        |    |          |       uint32
     |        |        |        |    |          +--rw (type)?
     |        |        |        |    |             +--:(numbered-node-hop)
     |        |        |        |    |             |  +--rw numbered-node-hop
     |        |        |        |    |             |     +--rw node-id
     |        |        |        |    |             |     |       te-node-id
     |        |        |        |    |             |     +--rw hop-type?
     |        |        |        |    |             |             te-hop-type
     |        |        |        |    |             +--:(numbered-link-hop)
     |        |        |        |    |             |  +--rw numbered-link-hop
     |        |        |        |    |             |     +--rw link-tp-id
     |        |        |        |    |             |     |       te-tp-id
     |        |        |        |    |             |     +--rw hop-type?
     |        |        |        |    |             |     |       te-hop-type
     |        |        |        |    |             |     +--rw direction?
     |        |        |        |    |             |             te-link-direction
     |        |        |        |    |             +--:(unnumbered-link-hop)
     |        |        |        |    |             |  +--rw unnumbered-link-hop
     |        |        |        |    |             |     +--rw link-tp-id
     |        |        |        |    |             |     |       te-tp-id
     |        |        |        |    |             |     +--rw node-id
     |        |        |        |    |             |     |       te-node-id
     |        |        |        |    |             |     +--rw hop-type?
     |        |        |        |    |             |     |       te-hop-type
     |        |        |        |    |             |     +--rw direction?
     |        |        |        |    |             |             te-link-direction
```

```
     |     |     |     |  |           +--:(as-number)
     |     |     |     |  |           |  +--rw as-number-hop
     |     |     |     |  |           |     +--rw as-number
     |     |     |     |  |           |     |       inet:as-number
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |             te-hop-type
     |     |     |     |  |           +--:(label)
     |     |     |     |  |              +--rw label-hop
     |     |     |     |  |                 +--rw te-label
     |     |     |     |  |                    +--rw (technology)?
     |     |     |     |  |                    |  +--:(generic)
     |     |     |     |  |                    |     +--rw generic?
     |     |     |     |  |                    |             rt-types:ge
neralized-label
     |     |     |     |  |                    +--rw direction?
     |     |     |     |  |                            te-label-directio
n
     |     |     |     |  +--rw tiebreakers
     |     |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |     |     |     |        +--rw tiebreaker-type    identityref
     |     |     |     +--:(objective-function)
     |     |     |              {path-optimization-objective-function}?
     |     |     |        +--rw objective-function
     |     |     |           +--rw objective-function-type?
     |     |     |                   identityref
     |     |     +--rw named-path-constraint?           leafref
     |     |     |       {te-types:named-path-constraints}?
     |     |     +--rw te-bandwidth
     |     |     |  +--rw (technology)?
     |     |     |     +--:(generic)
     |     |     |        +--rw generic?   te-bandwidth
     |     |     +--rw link-protection?                 identityref
     |     |     +--rw setup-priority?                  uint8
     |     |     +--rw hold-priority?                   uint8
     |     |     +--rw signaling-type?                  identityref
     |     |     +--rw path-metric-bounds
     |     |     |  +--rw path-metric-bound* [metric-type]
     |     |     |     +--rw metric-type    identityref
     |     |     |     +--rw upper-bound?   uint64
     |     |     +--rw path-affinities-values
     |     |     |  +--rw path-affinities-value* [usage]
     |     |     |     +--rw usage    identityref
     |     |     |     +--rw value?   admin-groups
     |     |     +--rw path-affinity-names
     |     |     |  +--rw path-affinity-name* [usage]
     |     |     |     +--rw usage            identityref
     |     |     |     +--rw affinity-name* [name]
     |     |     |        +--rw name    string
```

```
     |        |        +--rw path-srlgs-lists
     |        |        |  +--rw path-srlgs-list* [usage]
     |        |        |     +--rw usage     identityref
     |        |        |     +--rw values*   srlg
     |        |        +--rw path-srlgs-names
     |        |        |  +--rw path-srlgs-name* [usage]
     |        |        |     +--rw usage    identityref
     |        |        |     +--rw names*   string
     |        |        +--rw disjointness?
     |        |        |       te-path-disjointness
     |        |        +--rw explicit-route-objects-always
     |        |        |  +--rw route-object-exclude-always* [index]
     |        |        |  |  +--rw index                         uint32
     |        |        |  |  +--rw (type)?
     |        |        |  |     +--:(numbered-node-hop)
     |        |        |  |     |  +--rw numbered-node-hop
     |        |        |  |     |     +--rw node-id     te-node-id
     |        |        |  |     |     +--rw hop-type?   te-hop-type
     |        |        |  |     +--:(numbered-link-hop)
     |        |        |  |     |  +--rw numbered-link-hop
     |        |        |  |     |     +--rw link-tp-id    te-tp-id
     |        |        |  |     |     +--rw hop-type?     te-hop-type
     |        |        |  |     |     +--rw direction?    te-link-direction
     |        |        |  |     +--:(unnumbered-link-hop)
     |        |        |  |     |  +--rw unnumbered-link-hop
     |        |        |  |     |     +--rw link-tp-id    te-tp-id
     |        |        |  |     |     +--rw node-id       te-node-id
     |        |        |  |     |     +--rw hop-type?     te-hop-type
     |        |        |  |     |     +--rw direction?    te-link-direction
     |        |        |  |     +--:(as-number)
     |        |        |  |     |  +--rw as-number-hop
     |        |        |  |     |     +--rw as-number    inet:as-number
     |        |        |  |     |     +--rw hop-type?    te-hop-type
     |        |        |  |     +--:(label)
     |        |        |  |        +--rw label-hop
     |        |        |  |           +--rw te-label
     |        |        |  |              +--rw (technology)?
     |        |        |  |              |  +--:(generic)
     |        |        |  |              |     +--rw generic?
     |        |        |  |              |             rt-types:generalized-la
bel
     |        |        |  |              +--rw direction?
     |        |        |  |                      te-label-direction
     |        |        |  +--rw route-object-include-exclude* [index]
     |        |        |     +--rw explicit-route-usage?        identityref
     |        |        |     +--rw index                         uint32
     |        |        |     +--rw (type)?
     |        |        |        +--:(numbered-node-hop)
```

```
     |     |     |        |  +--rw numbered-node-hop
     |     |     |        |     +--rw node-id     te-node-id
     |     |     |        |     +--rw hop-type?   te-hop-type
     |     |     |        +--:(numbered-link-hop)
     |     |     |        |  +--rw numbered-link-hop
     |     |     |        |     +--rw link-tp-id    te-tp-id
     |     |     |        |     +--rw hop-type?     te-hop-type
     |     |     |        |     +--rw direction?    te-link-direction
     |     |     |        +--:(unnumbered-link-hop)
     |     |     |        |  +--rw unnumbered-link-hop
     |     |     |        |     +--rw link-tp-id    te-tp-id
     |     |     |        |     +--rw node-id       te-node-id
     |     |     |        |     +--rw hop-type?     te-hop-type
     |     |     |        |     +--rw direction?    te-link-direction
     |     |     |        +--:(as-number)
     |     |     |        |  +--rw as-number-hop
     |     |     |        |     +--rw as-number    inet:as-number
     |     |     |        |     +--rw hop-type?    te-hop-type
     |     |     |        +--:(label)
     |     |     |        |  +--rw label-hop
     |     |     |        |     +--rw te-label
     |     |     |        |        +--rw (technology)?
     |     |     |        |        |  +--:(generic)
     |     |     |        |        |     +--rw generic?
     |     |     |        |        |             rt-types:generalized-la
bel
     |     |     |        |        +--rw direction?
     |     |     |        |                te-label-direction
     |     |     |        +--:(srlg)
     |     |     |           +--rw srlg
     |     |     |              +--rw srlg?   uint32
     |     |     +--rw path-in-segment!
     |     |     |  +--rw label-restrictions
     |     |     |     +--rw label-restriction* [index]
     |     |     |        +--rw restriction?    enumeration
     |     |     |        +--rw index           uint32
     |     |     |        +--rw label-start
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-end
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
```

```
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-step
     |     |     |        |  +--rw (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--rw generic?   int32
     |     |     |        +--rw range-bitmap?   yang:hex-string
     |     |     +--rw path-out-segment!
     |     |     |  +--rw label-restrictions
     |     |     |     +--rw label-restriction* [index]
     |     |     |        +--rw restriction?    enumeration
     |     |     |        +--rw index           uint32
     |     |     |        +--rw label-start
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-end
     |     |     |        |  +--rw te-label
     |     |     |        |     +--rw (technology)?
     |     |     |        |     |  +--:(generic)
     |     |     |        |     |     +--rw generic?
     |     |     |        |     |             rt-types:generalized-label
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-step
     |     |     |        |  +--rw (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--rw generic?   int32
     |     |     |        +--rw range-bitmap?   yang:hex-string
     |     |     +--rw protection
     |     |     |  +--rw enable?                          boolean
     |     |     |  +--rw protection-type?                 identityref
     |     |     |  +--rw protection-reversion-disable?   boolean
     |     |     |  +--rw hold-off-time?                   uint32
     |     |     |  +--rw wait-to-revert?                  uint16
     |     |     |  +--rw aps-signal-id?                   uint8
     |     |     +--rw restoration
     |     |     |  +--rw enable?                           boolean
     |     |     |  +--rw restoration-type?
     |     |     |  |       identityref
     |     |     |  +--rw restoration-scheme?
     |     |     |  |       identityref
```

```
     |        |        |  +--rw restoration-reversion-disable?   boolean
     |        |        |  +--rw hold-off-time?                   uint32
     |        |        |  +--rw wait-to-restore?                 uint16
     |        |        |  +--rw wait-to-revert?                  uint16
     |        |        +--ro computed-paths-properties
     |        |        |  +--ro computed-path-properties* [k-index]
     |        |        |     +--ro k-index            uint8
     |        |        |     +--ro path-properties
     |        |        |        +--ro path-metric* [metric-type]
     |        |        |        |  +--ro metric-type           identityref
     |        |        |        |  +--ro accumulative-value?   uint64
     |        |        |        +--ro path-affinities-values
     |        |        |        |  +--ro path-affinities-value* [usage]
     |        |        |        |     +--ro usage    identityref
     |        |        |        |     +--ro value?   admin-groups
     |        |        |        +--ro path-affinity-names
     |        |        |        |  +--ro path-affinity-name* [usage]
     |        |        |        |     +--ro usage            identityref
     |        |        |        |     +--ro affinity-name* [name]
     |        |        |        |        +--ro name    string
     |        |        |        +--ro path-srlgs-lists
     |        |        |        |  +--ro path-srlgs-list* [usage]
     |        |        |        |     +--ro usage     identityref
     |        |        |        |     +--ro values*   srlg
     |        |        |        +--ro path-srlgs-names
     |        |        |        |  +--ro path-srlgs-name* [usage]
     |        |        |        |     +--ro usage    identityref
     |        |        |        |     +--ro names*   string
     |        |        |        +--ro path-route-objects
     |        |        |        |  +--ro path-route-object* [index]
     |        |        |        |     +--ro index
     |        |        |        |     |       uint32
     |        |        |        |     +--ro (type)?
     |        |        |        |        +--:(numbered-node-hop)
     |        |        |        |        |  +--ro numbered-node-hop
     |        |        |        |        |     +--ro node-id     te-node-id
     |        |        |        |        |     +--ro hop-type?
     |        |        |        |        |             te-hop-type
     |        |        |        |        +--:(numbered-link-hop)
     |        |        |        |        |  +--ro numbered-link-hop
     |        |        |        |        |     +--ro link-tp-id    te-tp-id
     |        |        |        |        |     +--ro hop-type?
     |        |        |        |        |     |       te-hop-type
     |        |        |        |        |     +--ro direction?
     |        |        |        |        |             te-link-direction
     |        |        |        |        +--:(unnumbered-link-hop)
     |        |        |        |        |  +--ro unnumbered-link-hop
     |        |        |        |        |     +--ro link-tp-id    te-tp-id
```

```
     |        |       |        |        |     +--ro node-id
     |        |       |        |        |     |       te-node-id
     |        |       |        |        |     +--ro hop-type?
     |        |       |        |        |     |       te-hop-type
     |        |       |        |        |     +--ro direction?
     |        |       |        |        |             te-link-direction
     |        |       |        |        +--:(as-number)
     |        |       |        |        |  +--ro as-number-hop
     |        |       |        |        |     +--ro as-number
     |        |       |        |        |     |       inet:as-number
     |        |       |        |        |     +--ro hop-type?
     |        |       |        |        |             te-hop-type
     |        |       |        |        +--:(label)
     |        |       |        |           +--ro label-hop
     |        |       |        |              +--ro te-label
     |        |       |        |                 +--ro (technology)?
     |        |       |        |                 |  +--:(generic)
     |        |       |        |                 |     +--ro generic?
     |        |       |        |                 |             rt-types:gener
alized-label
     |        |       |        |                 +--ro direction?
     |        |       |        |                         te-label-direction
     |        |       |        +--ro te-bandwidth
     |        |       |        |  +--ro (technology)?
     |        |       |        |     +--:(generic)
     |        |       |        |        +--ro generic?   te-bandwidth
     |        |       |        +--ro disjointness-type?
     |        |       |                te-types:te-path-disjointness
     |        |       +--ro computed-path-error-infos
     |        |       |  +--ro computed-path-error-info* []
     |        |       |     +--ro error-description?   string
     |        |       |     +--ro error-timestamp?     yang:date-and-time
     |        |       |     +--ro error-reason?        identityref
     |        |       +--ro lsp-provisioning-error-infos
     |        |       |  +--ro lsp-provisioning-error-info* []
     |        |       |     +--ro error-description?   string
     |        |       |     +--ro error-timestamp?     yang:date-and-time
     |        |       |     +--ro error-node-id?       te-types:te-node-id
     |        |       |     +--ro error-link-id?       te-types:te-tp-id
     |        |       |     +--ro lsp-id?              uint16
     |        |       +--ro lsps
     |        |          +--ro lsp* [node lsp-id]
     |        |             +--ro tunnel-name?
     |        |             |       -> /te/lsps/lsp/tunnel-name
     |        |             +--ro node           -> /te/lsps/lsp/node
     |        |             +--ro lsp-id         -> /te/lsps/lsp/lsp-id
     |        +--rw secondary-reverse-paths
     |        |  +--rw secondary-reverse-path* [name]
```

```
     |        |       +--rw name                            string
     |        |       +--rw path-computation-method?        identityref
     |        |       +--rw path-computation-server
     |        |       |  +--rw id?     te-gen-node-id
     |        |       |  +--rw type?   enumeration
     |        |       +--rw compute-only?                   empty
     |        |       +--rw use-path-computation?           boolean
     |        |       +--rw lockdown?                       empty
     |        |       +--ro path-scope?                     identityref
     |        |       +--rw preference?                     uint8
     |        |       +--rw association-objects
     |        |       |  +--rw association-object* [association-key]
     |        |       |  |  +--rw association-key    string
     |        |       |  |  +--rw type?              identityref
     |        |       |  |  +--rw id?                uint16
     |        |       |  |  +--rw source
     |        |       |  |     +--rw id?     te-gen-node-id
     |        |       |  |     +--rw type?   enumeration
     |        |       |  +--rw association-object-extended*
     |        |       |          [association-key]
     |        |       |     +--rw association-key    string
     |        |       |     +--rw type?              identityref
     |        |       |     +--rw id?                uint16
     |        |       |     +--rw source
     |        |       |     |  +--rw id?     te-gen-node-id
     |        |       |     |  +--rw type?   enumeration
     |        |       |     +--rw global-source?     uint32
     |        |       |     +--rw extended-id?       yang:hex-string
     |        |       +--rw optimizations
     |        |       |  +--rw (algorithm)?
     |        |       |     +--:(metric) {path-optimization-metric}?
     |        |       |     |  +--rw optimization-metric* [metric-type]
     |        |       |     |  |  +--rw metric-type
     |        |       |     |  |  |       identityref
     |        |       |     |  |  +--rw weight?
     |        |       |     |  |  |       uint8
     |        |       |     |  |  +--rw explicit-route-exclude-objects
     |        |       |     |  |  |  +--rw route-object-exclude-object*
     |        |       |     |  |  |          [index]
     |        |       |     |  |  |     +--rw index
     |        |       |     |  |  |     |       uint32
     |        |       |     |  |  |     +--rw (type)?
     |        |       |     |  |  |        +--:(numbered-node-hop)
     |        |       |     |  |  |        |  +--rw numbered-node-hop
     |        |       |     |  |  |        |     +--rw node-id
     |        |       |     |  |  |        |     |       te-node-id
     |        |       |     |  |  |        |     +--rw hop-type?
     |        |       |     |  |  |        |             te-hop-type
```

```
     |     |     |     |  |  |        +--:(numbered-link-hop)
     |     |     |     |  |  |        |  +--rw numbered-link-hop
     |     |     |     |  |  |        |     +--rw link-tp-id
     |     |     |     |  |  |        |     |       te-tp-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |     |       te-hop-type
     |     |     |     |  |  |        |     +--rw direction?
     |     |     |     |  |  |        |             te-link-direction
     |     |     |     |  |  |        +--:(unnumbered-link-hop)
     |     |     |     |  |  |        |  +--rw unnumbered-link-hop
     |     |     |     |  |  |        |     +--rw link-tp-id
     |     |     |     |  |  |        |     |       te-tp-id
     |     |     |     |  |  |        |     +--rw node-id
     |     |     |     |  |  |        |     |       te-node-id
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |     |       te-hop-type
     |     |     |     |  |  |        |     +--rw direction?
     |     |     |     |  |  |        |             te-link-direction
     |     |     |     |  |  |        +--:(as-number)
     |     |     |     |  |  |        |  +--rw as-number-hop
     |     |     |     |  |  |        |     +--rw as-number
     |     |     |     |  |  |        |     |       inet:as-number
     |     |     |     |  |  |        |     +--rw hop-type?
     |     |     |     |  |  |        |             te-hop-type
     |     |     |     |  |  |        +--:(label)
     |     |     |     |  |  |        |  +--rw label-hop
     |     |     |     |  |  |        |     +--rw te-label
     |     |     |     |  |  |        |        +--rw (technology)?
     |     |     |     |  |  |        |        |  +--:(generic)
     |     |     |     |  |  |        |        |     +--rw generic?
     |     |     |     |  |  |        |        |             rt-types:ge
neralized-label
     |     |     |     |  |  |        |        +--rw direction?
     |     |     |     |  |  |        |                te-label-directio
n
     |     |     |     |  |  |        +--:(srlg)
     |     |     |     |  |  |           +--rw srlg
     |     |     |     |  |  |              +--rw srlg?   uint32
     |     |     |     |  |  +--rw explicit-route-include-objects
     |     |     |     |  |     +--rw route-object-include-object*
     |     |     |     |  |             [index]
     |     |     |     |  |        +--rw index
     |     |     |     |  |        |       uint32
     |     |     |     |  |        +--rw (type)?
     |     |     |     |  |           +--:(numbered-node-hop)
     |     |     |     |  |           |  +--rw numbered-node-hop
     |     |     |     |  |           |     +--rw node-id
     |     |     |     |  |           |     |       te-node-id
```

```
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |             te-hop-type
     |     |     |     |  |           +--:(numbered-link-hop)
     |     |     |     |  |           |  +--rw numbered-link-hop
     |     |     |     |  |           |     +--rw link-tp-id
     |     |     |     |  |           |     |       te-tp-id
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |     |       te-hop-type
     |     |     |     |  |           |     +--rw direction?
     |     |     |     |  |           |             te-link-direction
     |     |     |     |  |           +--:(unnumbered-link-hop)
     |     |     |     |  |           |  +--rw unnumbered-link-hop
     |     |     |     |  |           |     +--rw link-tp-id
     |     |     |     |  |           |     |       te-tp-id
     |     |     |     |  |           |     +--rw node-id
     |     |     |     |  |           |     |       te-node-id
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |     |       te-hop-type
     |     |     |     |  |           |     +--rw direction?
     |     |     |     |  |           |             te-link-direction
     |     |     |     |  |           +--:(as-number)
     |     |     |     |  |           |  +--rw as-number-hop
     |     |     |     |  |           |     +--rw as-number
     |     |     |     |  |           |     |       inet:as-number
     |     |     |     |  |           |     +--rw hop-type?
     |     |     |     |  |           |             te-hop-type
     |     |     |     |  |           +--:(label)
     |     |     |     |  |              +--rw label-hop
     |     |     |     |  |                 +--rw te-label
     |     |     |     |  |                    +--rw (technology)?
     |     |     |     |  |                    |  +--:(generic)
     |     |     |     |  |                    |     +--rw generic?
     |     |     |     |  |                    |             rt-types:ge
neralized-label
     |     |     |     |  |                    +--rw direction?
     |     |     |     |  |                            te-label-directio
n
     |     |     |     |  +--rw tiebreakers
     |     |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |     |     |     |        +--rw tiebreaker-type    identityref
     |     |     |     +--:(objective-function)
     |     |     |              {path-optimization-objective-function}?
     |     |     |        +--rw objective-function
     |     |     |           +--rw objective-function-type?
     |     |     |                   identityref
     |     |     +--rw named-path-constraint?           leafref
     |     |     |       {te-types:named-path-constraints}?
     |     |     +--rw te-bandwidth
```

```
     |        |        |  +--rw (technology)?
     |        |        |     +--:(generic)
     |        |        |        +--rw generic?   te-bandwidth
     |        |        +--rw link-protection?                identityref
     |        |        +--rw setup-priority?                 uint8
     |        |        +--rw hold-priority?                  uint8
     |        |        +--rw signaling-type?                 identityref
     |        |        +--rw path-metric-bounds
     |        |        |  +--rw path-metric-bound* [metric-type]
     |        |        |     +--rw metric-type    identityref
     |        |        |     +--rw upper-bound?   uint64
     |        |        +--rw path-affinities-values
     |        |        |  +--rw path-affinities-value* [usage]
     |        |        |     +--rw usage    identityref
     |        |        |     +--rw value?   admin-groups
     |        |        +--rw path-affinity-names
     |        |        |  +--rw path-affinity-name* [usage]
     |        |        |     +--rw usage            identityref
     |        |        |     +--rw affinity-name* [name]
     |        |        |        +--rw name    string
     |        |        +--rw path-srlgs-lists
     |        |        |  +--rw path-srlgs-list* [usage]
     |        |        |     +--rw usage     identityref
     |        |        |     +--rw values*   srlg
     |        |        +--rw path-srlgs-names
     |        |        |  +--rw path-srlgs-name* [usage]
     |        |        |     +--rw usage    identityref
     |        |        |     +--rw names*   string
     |        |        +--rw disjointness?
     |        |        |       te-path-disjointness
     |        |        +--rw explicit-route-objects-always
     |        |        |  +--rw route-object-exclude-always* [index]
     |        |        |  |  +--rw index                          uint32
     |        |        |  |  +--rw (type)?
     |        |        |  |     +--:(numbered-node-hop)
     |        |        |  |     |  +--rw numbered-node-hop
     |        |        |  |     |     +--rw node-id     te-node-id
     |        |        |  |     |     +--rw hop-type?   te-hop-type
     |        |        |  |     +--:(numbered-link-hop)
     |        |        |  |     |  +--rw numbered-link-hop
     |        |        |  |     |     +--rw link-tp-id    te-tp-id
     |        |        |  |     |     +--rw hop-type?     te-hop-type
     |        |        |  |     |     +--rw direction?    te-link-direction
     |        |        |  |     +--:(unnumbered-link-hop)
     |        |        |  |     |  +--rw unnumbered-link-hop
     |        |        |  |     |     +--rw link-tp-id    te-tp-id
     |        |        |  |     |     +--rw node-id       te-node-id
     |        |        |  |     |     +--rw hop-type?     te-hop-type
```

```
     |     |     |  |     |     +--rw direction?    te-link-direction
     |     |     |  |     +--:(as-number)
     |     |     |  |     |  +--rw as-number-hop
     |     |     |  |     |     +--rw as-number    inet:as-number
     |     |     |  |     |     +--rw hop-type?    te-hop-type
     |     |     |  |     +--:(label)
     |     |     |  |        +--rw label-hop
     |     |     |  |           +--rw te-label
     |     |     |  |              +--rw (technology)?
     |     |     |  |              |  +--:(generic)
     |     |     |  |              |     +--rw generic?
     |     |     |  |              |             rt-types:generalized-la
bel
     |     |     |  |              +--rw direction?
     |     |     |  |                      te-label-direction
     |     |     |  +--rw route-object-include-exclude* [index]
     |     |     |     +--rw explicit-route-usage?        identityref
     |     |     |     +--rw index                        uint32
     |     |     |     +--rw (type)?
     |     |     |        +--:(numbered-node-hop)
     |     |     |        |  +--rw numbered-node-hop
     |     |     |        |     +--rw node-id     te-node-id
     |     |     |        |     +--rw hop-type?   te-hop-type
     |     |     |        +--:(numbered-link-hop)
     |     |     |        |  +--rw numbered-link-hop
     |     |     |        |     +--rw link-tp-id    te-tp-id
     |     |     |        |     +--rw hop-type?     te-hop-type
     |     |     |        |     +--rw direction?    te-link-direction
     |     |     |        +--:(unnumbered-link-hop)
     |     |     |        |  +--rw unnumbered-link-hop
     |     |     |        |     +--rw link-tp-id    te-tp-id
     |     |     |        |     +--rw node-id       te-node-id
     |     |     |        |     +--rw hop-type?     te-hop-type
     |     |     |        |     +--rw direction?    te-link-direction
     |     |     |        +--:(as-number)
     |     |     |        |  +--rw as-number-hop
     |     |     |        |     +--rw as-number    inet:as-number
     |     |     |        |     +--rw hop-type?    te-hop-type
     |     |     |        +--:(label)
     |     |     |        |  +--rw label-hop
     |     |     |        |     +--rw te-label
     |     |     |        |        +--rw (technology)?
     |     |     |        |        |  +--:(generic)
     |     |     |        |        |     +--rw generic?
     |     |     |        |        |             rt-types:generalized-la
bel
     |     |     |        |        +--rw direction?
     |     |     |        |                te-label-direction
```

```
     |        |        |         +--:(srlg)
     |        |        |            +--rw srlg
     |        |        |               +--rw srlg?   uint32
     |        |        +--rw path-in-segment!
     |        |        |  +--rw label-restrictions
     |        |        |     +--rw label-restriction* [index]
     |        |        |        +--rw restriction?    enumeration
     |        |        |        +--rw index           uint32
     |        |        |        +--rw label-start
     |        |        |        |  +--rw te-label
     |        |        |        |     +--rw (technology)?
     |        |        |        |     |  +--:(generic)
     |        |        |        |     |     +--rw generic?
     |        |        |        |     |             rt-types:generalized-label
     |        |        |        |     +--rw direction?
     |        |        |        |             te-label-direction
     |        |        |        +--rw label-end
     |        |        |        |  +--rw te-label
     |        |        |        |     +--rw (technology)?
     |        |        |        |     |  +--:(generic)
     |        |        |        |     |     +--rw generic?
     |        |        |        |     |             rt-types:generalized-label
     |        |        |        |     +--rw direction?
     |        |        |        |             te-label-direction
     |        |        |        +--rw label-step
     |        |        |        |  +--rw (technology)?
     |        |        |        |     +--:(generic)
     |        |        |        |        +--rw generic?   int32
     |        |        |        +--rw range-bitmap?   yang:hex-string
     |        |        +--rw path-out-segment!
     |        |        |  +--rw label-restrictions
     |        |        |     +--rw label-restriction* [index]
     |        |        |        +--rw restriction?    enumeration
     |        |        |        +--rw index           uint32
     |        |        |        +--rw label-start
     |        |        |        |  +--rw te-label
     |        |        |        |     +--rw (technology)?
     |        |        |        |     |  +--:(generic)
     |        |        |        |     |     +--rw generic?
     |        |        |        |     |             rt-types:generalized-label
     |        |        |        |     +--rw direction?
     |        |        |        |             te-label-direction
     |        |        |        +--rw label-end
     |        |        |        |  +--rw te-label
     |        |        |        |     +--rw (technology)?
     |        |        |        |     |  +--:(generic)
     |        |        |        |     |     +--rw generic?
     |        |        |        |     |             rt-types:generalized-label
```

```
     |     |     |        |     +--rw direction?
     |     |     |        |             te-label-direction
     |     |     |        +--rw label-step
     |     |     |        |  +--rw (technology)?
     |     |     |        |     +--:(generic)
     |     |     |        |        +--rw generic?   int32
     |     |     |        +--rw range-bitmap?   yang:hex-string
     |     |     +--rw protection
     |     |     |  +--rw enable?                          boolean
     |     |     |  +--rw protection-type?                 identityref
     |     |     |  +--rw protection-reversion-disable?   boolean
     |     |     |  +--rw hold-off-time?                   uint32
     |     |     |  +--rw wait-to-revert?                  uint16
     |     |     |  +--rw aps-signal-id?                   uint8
     |     |     +--rw restoration
     |     |     |  +--rw enable?                           boolean
     |     |     |  +--rw restoration-type?
     |     |     |  |       identityref
     |     |     |  +--rw restoration-scheme?
     |     |     |  |       identityref
     |     |     |  +--rw restoration-reversion-disable?   boolean
     |     |     |  +--rw hold-off-time?                    uint32
     |     |     |  +--rw wait-to-restore?                  uint16
     |     |     |  +--rw wait-to-revert?                   uint16
     |     |     +--ro computed-paths-properties
     |     |     |  +--ro computed-path-properties* [k-index]
     |     |     |     +--ro k-index            uint8
     |     |     |     +--ro path-properties
     |     |     |        +--ro path-metric* [metric-type]
     |     |     |        |  +--ro metric-type           identityref
     |     |     |        |  +--ro accumulative-value?   uint64
     |     |     |        +--ro path-affinities-values
     |     |     |        |  +--ro path-affinities-value* [usage]
     |     |     |        |     +--ro usage    identityref
     |     |     |        |     +--ro value?   admin-groups
     |     |     |        +--ro path-affinity-names
     |     |     |        |  +--ro path-affinity-name* [usage]
     |     |     |        |     +--ro usage            identityref
     |     |     |        |     +--ro affinity-name* [name]
     |     |     |        |        +--ro name    string
     |     |     |        +--ro path-srlgs-lists
     |     |     |        |  +--ro path-srlgs-list* [usage]
     |     |     |        |     +--ro usage     identityref
     |     |     |        |     +--ro values*   srlg
     |     |     |        +--ro path-srlgs-names
     |     |     |        |  +--ro path-srlgs-name* [usage]
     |     |     |        |     +--ro usage    identityref
     |     |     |        |     +--ro names*   string
```

```
     |        |        |           +--ro path-route-objects
     |        |        |           |  +--ro path-route-object* [index]
     |        |        |           |     +--ro index
     |        |        |           |     |       uint32
     |        |        |           |     +--ro (type)?
     |        |        |           |        +--:(numbered-node-hop)
     |        |        |           |        |  +--ro numbered-node-hop
     |        |        |           |        |     +--ro node-id     te-node-id
     |        |        |           |        |     +--ro hop-type?
     |        |        |           |        |             te-hop-type
     |        |        |           |        +--:(numbered-link-hop)
     |        |        |           |        |  +--ro numbered-link-hop
     |        |        |           |        |     +--ro link-tp-id    te-tp-id
     |        |        |           |        |     +--ro hop-type?
     |        |        |           |        |     |       te-hop-type
     |        |        |           |        |     +--ro direction?
     |        |        |           |        |             te-link-direction
     |        |        |           |        +--:(unnumbered-link-hop)
     |        |        |           |        |  +--ro unnumbered-link-hop
     |        |        |           |        |     +--ro link-tp-id    te-tp-id
     |        |        |           |        |     +--ro node-id
     |        |        |           |        |     |       te-node-id
     |        |        |           |        |     +--ro hop-type?
     |        |        |           |        |     |       te-hop-type
     |        |        |           |        |     +--ro direction?
     |        |        |           |        |             te-link-direction
     |        |        |           |        +--:(as-number)
     |        |        |           |        |  +--ro as-number-hop
     |        |        |           |        |     +--ro as-number
     |        |        |           |        |     |       inet:as-number
     |        |        |           |        |     +--ro hop-type?
     |        |        |           |        |             te-hop-type
     |        |        |           |        +--:(label)
     |        |        |           |           +--ro label-hop
     |        |        |           |              +--ro te-label
     |        |        |           |                 +--ro (technology)?
     |        |        |           |                 |  +--:(generic)
     |        |        |           |                 |     +--ro generic?
     |        |        |           |                 |             rt-types:gener
alized-label
     |        |        |           |                 +--ro direction?
     |        |        |           |                         te-label-direction
     |        |        |           +--ro te-bandwidth
     |        |        |           |  +--ro (technology)?
     |        |        |           |     +--:(generic)
     |        |        |           |        +--ro generic?   te-bandwidth
     |        |        |           +--ro disjointness-type?
     |        |        |                   te-types:te-path-disjointness
```

```
     |     |     +--ro computed-path-error-infos
     |     |     |  +--ro computed-path-error-info* []
     |     |     |     +--ro error-description?   string
     |     |     |     +--ro error-timestamp?     yang:date-and-time
     |     |     |     +--ro error-reason?        identityref
     |     |     +--ro lsp-provisioning-error-infos
     |     |     |  +--ro lsp-provisioning-error-info* []
     |     |     |     +--ro error-description?   string
     |     |     |     +--ro error-timestamp?     yang:date-and-time
     |     |     |     +--ro error-node-id?       te-types:te-node-id
     |     |     |     +--ro error-link-id?       te-types:te-tp-id
     |     |     |     +--ro lsp-id?              uint16
     |     |     +--ro lsps
     |     |        +--ro lsp* [node lsp-id]
     |     |           +--ro tunnel-name?
     |     |           |       -> /te/lsps/lsp/tunnel-name
     |     |           +--ro node           -> /te/lsps/lsp/node
     |     |           +--ro lsp-id         -> /te/lsps/lsp/lsp-id
     |     +---x tunnel-action
     |     |  +---w input
     |     |  |  +---w action-type?   identityref
     |     |  +--ro output
     |     |     +--ro action-result?   identityref
     |     +---x protection-external-commands
     |        +---w input
     |           +---w protection-external-command?
     |           |       identityref
     |           +---w protection-group-ingress-node-id?
     |           |       te-types:te-node-id
     |           +---w protection-group-egress-node-id?
     |           |       te-types:te-node-id
     |           +---w path-ref?                          path-ref
     |           +---w traffic-type?
     |           |       enumeration
     |           +---w extra-traffic-tunnel-ref?          tunnel-ref
     +--ro lsps
        +--ro lsp* [tunnel-name lsp-id node]
           +--ro tunnel-name                        string
           +--ro lsp-id                             uint16
           +--ro node
           |       te-types:te-node-id
           +--ro source?
           |       te-types:te-node-id
           +--ro destination?
           |       te-types:te-node-id
           +--ro tunnel-id?                         uint16
           +--ro extended-tunnel-id?                yang:dotted-quad
           +--ro operational-state?                 identityref
```

```
          +--ro signaling-type?                    identityref
          +--ro origin-type?                       enumeration
          +--ro lsp-resource-status?               enumeration
          +--ro lockout-of-normal?                 boolean
          +--ro freeze?                            boolean
          +--ro lsp-protection-role?               enumeration
          +--ro lsp-protection-state?              identityref
          +--ro protection-group-ingress-node-id?
          |       te-types:te-node-id
          +--ro protection-group-egress-node-id?
          |       te-types:te-node-id
          +--ro lsp-record-route-information
             +--ro lsp-record-route-information* [index]
                +--ro index                        uint32
                +--ro (type)?
                   +--:(numbered-node-hop)
                   |  +--ro numbered-node-hop
                   |     +--ro node-id    te-node-id
                   |     +--ro flags*     path-attribute-flags
                   +--:(numbered-link-hop)
                   |  +--ro numbered-link-hop
                   |     +--ro link-tp-id    te-tp-id
                   |     +--ro flags*        path-attribute-flags
                   +--:(unnumbered-link-hop)
                   |  +--ro unnumbered-link-hop
                   |     +--ro link-tp-id    te-tp-id
                   |     +--ro node-id?      te-node-id
                   |     +--ro flags*        path-attribute-flags
                   +--:(label)
                      +--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

  rpcs:
    +---x tunnels-path-compute
    |  +---w input
    |  |  +---w path-compute-info
    |  +--ro output
    |     +--ro path-compute-result
    +---x tunnels-actions
       +---w input
       |  +---w tunnel-info
```

```
       |  |  +---w (filter-type)
       |  |     +--:(all-tunnels)
       |  |     |  +---w all       empty
       |  |     +--:(one-tunnel)
       |  |        +---w tunnel?   tunnel-ref
       |  +---w action-info
       |     +---w action?       identityref
       |     +---w disruptive?   empty
       +--ro output
          +--ro action-result?   identityref
```

Figure 7: TE generic YANG model tree diagram.

5.3. YANG Module

The generic TE 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 [RFC8776]

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

```
<CODE BEGINS> file "ietf-te@2021-02-20.yang"
module ietf-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te";

  /* Replace with IANA when assigned */

  prefix te;

  /* Import TE generic types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC6991: Common YANG Data Types.";
  }
  import ietf-yang-types {
    prefix yang;
```

```
    reference
      "RFC6991: Common YANG Data Types.";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group.";
  contact
    "WG Web:   <http://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>

     Editor:   Tarek Saad
               <mailto:tsaad@juniper.net>

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

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

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

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

     Editor:   Igor Bryskin
               <mailto:i_bryskin@yahoo.com>";
  description
    "YANG data module for TE configuration, state, and RPCs.
     The model fully conforms to the Network Management
     Datastore Architecture (NMDA).

     Copyright (c) 2019 IETF Trust and the persons
     identified as authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject
     to the license terms contained in, the Simplified BSD License
     set forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).
     This version of this YANG module is part of RFC XXXX; see
     the RFC itself for full legal notices.";

  // RFC Ed.: replace XXXX with actual RFC number and remove this
  // note.
  // RFC Ed.: update the date below with the date of RFC publication
```

```
  // and remove this note.

  revision 2021-02-20 {
    description
      "Latest update to TE generic YANG module.";
    reference
      "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
       and Interfaces.";
  }

  identity path-computation-error-reason {
    description
      "Base identity for path computation error reasons.";
  }

  identity path-computation-error-no-topology {
    base path-computation-error-reason;
    description
      "Path computation has failed because there is no topology
       with the provided the topology-identifier.";
  }

  identity path-computation-error-no-dependent-server {
    base path-computation-error-reason;
    description
      "Path computation has failed because one or more dependent
       path computation servers are unavailable.
       The dependent path computation server could be
       a BRPC downstream PCE or a child PCE.";
    reference
      "RFC5441, RFC 8685";
  }

  identity path-computation-error-pce-unavailable {
    base path-computation-error-reason;
    description
      "Path computation has failed because PCE is not available.";
    reference
      "RFC5440";
  }

  identity path-computation-error-no-inclusion-hop {
    base path-computation-error-reason;
    description
      "Path computation has failed because there is no
       node or link provided by one or more inclusion hops.";
    reference
      "RFC8685";
```

```
  }

  identity path-computation-error-destination-unknown-in-domain {
    base path-computation-error-reason;
    description
      "Path computation has failed because the destination node is
       unknown in indicated destination domain.";
    reference
      "RFC8685";
  }

  identity path-computation-error-no-resource {
    base path-computation-error-reason;
    description
      "Path computation has failed because there is no
       available resource in one or more domains.";
    reference
      "RFC8685";
  }

  identity path-computation-error-child-pce-unresponsive {
    base path-computation-error-reason;
    description
      "Path computation has failed because child PCE is not
       responsive.";
    reference
      "RFC8685";
  }

  identity path-computation-error-destination-domain-unknown {
    base path-computation-error-reason;
    description
      "Path computation has failed because the destination domain
       was unknown.";
    reference
      "RFC8685";
  }

  identity path-computation-error-p2mp {
    base path-computation-error-reason;
    description
      "Path computation has failed because of P2MP reachability
       problem.";
    reference
      "RFC8306";
  }

  identity path-computation-error-no-gco-migration {
```

```
    base path-computation-error-reason;
    description
      "Path computation has failed because of no GCO migration
       path found.";
    reference
      "RFC5557";
  }

  identity path-computation-error-no-gco-solution {
    base path-computation-error-reason;
    description
      "Path computation has failed because of no GCO solution
       found.";
    reference
      "RFC5557";
  }

  identity path-computation-error-path-not-found {
    base path-computation-error-reason;
    description
      "Path computation no path found error reason.";
    reference
      "RFC5440";
  }

  identity path-computation-error-pks-expansion {
    base path-computation-error-reason;
    description
      "Path computation has failed because of PKS expansion
       failure.";
    reference
      "RFC5520";
  }

  identity path-computation-error-brpc-chain-unavailable {
    base path-computation-error-reason;
    description
      "Path computation has failed because PCE BRPC chain
       unavailable.";
    reference
      "RFC5441";
  }

  identity path-computation-error-source-unknown {
    base path-computation-error-reason;
    description
      "Path computation has failed because source node is unknown.";
    reference
```

```
      "RFC5440";
  }

  identity path-computation-error-destination-unknown {
    base path-computation-error-reason;
    description
      "Path computation has failed because destination node is
       unknown.";
    reference
      "RFC5440";
  }

  identity path-computation-error-no-server {
    base path-computation-error-reason;
    description
      "Path computation has failed because path computation
       server is unavailable.";
    reference
      "RFC5440";
  }

  identity tunnel-actions-type {
    description
      "TE tunnel actions type.";
  }

  identity tunnel-action-reoptimize {
    base tunnel-actions-type;
    description
      "Reoptimize tunnel action type.";
  }

  identity tunnel-admin-auto {
    base te-types:tunnel-admin-state-type;
    description
      "Tunnel administrative auto state. The administrative status in
       state datastore transitions to 'tunnel-admin-up' when the tunnel
       used by the client layer, and to 'tunnel-admin-down' when it is
       not used by the client layer.";
  }

  identity association-type-diversity {
    base te-types:association-type;
    description
      "Association Type diversity used to associate LSPs whose paths
       are to be diverse from each other.";
    reference
      "ID.ietf-pce-association-diversity.";
```

```
  }

  identity path-metric-loss {
    base te-types:path-metric-type;
    description
      "The path loss (as a packet percentage) metric type that
       encodes a function of the unidirectional loss metrics of all
       links traversed by a P2P path.";
    reference "RFC8233";
  }

  identity protocol-origin-type {
    description
      "Base identity for protocol origin type.";
  }
  identity protocol-origin-api {
    base protocol-origin-type;
    description
      "Protocol origin is via Application Programmable Interface
       (API).";
  }
  identity protocol-origin-pcep {
    base protocol-origin-type;
    description
      "Protocol origin is Path Computation Engine Protocol (PCEP).";
    reference "RFC5440";
  }
  identity protocol-origin-bgp {
    base protocol-origin-type;
    description
      "Protocol origin is Border Gateway Protocol (BGP).";
    reference "RFC5512";
  }

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

  typedef path-ref {
    type union {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/"
           + "te:primary-paths/te:primary-path/te:name";
```

```
      }
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/"
           + "te:secondary-paths/te:secondary-path/te:name";
      }
    }
    description
      "This type is used by data models that need to reference
       configured primary or secondary path of a TE tunnel.";
  }

  typedef te-gen-node-id {
    type union {
      type te-types:te-node-id;
      type inet:ip-address;
    }
    description
      "Generic type that identifies a node in a TE topology.";
  }

  /**
   * TE tunnel generic groupings
   */

  grouping te-generic-node-id {
    description
      "A reusable grouping for a TE generic node identifier.";
    leaf id {
      type te-gen-node-id;
      description
        "The identifier of the node. Can be represented as IP
         address or dotted quad address.";
    }
    leaf type {
      type enumeration {
        enum ip {
          description
            "IP address representation of the node identifier.";
        }
        enum dotted-quad {
          description
            "Dotted quad address representation of the node
             identifier.";
        }
      }
      description
        "Type of node identifier representation.";
    }
```

```
  }

  grouping primary-path {
    description
      "The tunnel primary path properties.";
    uses path-common-properties;
    uses path-preference;
    uses k-requested-paths;
    uses path-compute-info;
    uses path-state;
  }

  grouping primary-reverse-path {
    description
      "The tunnel primary reverse path properties.";
    reference
      "RFC7551";
    uses path-common-properties;
    uses path-compute-info;
    uses path-state;
  }

  grouping secondary-path {
    description
      "The tunnel secondary path properties.";
    uses path-common-properties;
    uses path-preference;
    uses path-compute-info;
    uses protection-restoration-properties;
    uses path-state;
  }

  grouping secondary-reverse-path {
    description
      "The tunnel secondary reverse path properties.";
    uses path-common-properties;
    uses path-preference;
    uses path-compute-info;
    uses protection-restoration-properties;
    uses path-state;
  }

  grouping path-common-properties {
    description
      "Common path attributes.";
    leaf name {
      type string;
      description
```

```
          "TE path name.";
     }
     leaf path-computation-method {
       type identityref {
         base te-types:path-computation-method;
       }
       default "te-types:path-locally-computed";
       description
         "The method used for computing the path, either
          locally computed, queried from a server or not
          computed at all (explicitly configured).";
     }
     container path-computation-server {
       when "derived-from-or-self(../path-computation-method, "
          + "'te-types:path-externally-queried')" {
         description
           "The path-computation server when the path is
            externally queried.";
       }
       uses te-generic-node-id;
       description
         "Address of the external path computation
          server.";
     }
     leaf compute-only {
       type empty;
       description
         "When set, the path is computed and updated whenever
          the topology is updated. No resources are committed
          or reserved in the network.";
     }
     leaf use-path-computation {
       when "derived-from-or-self(../path-computation-method, "
          + "'te-types:path-locally-computed')";
       type boolean;
       default "true";
       description
         "When 'true' indicates the path is dynamically computed and/or
          validated against the Traffic-Engineering Database (TED),
          and when 'false' indicates no validation against the TED is
          required.";
     }
     leaf lockdown {
       type empty;
       description
         "Indicates no reoptimization to be attempted for this path.";
     }
     leaf path-scope {
```

```
      type identityref {
        base te-types:path-scope-type;
      }
      default "te-types:path-scope-end-to-end";
      config false;
      description
        "Path scope if segment or an end-to-end path.";
    }
  }

  /* This grouping will be re-used in path-computation rpc */

  grouping path-compute-info {
    description
      "Attributes used for path computation request.";
    uses tunnel-associations-properties;
    uses te-types:generic-path-optimization;
    leaf named-path-constraint {
      if-feature "te-types:named-path-constraints";
      type leafref {
        path "/te:te/te:globals/te:named-path-constraints/"
           + "te:named-path-constraint/te:name";
      }
      description
        "Reference to a globally defined named path constraint set.";
    }
    uses path-constraints-common;
  }

  /* This grouping will be re-used in path-computation rpc */

  grouping path-preference {
    description
      "The path preference.";
    leaf preference {
      type uint8 {
        range "1..255";
      }
      default "1";
      description
        "Specifies a preference for this path. The lower the number
         higher the preference.";
    }
  }

  /* This grouping will be re-used in path-computation rpc */

  grouping k-requested-paths {
```

```
    description
      "The k-shortest paths requests.";
    leaf k-requested-paths {
      type uint8;
      default "1";
      description
        "The number of k-shortest-paths requested from the path
         computation server and returned sorted by its optimization
         objective. The value 0 all possible paths.";
    }
  }

  grouping path-properties {
    description
      "TE computed path properties grouping.";
    uses te-types:generic-path-properties {
      augment "path-properties" {
        description
          "additional path properties returned by path computation.";
        uses te-types:te-bandwidth;
        leaf disjointness-type {
          type te-types:te-path-disjointness;
          config false;
          description
            "The type of resource disjointness.
             When reported for a primary path, it represents the
             minimum level of disjointness of all the secondary
             paths.
             When reported for a secondary path, it represents the
             disjointness of the secondary path.";
        }
      }
    }
  }

  grouping path-state {
    description
      "TE per path state parameters.";
    uses path-computation-response;
    uses lsp-provisioning-error-info {
      augment "lsp-provisioning-error-infos/"
            + "lsp-provisioning-error-info" {
        description
          "Augmentation of LSP provisioning information under a
           specific path.";
        leaf lsp-id {
          type uint16;
          description
```

```
            "The LSP-ID for which path computation was performed.";
        }
      }
    }
    container lsps {
      config false;
      description
        "The TE LSPs container.";
      list lsp {
        key "node lsp-id";
        description
          "List of LSPs associated with the tunnel.";
        leaf tunnel-name {
          type leafref {
            path "/te:te/te:lsps/te:lsp/te:tunnel-name";
          }
          description "TE tunnel name.";
        }
        leaf node {
          type leafref {
            path "/te:te/te:lsps/te:lsp/te:node";
          }
          description "The node where the LSP state resides on.";
        }
        leaf lsp-id {
          type leafref {
            path "/te:te/te:lsps/te:lsp/te:lsp-id";
          }
          description "The TE LSP identifier.";
        }
      }
    }
  }

  /* This grouping will be re-used in path-computation rpc */

  grouping path-computation-response {
    description
      "Attributes reported by path computation response.";
    container computed-paths-properties {
      config false;
      description
        "Computed path properties container.";
      list computed-path-properties {
        key "k-index";
        description
          "List of computed paths.";
        leaf k-index {
```

```
          type uint8;
          description
            "The k-th path returned from the computation server.
             A lower k value path is more optimal than higher k
             value path(s)";
        }
        uses path-properties {
          description
            "The TE path computed properties.";
        }
      }
    }
    container computed-path-error-infos {
      config false;
      description
        "Path computation information container.";
      list computed-path-error-info {
        description
          "List of path computation info entries.";
        leaf error-description {
          type string;
          description
            "Textual representation of the error occurred during
             path computation.";
        }
        leaf error-timestamp {
          type yang:date-and-time;
          description
            "Timestamp of last path computation attempt.";
        }
        leaf error-reason {
          type identityref {
            base path-computation-error-reason;
          }
          description
            "Reason for the path computation error.";
        }
      }
    }
  }

  grouping lsp-provisioning-error-info {
    description
      "Grouping for LSP provisioning error information.";
    container lsp-provisioning-error-infos {
      config false;
      description
        "LSP provisioning error information.";
```

```
      list lsp-provisioning-error-info {
        description
          "List of LSP provisioning error info entries.";
        leaf error-description {
          type string;
          description
            "Textual representation of the error occurred during
             path computation.";
        }
        leaf error-timestamp {
          type yang:date-and-time;
          description
            "Timestamp of when the reported error occurred.";
        }
        leaf error-node-id {
          type te-types:te-node-id;
          default "0.0.0.0";
          description
            "Node identifier of node where error occurred.";
        }
        leaf error-link-id {
          type te-types:te-tp-id;
          default "0";
          description
            "Link ID where the error occurred.";
        }
      }
    }
  }

  grouping protection-restoration-properties-state {
    description
      "Protection parameters grouping.";
    leaf lockout-of-normal {
      type boolean;
      default "false";
      description
        "When set to 'True', it represents a lockout of normal
         traffic external command. When set to 'False', it
         represents a clear lockout of normal traffic external
         command. The lockout of normal traffic command applies
         to this Tunnel.";
      reference
        "RFC4427";
    }
    leaf freeze {
      type boolean;
      default "false";
```

```
      description
        "When set to 'True', it represents a freeze external command.
         When set to 'False', it represents a clear freeze external
         command. The freeze command applies to all the Tunnels which
         are sharing the protection resources with this Tunnel.";
      reference
        "RFC4427";
    }
    leaf lsp-protection-role {
      type enumeration {
        enum working {
          description
            "A working LSP must be a primary LSP whilst a protecting
             LSP can be either a primary or a secondary LSP. Also,
             known as protected LSPs when working LSPs are associated
             with protecting LSPs.";
        }
        enum protecting {
          description
            "A secondary LSP is an LSP that has been provisioned
             in the control plane only; e.g. resource allocation
             has not been committed at the data plane.";
        }
      }
      default "working";
      description
        "LSP role type.";
      reference
        "RFC4872, section 4.2.1";
    }
    leaf lsp-protection-state {
      type identityref {
        base te-types:lsp-protection-state;
      }
      default "te-types:normal";
      description
        "The state of the APS state machine controlling which
         tunnels is using the resources of the protecting LSP.";
    }
    leaf protection-group-ingress-node-id {
      type te-types:te-node-id;
      default "0.0.0.0";
      description
        "Indicates the te-node-id of the protection group
         ingress node when the APS state represents an external
         command (LoP, SF, MS) applied to it or a WTR timer
         running on it. If the external command is not applied to
         the ingress node or the WTR timer is not running on it,
```

```
         this attribute is not specified. A value 0.0.0.0 is used
         when the te-node-id of the protection group ingress node is
         unknown (e.g., because the ingress node is outside the scope
         of control of the server)";
    }
    leaf protection-group-egress-node-id {
      type te-types:te-node-id;
      default "0.0.0.0";
      description
        "Indicates the te-node-id of the protection group egress node
         when the APS state represents an external command (LoP, SF,
         MS) applied to it or a WTR timer running on it. If the
         external command is not applied to the ingress node or
         the WTR timer is not running on it, this attribute is not
         specified. A value 0.0.0.0 is used when the te-node-id of
         the protection group ingress node is unknown (e.g., because
         the ingress node is outside the scope of control of the
         server)";
    }
  }

  grouping protection-restoration-properties {
    description
      "Protection and restoration parameters.";
    container protection {
      description
        "Protection parameters.";
      leaf enable {
        type boolean;
        default "false";
        description
          "A flag to specify if LSP protection is enabled.";
        reference
          "RFC4427";
      }
      leaf protection-type {
        type identityref {
          base te-types:lsp-protection-type;
        }
        default "te-types:lsp-protection-unprotected";
        description
          "LSP protection type.";
      }
      leaf protection-reversion-disable {
        type boolean;
        default "false";
        description
          "Disable protection reversion to working path.";
```

```
      }
      leaf hold-off-time {
        type uint32;
        units "milli-seconds";
        default "0";
        description
          "The time between the declaration of an SF or SD condition
           and the initialization of the protection switching
           algorithm.";
        reference
          "RFC4427";
      }
      leaf wait-to-revert {
        type uint16;
        units "seconds";
        description
          "Time to wait before attempting LSP reversion.";
        reference
          "RFC4427";
      }
      leaf aps-signal-id {
        type uint8 {
          range "1..255";
        }
        default "1";
        description
          "The APS signal number used to reference the traffic of this
           tunnel. The default value for normal traffic is 1.
           The default value for extra-traffic is 255. If not specified,
           non-default values can be assigned by the server,
           if and only if, the server controls both endpoints.";
        reference
          "RFC4427";
      }
    }
    container restoration {
      description
        "Restoration parameters.";
      leaf enable {
        type boolean;
        default "false";
        description
          "A flag to specify if LSP restoration is enabled.";
        reference
          "RFC4427";
      }
      leaf restoration-type {
        type identityref {
```

```
          base te-types:lsp-restoration-type;
        }
        default "te-types:lsp-restoration-restore-any";
        description
          "LSP restoration type.";
      }
      leaf restoration-scheme {
        type identityref {
          base te-types:restoration-scheme-type;
        }
        default "te-types:restoration-scheme-preconfigured";
        description
          "LSP restoration scheme.";
      }
      leaf restoration-reversion-disable {
        type boolean;
        default "false";
        description
          "Disable restoration reversion to working path.";
      }
      leaf hold-off-time {
        type uint32;
        units "milli-seconds";
        description
          "The time between the declaration of an SF or SD condition
           and the initialization of the protection switching
           algorithm.";
        reference
          "RFC4427";
      }
      leaf wait-to-restore {
        type uint16;
        units "seconds";
        description
          "Time to wait before attempting LSP restoration.";
        reference
          "RFC4427";
      }
      leaf wait-to-revert {
        type uint16;
        units "seconds";
        description
          "Time to wait before attempting LSP reversion.";
        reference
          "RFC4427";
      }
    }
  }
```

```
  grouping tunnel-associations-properties {
    description
      "TE tunnel association grouping.";
    container association-objects {
      description
        "TE tunnel associations.";
      list association-object {
        key "association-key";
        unique "type id source/id source/type";
        description
          "List of association base objects.";
        reference
          "RFC4872";
        leaf association-key {
          type string;
          description
            "Association key used to identify a specific
             association in the list";
        }
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description
            "Association type.";
          reference
            "RFC4872";
        }
        leaf id {
          type uint16;
          description
            "Association identifier.";
          reference
            "RFC4872";
        }
        container source {
          uses te-generic-node-id;
          description
            "Association source.";
          reference
            "RFC4872";
        }
      }
      list association-object-extended {
        key "association-key";
        unique
          "type id source/id source/type global-source extended-id";
        description
```

```
          "List of extended association objects.";
        reference
          "RFC6780";
        leaf association-key {
          type string;
          description
            "Association key used to identify a specific
             association in the list";
        }
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description
            "Association type.";
          reference
            "RFC4872, RFC6780";
        }
        leaf id {
          type uint16;
          description
            "Association identifier.";
          reference
            "RFC4872, RFC6780";
        }
        container source {
          uses te-generic-node-id;
          description
            "Association source.";
          reference
            "RFC4872, RFC6780";
        }
        leaf global-source {
          type uint32;
          description
            "Association global source.";
          reference
            "RFC6780";
        }
        leaf extended-id {
          type yang:hex-string;
          description
            "Association extended identifier.";
          reference
            "RFC6780";
        }
      }
    }
```

```
  }

  /* TE tunnel configuration/state grouping */
  /* This grouping will be re-used in path-computation rpc */

  grouping tunnel-hierarchy-properties {
    description
      "A grouping for TE tunnel hierarchy information.";
    container hierarchy {
      description
        "Container for TE hierarchy related information.";
      container dependency-tunnels {
        description
          "List of tunnels that this tunnel can be potentially
           dependent on.";
        list dependency-tunnel {
          key "name";
          description
            "A tunnel entry that this tunnel can potentially depend
             on.";
          leaf name {
            type leafref {
              path "/te:te/te:tunnels/te:tunnel/te:name";
              require-instance false;
            }
            description
              "Dependency tunnel name. The tunnel may not have been
               instantiated yet.";
          }
          leaf encoding {
            type identityref {
              base te-types:lsp-encoding-types;
            }
            default "te-types:lsp-encoding-packet";
            description
              "The LSP encoding type for the dependency tunnel.";
            reference
              "RFC3945";
          }
          leaf switching-type {
            type identityref {
              base te-types:switching-capabilities;
            }
            default "te-types:switching-psc1";
            description
              "The LSP switching type for the dependency tunnel.";
            reference
              "RFC3945";
```

```
          }
        }
      }
      container hierarchical-link {
        description
          "Identifies a hierarchical link (in client layer)
           that this tunnel is associated with.";
        reference
          "RFC4206";
        leaf local-te-node-id {
          type te-types:te-node-id;
          default "0.0.0.0";
          description
            "The local TE node identifier.";
        }
        leaf local-te-link-tp-id {
          type te-types:te-tp-id;
          default "0";
          description
            "The local TE link termination point identifier.";
        }
        leaf remote-te-node-id {
          type te-types:te-node-id;
          default "0.0.0.0";
          description
            "Remote TE node identifier.";
        }
        uses te-types:te-topology-identifier;
      }
    }
  }

  grouping tunnel-properties {
    description
      "Top level grouping for tunnel properties.";
    leaf operational-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      config false;
      description
        "TE tunnel operational state.";
    }
    leaf name {
      type string;
      description
        "TE tunnel name.";
    }
```

```
    leaf identifier {
      type uint32;
      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 admin-state {
      type identityref {
        base te-types:tunnel-admin-state-type;
      }
      default "te-types:tunnel-admin-state-up";
      description
        "TE tunnel administrative state.";
    }
    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 identifier.";
    }
    container controller {
      description
        "Contains tunnel data relevant to external controller(s).
         This target node may be augmented by external module(s),
         for example, to add data for PCEP initiated and/or
         delegated tunnels.";
      leaf protocol-origin {
        type identityref {
          base protocol-origin-type;
        }
        description
          "The protocol origin for instantiating the tunnel.";
      }
      leaf controller-entity-id {
        type string;
        description
          "An identifier unique within the scope of visibility that
           associated with the entity that controls the tunnel";
        reference "RFC8232";
      }
    }
    leaf src-tunnel-tp-id {
      type binary;
      description
        "TE tunnel source termination point identifier.";
    }
    leaf dst-tunnel-tp-id {
      type binary;
      description
        "TE tunnel destination termination point identifier.";
    }
    leaf bidirectional {
      type boolean;
      default "false";
      description
        "Indicates a bidirectional co-routed LSP.";
    }
    uses tunnel-associations-properties;
    uses protection-restoration-properties;
```

```
    uses te-types:tunnel-constraints;
    uses tunnel-hierarchy-properties;
    container primary-paths {
      description
        "The set of primary paths.";
      list primary-path {
        key "name";
        description
          "List of primary paths for this tunnel.";
        uses primary-path;
        container primary-reverse-path {
          description
            "The reverse primary path properties.";
          uses primary-reverse-path;
          container candidate-secondary-reverse-paths {
            description
              "The set of referenced candidate reverse secondary paths
               from the full set of secondary reverse paths which may be
               used for this primary path.";
            list candidate-secondary-reverse-path {
              key "secondary-path";
              ordered-by user;
              description
                "List of candidate secondary reverse path(s)";
              leaf secondary-path {
                type leafref {
                  path "../../../../../../te:secondary-reverse-paths/"
                     + "te:secondary-reverse-path/te:name";
                }
                description
                  "A reference to the secondary reverse path that
                   should be utilised when the containing primary
                   reverse path option is in use.";
              }
            }
          }
        }
        container candidate-secondary-paths {
          description
            "The set of candidate secondary paths which may be used
             for this primary path. When secondary paths are specified
             in the list the path of the secondary LSP in use must be
             restricted to those path options referenced. The
             priority of the secondary paths is specified within the
             list. Higher priority values are less preferred - that is
             to say that a path with priority 0 is the most preferred
             path. In the case that the list is empty, any secondary
             path option may be utilised when the current primary path
```

```
             is in use.";
          list candidate-secondary-path {
            key "secondary-path";
            ordered-by user;
            description
              "List of candidate secondary paths for this tunnel.";
            leaf secondary-path {
              type leafref {
                path "../../../../../te:secondary-paths/"
                   + "te:secondary-path/te:name";
              }
              description
                "A reference to the secondary path that should be
                 utilised when the containing primary path option is
                 in use.";
            }
            leaf active {
              type boolean;
              config false;
              description
                "Indicates the current active path option that has
                 been selected of the candidate secondary paths.";
            }
          }
        }
      }
    }
    container secondary-paths {
      description
        "The set of secondary paths.";
      list secondary-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses secondary-path;
      }
    }
    container secondary-reverse-paths {
      description
        "The set of secondary reverse paths.";
      list secondary-reverse-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses secondary-reverse-path;
      }
    }
  }
```

```
  grouping tunnel-actions {
    description
      "Tunnel actions.";
    action tunnel-action {
      description
        "Tunnel action.";
      input {
        leaf action-type {
          type identityref {
            base tunnel-actions-type;
          }
          description
            "Tunnel action type.";
        }
      }
      output {
        leaf action-result {
          type identityref {
            base te-types:te-action-result;
          }
          description
            "The result of the tunnel action operation.";
        }
      }
    }
  }

  grouping tunnel-protection-actions {
    description
      "Protection external command actions.";
    action protection-external-commands {
      input {
        leaf protection-external-command {
          type identityref {
            base te-types:protection-external-commands;
          }
          description
            "Protection external command.";
        }
        leaf protection-group-ingress-node-id {
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on ingress node.
             By default, if neither ingress nor egress node-id
             is set, the action applies to ingress node only.";
        }
        leaf protection-group-egress-node-id {
```

```
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on egress node.
             By default, if neither ingress nor egress node-id
             is set, the action applies to ingress node only.";
        }
        leaf path-ref {
          type path-ref;
          description
            "Indicates to which path the external command applies to.";
        }
        leaf traffic-type {
          type enumeration {
            enum normal-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the normal traffic (this Tunnel).";
            }
            enum null-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the null traffic.";
            }
            enum extra-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the extra traffic (the extra-traffic Tunnel sharing
                 protection bandwidth with this Tunnel).";
            }
          }
          description
            "Indicates whether the manual-switch or forced-switch
             commands applies to the normal traffic, the null traffic
             or the extra-traffic.";
          reference
            "RFC4427";
        }
        leaf extra-traffic-tunnel-ref {
          type tunnel-ref;
          description
            "In case there are multiple extra-traffic tunnels sharing
             protection bandwidth with this Tunnel (m:n protection),
             represents which extra-traffic Tunnel the manual-switch or
             forced-switch to extra-traffic command applies to.";
        }
      }
    }
```

```
  }

  /*** End of TE tunnel groupings ***/
  /**
   * LSP related generic groupings
   */

  grouping lsp-record-route-information-state {
    description
      "LSP Recorded route information grouping.";
    container lsp-record-route-information {
      description
        "RSVP recorded route object information.";
      list lsp-record-route-information {
        when "../../origin-type = 'ingress'" {
          description
            "Applicable on ingress LSPs only.";
        }
        key "index";
        description
          "Record route list entry.";
        uses te-types:record-route-state;
      }
    }
  }

  grouping lsps-grouping {
    description
      "LSPs state operational data grouping.";
    container lsps {
      config false;
      description
        "TE LSPs state container.";
      list lsp {
        key "tunnel-name lsp-id node";
        unique "source destination tunnel-id lsp-id "
          + "extended-tunnel-id";
        description
          "List of LSPs associated with the tunnel.";
        leaf tunnel-name {
          type string;
          description "The TE tunnel name.";
        }
        leaf lsp-id {
          type uint16;
          description
            "Identifier used in the SENDER_TEMPLATE and the FILTER_SPEC
             that can be changed to allow a sender to share resources
```

```
             with itself.";
          reference
            "RFC3209";
        }
        leaf node {
          type te-types:te-node-id;
          description
            "The node where the TE LSP state resides on.";
        }
        uses lsp-properties-state;
        uses lsp-record-route-information-state;
      }
    }
  }

  /*** End of TE LSP groupings ***/
  /**
   * TE global generic groupings
   */
  /* Global named admin-groups configuration data */

  grouping named-admin-groups-properties {
    description
      "Global named administrative groups configuration
       grouping.";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
         interface named admin-group.";
    }
    leaf bit-position {
      type uint32;
      description
        "Bit position representing the administrative group.";
      reference
        "RFC3209 and RFC7308";
    }
  }

  grouping named-admin-groups {
    description
      "Global named administrative groups configuration
       grouping.";
    container named-admin-groups {
      description
        "TE named admin groups container.";
      list named-admin-group {
```

```
        if-feature "te-types:extended-admin-groups";
        if-feature "te-types:named-extended-admin-groups";
        key "name";
        description
          "List of named TE admin-groups.";
        uses named-admin-groups-properties;
      }
    }
  }

  /* Global named admin-srlgs configuration data */

  grouping named-srlgs {
    description
      "Global named SRLGs configuration grouping.";
    container named-srlgs {
      description
        "TE named SRLGs container.";
      list named-srlg {
        if-feature "te-types:named-srlg-groups";
        key "name";
        description
          "A list of named SRLG groups.";
        leaf name {
          type string;
          description
            "A string name that uniquely identifies a TE
             interface named SRLG.";
        }
        leaf value {
          type te-types:srlg;
          description
            "An SRLG value.";
        }
        leaf cost {
          type uint32;
          description
            "SRLG associated cost. Used during path to append
             the path cost when traversing a link with this SRLG.";
        }
      }
    }
  }

  /* Global named paths constraints configuration data */

  grouping path-constraints-common {
    description
```

```
      "Global named path constraints configuration
       grouping.";
    uses te-types:common-path-constraints-attributes;
    uses te-types:generic-path-disjointness;
    uses te-types:path-constraints-route-objects;
    container path-in-segment {
      presence "The end-to-end tunnel starts in a previous domain;
                this tunnel is a segment in the current domain.";
      description
        "If an end-to-end tunnel crosses multiple domains using
         the same technology, some additional constraints have to be
         taken in consideration in each domain.
         This 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
        "If an end-to-end tunnel crosses multiple domains using
         the same technology, some additional constraints have to be
         taken in consideration in each domain.
         This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
  }

  grouping named-path-constraints {
    description
      "Global named path constraints configuration
       grouping.";
    container named-path-constraints {
      description
        "TE named path constraints container.";
      list named-path-constraint {
        if-feature "te-types:named-path-constraints";
        key "name";
        leaf name {
          type string;
          description
            "A string name that uniquely identifies a
             path constraint set.";
        }
        uses path-constraints-common;
        description
```

```
          "A list of named path constraints.";
      }
    }
  }

  /* TE globals container data */

  grouping globals-grouping {
    description
      "Globals TE system-wide configuration data grouping.";
    container globals {
      description
        "Globals TE system-wide configuration data container.";
      uses named-admin-groups;
      uses named-srlgs;
      uses named-path-constraints;
    }
  }

  /* TE tunnels container data */

  grouping tunnels-grouping {
    description
      "Tunnels TE configuration data grouping.";
    container tunnels {
      description
        "Tunnels TE configuration data container.";
      list tunnel {
        key "name";
        description
          "The list of TE tunnels.";
        uses tunnel-properties;
        uses tunnel-actions;
        uses tunnel-protection-actions;
      }
    }
  }

  /* TE LSPs ephemeral state container data */

  grouping lsp-properties-state {
    description
      "LSPs state operational data grouping.";
    leaf source {
      type te-types:te-node-id;
      description
        "Tunnel sender address extracted from
         SENDER_TEMPLATE  object.";
```

```
      reference
        "RFC3209";
    }
    leaf destination {
      type te-types:te-node-id;
      description
        "The tunnel endpoint address extracted from SESSION object.";
      reference
        "RFC3209";
    }
    leaf tunnel-id {
      type uint16;
      description
        "The tunnel identifier used in the SESSION that remains constant
         over the life of the tunnel.";
      reference
        "RFC3209";
    }
    leaf extended-tunnel-id {
      type yang:dotted-quad;
      description
        "The LSP Extended Tunnel ID.";
      reference
        "RFC3209";
    }
    leaf operational-state {
      type identityref {
        base te-types:lsp-state-type;
      }
      description
        "The LSP operational state.";
    }
    leaf signaling-type {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "The signaling protocol used to set up this LSP.";
    }
    leaf origin-type {
      type enumeration {
        enum ingress {
          description
            "Origin ingress.";
        }
        enum egress {
          description
            "Origin egress.";
```

```
        }
        enum transit {
          description
            "Origin transit.";
        }
      }
      default "ingress";
      description
        "The origin of the LSP relative to the location of the local
         switch in the path.";
    }
    leaf lsp-resource-status {
      type enumeration {
        enum primary {
          description
            "A primary LSP is a fully established LSP for which the
             resource allocation has been committed at the data
             plane.";
        }
        enum secondary {
          description
            "A secondary LSP is an LSP that has been provisioned
             in the control plane only; e.g. resource allocation
             has not been committed at the data plane.";
        }
      }
      default "primary";
      description
        "LSP resource allocation state.";
      reference
        "RFC4872, section 4.2.1";
    }
    uses protection-restoration-properties-state;
  }

  /*** End of TE global groupings ***/
  /**
   * TE container
   */

  container te {
    presence "Enable TE feature.";
    description
      "TE global container.";
    /* TE Global Data */
    uses globals-grouping;

    /* TE Tunnel Data */
```

```
    uses tunnels-grouping;

    /* TE LSPs Data */
    uses lsps-grouping;
  }

  /* TE Tunnel RPCs/execution Data */

  rpc tunnels-path-compute {
    description
      "TE tunnels RPC nodes.";
    input {
      container path-compute-info {
        /* An external path compute module may augment this target. */
        description
          "RPC input information.";
      }
    }
    output {
      container path-compute-result {
        /* An external path compute module may augment this target. */
        description
          "RPC output information.";
      }
    }
  }

  rpc tunnels-actions {
    description
      "TE tunnels actions RPC";
    input {
      container tunnel-info {
        description
          "TE tunnel information.";
        choice filter-type {
          mandatory true;
          description
            "Filter choice.";
          case all-tunnels {
            leaf all {
              type empty;
              mandatory true;
              description
                "Apply action on all TE tunnels.";
            }
          }
          case one-tunnel {
            leaf tunnel {
```

```
              type tunnel-ref;
              description
                "Apply action on the specific TE tunnel.";
            }
          }
        }
      }
      container action-info {
        description
          "TE tunnel action information.";
        leaf action {
          type identityref {
            base tunnel-actions-type;
          }
          description
            "The action type.";
        }
        leaf disruptive {
          when "derived-from-or-self(../action, "
             + "'te:tunnel-action-reoptimize')";
          type empty;
          description
            "Specifies whether or not the reoptimization action
             is allowed to be disruptive.";
        }
      }
    }
    output {
      leaf action-result {
        type identityref {
          base te-types:te-action-result;
        }
        description
          "The result of the tunnel action operation.";
      }
    }
  }
}
<CODE ENDS>
```

Figure 8: generic TE YANG module

6. TE Device YANG Model

The device TE YANG module ('ietf-te-device') models data that is specific to managing a TE device. This module augments the generic TE YANG module.

6.1. Module Structure

6.1.1. TE Interfaces

This branch of the model manages TE interfaces that are present on a device. Examples of TE interface properties are:

- Maximum reservable bandwidth, bandwidth constraints (BC)
- Flooding parameters
  * Flooding intervals and threshold values
- interface attributes
  * (Extended) administrative groups
  * SRLG values
  * TE metric value
- Fast reroute backup tunnel properties (such as static, auto-tunnel)

The derived state associated with interfaces is grouped under the interface "state" sub-container as shown in Figure 9. This covers state data such as:

- Bandwidth information: maximum bandwidth, available bandwidth at different priorities and for each class-type (CT)
- List of admitted LSPs
  * Name, bandwidth value and pool, time, priority
- Statistics: state counters, flooding counters, admission counters (accepted/rejected), preemption counters
- Adjacency information
  * Neighbor address
  * Metric value

```
   module: ietf-te-device
     augment /te:te:
         +--rw interfaces
            .
            +-- rw te-dev:te-attributes
                   <<intended configuration>>
                .
                +-- ro state
                   <<derived state associated with the TE interface>>
```

Figure 9: TE interface state

6.2. Tree Diagram

Figure 10 shows the tree diagram of the device TE YANG model defined in modules 'ietf-te.yang'.

```
 module: ietf-te-device
   augment /te:te:
     +--rw interfaces
     |  +--rw threshold-type?            enumeration
     |  +--rw delta-percentage?          rt-types:percentage
     |  +--rw threshold-specification?   enumeration
     |  +--rw up-thresholds*             rt-types:percentage
     |  +--rw down-thresholds*           rt-types:percentage
     |  +--rw up-down-thresholds*        rt-types:percentage
     |  +--rw interface* [interface]
     |     +--rw interface                            if:interface-ref
     |     +--rw te-metric?
     |     |       te-types:te-metric
     |     +--rw (admin-group-type)?
     |     |  +--:(value-admin-groups)
     |     |  |  +--rw (value-admin-group-type)?
     |     |  |     +--:(admin-groups)
     |     |  |     |  +--rw admin-group?
     |     |  |     |          te-types:admin-group
     |     |  |     +--:(extended-admin-groups)
     |     |  |              {te-types:extended-admin-groups}?
     |     |  |        +--rw extended-admin-group?
     |     |  |                te-types:extended-admin-group
     |     |  +--:(named-admin-groups)
     |     |     +--rw named-admin-groups* [named-admin-group]
     |     |             {te-types:extended-admin-groups,te-types:named-
     extended-admin-groups}?
     |     |        +--rw named-admin-group    leafref
     |     +--rw (srlg-type)?
     |     |  +--:(value-srlgs)
     |     |  |  +--rw values* [value]
```

```
     |        |  |      +--rw value    uint32
     |        |  +--:(named-srlgs)
     |        |     +--rw named-srlgs* [named-srlg]
     |        |             {te-types:named-srlg-groups}?
     |        |        +--rw named-srlg    leafref
     |        +--rw threshold-type?                     enumeration
     |        +--rw delta-percentage?
     |        |       rt-types:percentage
     |        +--rw threshold-specification?            enumeration
     |        +--rw up-thresholds*
     |        |       rt-types:percentage
     |        +--rw down-thresholds*
     |        |       rt-types:percentage
     |        +--rw up-down-thresholds*
     |        |       rt-types:percentage
     |        +--rw switching-capabilities* [switching-capability]
     |        |  +--rw switching-capability    identityref
     |        |  +--rw encoding?               identityref
     |        +--ro state
     |           +--ro te-advertisements-state
     |              +--ro flood-interval?           uint32
     |              +--ro last-flooded-time?        uint32
     |              +--ro next-flooded-time?        uint32
     |              +--ro last-flooded-trigger?     enumeration
     |              +--ro advertised-level-areas* [level-area]
     |                 +--ro level-area    uint32
     +--rw performance-thresholds
   augment /te:te/te:globals:
     +--rw lsp-install-interval?        uint32
     +--rw lsp-cleanup-interval?        uint32
     +--rw lsp-invalidation-interval?   uint32
   augment /te:te/te:tunnels/te:tunnel:
     +--rw path-invalidation-action?    identityref
     +--rw lsp-install-interval?        uint32
     +--rw lsp-cleanup-interval?        uint32
     +--rw lsp-invalidation-interval?   uint32
   augment /te:te/te:lsps/te:lsp:
     +--ro lsp-timers
     |  +--ro life-time?         uint32
     |  +--ro time-to-install?   uint32
     |  +--ro time-to-destroy?   uint32
     +--ro downstream-info
     |  +--ro nhop?                 te-types:te-tp-id
     |  +--ro outgoing-interface?   if:interface-ref
     |  +--ro neighbor
     |  |  +--ro id?     te-gen-node-id
     |  |  +--ro type?   enumeration
     |  +--ro label?                rt-types:generalized-label
```

```
     +--ro upstream-info
        +--ro phop?       te-types:te-tp-id
        +--ro neighbor
        |  +--ro id?     te-gen-node-id
        |  +--ro type?   enumeration
        +--ro label?      rt-types:generalized-label

  rpcs:
    +---x link-state-update
       +---w input
          +---w (filter-type)
             +--:(match-all)
             |  +---w all          empty
             +--:(match-one-interface)
                +---w interface?   if:interface-ref
```

Figure 10: TE generic YANG model tree diagram.

6.3. YANG Module

The device TE YANG module 'ietf-te-device' imports the following module(s):

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

o ietf-interfaces defined in [RFC8343]

o ietf-routing-types defined in [RFC8294]

o ietf-te-types defined in [RFC8776]

o ietf-te defined in this document

```
<CODE BEGINS> file "ietf-te-device@2021-02-20.yang"


module ietf-te-device {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

  /* Replace with IANA when assigned */

  prefix te-dev;

  /* Import TE module */

  import ietf-te {
    prefix te;
```

```
    reference
      "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }

  /* Import TE types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC8343: A YANG Data Model for Interface Management";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC8294: Common YANG Data Types for the Routing Area";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>

     Editor:   Tarek Saad
               <mailto:tsaad@juniper.net>

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

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

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

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

     Editor:   Igor Bryskin
               <mailto:i_bryskin@yahoo.com>";
  description
```

```
     "YANG data module for TE device configurations,
      state, and RPCs. The model fully conforms to the
      Network Management Datastore Architecture (NMDA).

      Copyright (c) 2019 IETF Trust and the persons
      identified as authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject
      to the license terms contained in, the Simplified BSD License
      set forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).
      This version of this YANG module is part of RFC XXXX; see
      the RFC itself for full legal notices.";

   // RFC Ed.: replace XXXX with actual RFC number and remove this
   // note.
   // RFC Ed.: update the date below with the date of RFC publication
   // and remove this note.

   revision 2021-02-20 {
     description
       "Latest update to TE device YANG module.";
     reference
       "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
        and Interfaces";
   }

   /**
    * TE LSP device state grouping
    */

   grouping lsps-device-info {
     description
       "TE LSP device state grouping.";
     container lsp-timers {
       when "../te:origin-type = 'ingress'" {
         description
           "Applicable to ingress LSPs only.";
       }
       description
         "Ingress LSP timers.";
       leaf life-time {
         type uint32;
         units "seconds";
         description
           "TE LSP lifetime.";
```

```
        }
        leaf time-to-install {
          type uint32;
          units "seconds";
          description
            "TE LSP installation delay time.";
        }
        leaf time-to-destroy {
          type uint32;
          units "seconds";
          description
            "TE LSP expiration delay time.";
        }
      }
      container downstream-info {
        when "../te:origin-type != 'egress'" {
          description
            "Downstream information of the LSP.";
        }
        description
          "downstream information.";
        leaf nhop {
          type te-types:te-tp-id;
          description
            "downstream next-hop address.";
        }
        leaf outgoing-interface {
          type if:interface-ref;
          description
            "downstream interface.";
        }
        container neighbor {
          uses te:te-generic-node-id;
          description
            "downstream neighbor address.";
        }
        leaf label {
          type rt-types:generalized-label;
          description
            "downstream label.";
        }
      }
      container upstream-info {
        when "../te:origin-type != 'ingress'" {
          description
            "Upstream information of the LSP.";
        }
        description
```

```
         "upstream information.";
       leaf phop {
         type te-types:te-tp-id;
         description
           "upstream next-hop or previous-hop address.";
       }
       container neighbor {
         uses te:te-generic-node-id;
         description
           "upstream neighbor address.";
       }
       leaf label {
         type rt-types:generalized-label;
         description
           "upstream label.";
       }
     }
   }

   /**
    * Device general groupings.
    */

   grouping lsp-device-timers {
     description
       "Device TE LSP timers configs.";
     leaf lsp-install-interval {
       type uint32;
       units "seconds";
       description
         "TE LSP installation delay time.";
     }
     leaf lsp-cleanup-interval {
       type uint32;
       units "seconds";
       description
         "TE LSP cleanup delay time.";
     }
     leaf lsp-invalidation-interval {
       type uint32;
       units "seconds";
       description
         "TE LSP path invalidation before taking action delay time.";
     }
   }

   /**
    * TE global device groupings
```

```
    */
   /* TE interface container data */

   grouping interfaces-grouping {
     description
       "TE interface configuration data grouping.";
     container interfaces {
       description
         "Configuration data model for TE interfaces.";
       uses te-all-attributes;
       list interface {
         key "interface";
         description
           "TE interfaces.";
         leaf interface {
           type if:interface-ref;
           description
             "TE interface name.";
         }
         /* TE interface parameters */
         uses te-attributes;
       }
     }
   }

   /**
    * TE interface device groupings
    */

   grouping te-admin-groups-config {
     description
       "TE interface affinities grouping.";
     choice admin-group-type {
       description
         "TE interface administrative groups
          representation type.";
       case value-admin-groups {
         choice value-admin-group-type {
           description
             "choice of admin-groups.";
           case admin-groups {
             description
               "Administrative group/Resource
                class/Color.";
             leaf admin-group {
               type te-types:admin-group;
               description
                 "TE interface administrative group.";
```

```
            }
          }
          case extended-admin-groups {
            if-feature "te-types:extended-admin-groups";
            description
              "Extended administrative group/Resource
               class/Color.";
            leaf extended-admin-group {
              type te-types:extended-admin-group;
              description
                "TE interface extended administrative group.";
            }
          }
        }
      }
      case named-admin-groups {
        list named-admin-groups {
          if-feature "te-types:extended-admin-groups";
          if-feature "te-types:named-extended-admin-groups";
          key "named-admin-group";
          description
            "A list of named admin-group entries.";
          leaf named-admin-group {
            type leafref {
              path "../../../../te:globals/"
                 + "te:named-admin-groups/te:named-admin-group/"
                 + "te:name";
            }
            description
              "A named admin-group entry.";
          }
        }
      }
    }
  }

  /* TE interface SRLGs */

  grouping te-srlgs-config {
    description
      "TE interface SRLG grouping.";
    choice srlg-type {
      description
        "Choice of SRLG configuration.";
      case value-srlgs {
        list values {
          key "value";
          description
```

```
             "List of SRLG values that
              this link is part of.";
           leaf value {
             type uint32 {
               range "0..4294967295";
             }
             description
               "Value of the SRLG";
           }
         }
       }
       case named-srlgs {
         list named-srlgs {
           if-feature "te-types:named-srlg-groups";
           key "named-srlg";
           description
             "A list of named SRLG entries.";
           leaf named-srlg {
             type leafref {
               path "../../../../te:globals/"
                  + "te:named-srlgs/te:named-srlg/te:name";
             }
             description
               "A named SRLG entry.";
           }
         }
       }
     }
   }

   grouping te-igp-flooding-bandwidth-config {
     description
       "Configurable items for igp flooding bandwidth
        threshold configuration.";
     leaf threshold-type {
       type enumeration {
         enum delta {
           description
             "'delta' indicates that the local
              system should flood IGP updates when a
              change in reserved bandwidth >= the specified
              delta occurs on the interface.";
         }
         enum threshold-crossed {
           description
             "THRESHOLD-CROSSED indicates that
              the local system should trigger an update (and
              hence flood) the reserved bandwidth when the
```

```
              reserved bandwidth changes such that it crosses,
              or becomes equal to one of the threshold values.";
          }
        }
        description
          "The type of threshold that should be used to specify the
           values at which bandwidth is flooded. 'delta' indicates that
           the local system should flood IGP updates when a change in
           reserved bandwidth >= the specified delta occurs on the
           interface. Where 'threshold-crossed' is specified, the local
           system should trigger an update (and hence flood) the
           reserved bandwidth when the reserved bandwidth changes such
           that it crosses, or becomes equal to one of the threshold
           values.";
      }
      leaf delta-percentage {
        when "../threshold-type = 'delta'" {
          description
            "The percentage delta can only be specified when the
             threshold type is specified to be a percentage delta of
             the reserved bandwidth.";
        }
        type rt-types:percentage;
        description
          "The percentage of the maximum-reservable-bandwidth
           considered as the delta that results in an IGP update
           being flooded.";
      }
      leaf threshold-specification {
        when "../threshold-type = 'threshold-crossed'" {
          description
            "The selection of whether mirrored or separate threshold
             values are to be used requires user specified thresholds to
             be set.";
        }
        type enumeration {
          enum mirrored-up-down {
            description
              "mirrored-up-down indicates that a single set of
               threshold values should be used for both increasing
               and decreasing bandwidth when determining whether
               to trigger updated bandwidth values to be flooded
               in the IGP TE extensions.";
          }
          enum separate-up-down {
            description
              "separate-up-down indicates that a separate
               threshold values should be used for the increasing
```

```
              and decreasing bandwidth when determining whether
              to trigger updated bandwidth values to be flooded
              in the IGP TE extensions.";
         }
       }
       description
         "This value specifies whether a single set of threshold
          values should be used for both increasing and decreasing
          bandwidth when determining whether to trigger updated
          bandwidth values to be flooded in the IGP TE extensions.
          'mirrored-up-down' indicates that a single value (or set of
          values) should be used for both increasing and decreasing
          values, where 'separate-up-down' specifies that the increasing
          and decreasing values will be separately specified.";
     }
     leaf-list up-thresholds {
       when "../threshold-type = 'threshold-crossed'"
          + "and ../threshold-specification = 'separate-up-down'" {
         description
           "A list of up-thresholds can only be specified when the
            bandwidth update is triggered based on crossing a
            threshold and separate up and down thresholds are
            required.";
       }
       type rt-types:percentage;
       description
         "The thresholds (expressed as a percentage of the maximum
          reservable bandwidth) at which bandwidth updates are to be
          triggered when the bandwidth is increasing.";
     }
     leaf-list down-thresholds {
       when "../threshold-type = 'threshold-crossed'"
          + "and ../threshold-specification = 'separate-up-down'" {
         description
           "A list of down-thresholds can only be specified when the
            bandwidth update is triggered based on crossing a
            threshold and separate up and down thresholds are
            required.";
       }
       type rt-types:percentage;
       description
         "The thresholds (expressed as a percentage of the maximum
          reservable bandwidth) at which bandwidth updates are to be
          triggered when the bandwidth is decreasing.";
     }
     leaf-list up-down-thresholds {
       when "../threshold-type = 'threshold-crossed'"
          + "and ../threshold-specification = 'mirrored-up-down'" {
```

```
          description
            "A list of thresholds corresponding to both increasing
             and decreasing bandwidths can be specified only when an
             update is triggered based on crossing a threshold, and
             the same up and down thresholds are required.";
        }
        type rt-types:percentage;
        description
          "The thresholds (expressed as a percentage of the maximum
           reservable bandwidth of the interface) at which bandwidth
           updates are flooded - used both when the bandwidth is
           increasing and decreasing.";
      }
    }

    /* TE interface metric */

    grouping te-metric-config {
      description
        "TE interface metric grouping.";
      leaf te-metric {
        type te-types:te-metric;
        description
          "TE interface metric.";
      }
    }

    /* TE interface switching capabilities */

    grouping te-switching-cap-config {
      description
        "TE interface switching capabilities.";
      list switching-capabilities {
        key "switching-capability";
        description
          "List of interface capabilities for this interface.";
        leaf switching-capability {
          type identityref {
            base te-types:switching-capabilities;
          }
          description
            "Switching Capability for this interface.";
        }
        leaf encoding {
          type identityref {
            base te-types:lsp-encoding-types;
          }
          description
```

```
          "Encoding supported by this interface.";
      }
    }
  }

  grouping te-advertisements-state {
    description
      "TE interface advertisements state grouping.";
    container te-advertisements-state {
      description
        "TE interface advertisements state container.";
      leaf flood-interval {
        type uint32;
        description
          "The periodic flooding interval.";
      }
      leaf last-flooded-time {
        type uint32;
        units "seconds";
        description
          "Time elapsed since last flooding in seconds.";
      }
      leaf next-flooded-time {
        type uint32;
        units "seconds";
        description
          "Time remained for next flooding in seconds.";
      }
      leaf last-flooded-trigger {
        type enumeration {
          enum link-up {
            description
              "Link-up flooding trigger.";
          }
          enum link-down {
            description
              "Link-down flooding trigger.";
          }
          enum threshold-up {
            description
              "Bandwidth reservation up threshold.";
          }
          enum threshold-down {
            description
              "Bandwidth reservation down threshold.";
          }
          enum bandwidth-change {
            description
```

```
                "Bandwidth capacity change.";
            }
            enum user-initiated {
              description
                "Initiated by user.";
            }
            enum srlg-change {
              description
                "SRLG property change.";
            }
            enum periodic-timer {
              description
                "Periodic timer expired.";
            }
          }
          default "periodic-timer";
          description
            "Trigger for the last flood.";
        }
        list advertised-level-areas {
          key "level-area";
          description
            "List of level-areas that the TE interface is advertised
             in.";
          leaf level-area {
            type uint32;
            description
              "The IGP area or level where the TE interface link state
               is advertised in.";
          }
        }
      }
    }

    /* TE interface attributes grouping */

    grouping te-attributes {
      description
        "TE attributes configuration grouping.";
      uses te-metric-config;
      uses te-admin-groups-config;
      uses te-srlgs-config;
      uses te-igp-flooding-bandwidth-config;
      uses te-switching-cap-config;
      container state {
        config false;
        description
          "State parameters for interface TE metric.";
```

```
        uses te-advertisements-state;
      }
    }

    grouping te-all-attributes {
      description
        "TE attributes configuration grouping for all
         interfaces.";
      uses te-igp-flooding-bandwidth-config;
    }

    /*** End of TE interfaces device groupings ***/
    /**
     * TE device augmentations
     */

    augment "/te:te" {
      description
        "TE global container.";
      /* TE Interface Configuration Data */
      uses interfaces-grouping;
      container performance-thresholds {
        description
          "Performance parameters configurable thresholds.";
      }
    }

    /* TE globals device augmentation */

    augment "/te:te/te:globals" {
      description
        "Global TE device specific configuration parameters.";
      uses lsp-device-timers;
    }

    /* TE tunnels device configuration augmentation */

    augment "/te:te/te:tunnels/te:tunnel" {
      description
        "Tunnel device dependent augmentation.";
      leaf path-invalidation-action {
        type identityref {
          base te-types:path-invalidation-action-type;
        }
        description
          "Tunnel path invalidation action.";
      }
      uses lsp-device-timers;
```

```
   }

   /* TE LSPs device state augmentation */

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

   /* TE interfaces RPCs/execution Data */

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

Figure 11: TE device specific YANG module

7. Notifications

Notifications are a key component of any topology data model.

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

o Subscribe to notifications on a per-client basis.

o Specify subtree filters or XML Path Language (XPath) filters so that only contents of interest will be sent.

o Specify either periodic or on-demand notifications.

8. TE Generic and Helper YANG Modules

9. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested to be made.

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

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

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

```
Name:       ietf-te
Namespace:  urn:ietf:params:xml:ns:yang:ietf-te
Prefix:     te
Reference:  RFCXXXX

Name:       ietf-te-device
Namespace:  urn:ietf:params:xml:ns:yang:ietf-te-device
Prefix:     te-device
Reference:  RFCXXXX
```

10. Security Considerations

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

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

"/te/globals": This module specifies the global TE configurations on a device. Unauthorized access to this container could cause the device to ignore packets it should receive and process.

"/te/tunnels": This list specifies the configured TE tunnels on a device. 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.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

"/te/lsps": this list contains information state about established LSPs in the network. An attacker can use this information to derive information about the network topology, and subsequently orchestrate further attacks.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

"unnels-actions": using this RPC, an attacker can modify existing paths that may be carrying live traffic, and hence result to interruption to services carried over the network.

"/te/tunnels-path-compute": using this RPC, an attacker can retrieve secured information about the network provider which can be used to orchestrate further attacks.

The security considerations spelled out in the YANG 1.1 specification [RFC7950] apply for this document as well.

11. Acknowledgement

The authors would like to thank the members of the multi-vendor YANG design team who are involved in the definition of this model.

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, for providing useful feedback on this document.

12. Contributors

Himanshu Shah
Ciena

Email: hshah@ciena.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

13. Appendix A: Examples

This section contains examples of use of the model with RESTCONF [RFC8040] and JSON encoding.

For the example we will use a 4 nodes MPLS network were RSVP-TE tunnels can be setup. The loopbacks of each router are shown. The router network in figure X will be used across the section

```
 10.0.0.1        10.0.0.2      10.0.0.4
 +-----+         +------+      +------+
 |     |         |      |      |      |
 |  A  +---------+  B   +------+  D   |
 +--+--+         +------+      +--+---+
    |                             |
    |            +-------+        |
    |            |       |        |
    +------------+   C   +--------+
                 |       |
                 +-------+
                 10.0.0.3
```

Figure 12: Example TE topology.

13.1. Basic Tunnel Setup

This example uses the TE tunnel YANG data model defined in this document to create an RSVP-TE signaled Tunnel. First, the TE tunnel is created with no specific restrictions or constraints (e.g., protection or restoration). The TE tunnel ingresses on router A and egresses on router D.

In this case, the TE tunnel is created without specifying additional information about the primary paths.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
    Host: example.com
    Accept: application/yang-data+json
    Content-Type: application/yang-data+json
```


```
{
  "ietf-te:tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_2",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.4",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp"
    }
  ]
}
```


13.2. Global Named Path Constraints

This example uses the YANG data model to create a 'named path constraitnt' that can be reference by TE tunnels. The path constraint, in this case, limits the TE tunnel hops for the computed path.

```
POST /restconf/data/ietf-te:te/globals/named-path-constraints HTTP/1.1
    Host: example.com
    Accept: application/yang-data+json
    Content-Type: application/yang-data+json
```


```
{
  "ietf-te:named-path-constraint": {
          "name": "max-hop-3",
          "path-metric-bounds": {
            "path-metric-bound": {
              "metric-type": "te-types:path-metric-hop",
              "upper-bound": "3"
    }
   }
  }
}
```


13.3. Tunnel with Global Path Constraint

In this example, the previously created 'named path constraint' is applied to the TE tunnel created in Section 13.1.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
    Host: example.com
    Accept: application/yang-data+json
    Content-Type: application/yang-data+json
```


```
{
  "ietf-te:ietf-tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_4_1",
      "encoding": "te-types:lsp-encoding-packet",
      "description": "Simple_LSP_with_named_path",
      "admin-state": "te-types:tunnel-state-up",
      "source": "10.0.0.1",
      "destination": "10.0.0.4",
      "signaling-type": "path-setup-rsvp",
      "bidirectional": "false",
      "p2p-primary-paths": [
        {
          "p2p-primary-path": {
            "name": "Simple_LSP_1",
            "use-path-computation": "True",
            "named-path-constraint": "path-metric-delay-minimum"
          }
        }
      ]
    }
  ]
}
```


13.4. Tunnel with Per-tunnel Path Constraint

In this example, the a per tunnel path constraint is explicitly indicated under the TE tunnel created in Section 13.1 to constrain the computed path for the tunnel.

```
   POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
       Host: example.com
       Accept: application/yang-data+json
       Content-Type: application/yang-data+json
```


```
   {
     "ietf-te:tunnel": [
       {
         "name": "Example_LSP_Tunnel_A_4_2",
         "encoding": "te-types:lsp-encoding-packet",
         "admin-state": "te-types:tunnel-state-up",
         "source": "10.0.0.1",
         "destination": "10.0.0.4",
         "bidirectional": "false",
         "signaling-type": "te-types:path-setup-rsvp",
         "p2p-primary-paths": {
           "p2p-primary-path": [
             {
               "name": "path1",
               "path-metric-bounds": {
                 "path-metric-bound": {
                   "metric-type": "te-types:path-metric-delay-average",
                   "upper-bound": "3"
                 }
               }
             }
           ]
         }
       }
     ]
   }
```


13.5. Tunnel State

In this example, the 'GET' query is sent to return the state stored about the tunnel.

```
GET  /restconf/data/ietf-te:te/tunnels/tunnel="Example_LSP_Tunnel_A_4_1"
      /p2p-primary-paths/ HTTP/1.1
     Host: example.com
     Accept: application/yang-data+json
```

The request, with status code 200 would include, for example, the following json:

```
{
  "ietf-te:p2p-primary-paths": {
    "p2p-primary-path": [
      {
        "name": "path1",
        "path-computation-method": "te-types:path-locally-computed",
        "computed-paths-properties": {
          "computed-path-properties": [
            {
              "k-index": "1",
              "path-properties": {
                "path-route-objects": {
                  "path-route-object": [
                    {
                      "index": "1",
                      "numbered-node-hop": {
                        "node-id": "10.0.0.2"
                      }
                    },
                    {
                      "index": "2",
                      "numbered-node-hop": {
                        "node-id": "10.0.0.4"
                      }
                    }
                  ]
                }
              }
            }
          ]
        },
        "lsps": {
          "lsp": [
            {
              "tunnel-name": "Example_LSP_Tunnel_A_4_1",
              "node": "10.0.0.1 ",
              "lsp-id": "25356"
            }
          ]
        }
      }
    ]
  }
}
```

14. References

14.1. Normative References

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

[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>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <https://www.rfc-editor.org/info/rfc8776>.

14.2. 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

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Xufeng Liu
Volta Networks
Igor Bryskin
Huawei Technologies
Vishnu Pavan Beeram
Tarek Saad
Juniper Networks
Himanshu Shah
Ciena
Oscar Gonzalez De Dios
Telefonica

Expires: December 19, 2019 June 19, 2019

YANG Data Model for Traffic Engineering (TE) Topologies
draft-ietf-teas-yang-te-topo-22

Status of this Memo

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.

```
          ---------------                   
          | Native      |                   |  [ ] TE Node
          | TE-Topology |                   |  +++ TE Link
          ---------------                   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
         "Describibining 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   Y. Lee (Editor)
Internet Draft   Dhruv Dhody
Intended Status: Standard Track   Huawei
Expires: November 29, 2018   D. Ceccarelli
                                                             Ericsson
                                                         Igor Bryskin
                                                               Huawei
                                                    Bin Yeong Yoon
                                                                 ETRI

May 29, 2018

A Yang Data Model for ACTN VN Operation

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

Abstract

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

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

- o Characteristics of Access Points (APs) that describe customer's end point characteristics;

- o Characteristics of Virtual Network Access Points (VNAP) that describe How an AP is partitioned for multiple VNs sharing the AP and its reference to a Link Termination Point (LTP) of the Provider Edge (PE) Node;

- o Characteristics of Virtual Networks (VNs) that describe the customer's VNs in terms of VN Members comprising a VN, multi-source and/or multi-destination characteristics of VN Member, the VN's reference to TE-topology's Abstract Node;

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

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

1.1. Terminology

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

2. ACTN CMI context

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

```
              +-------+
              |  CNC  |
              +-------+
                  |
                  |    VN YANG + TE-topology YANG
                  |
       +------------------------+
       |         MDSC           |
       +------------------------+
```

Figure 1. ACTN CMI

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

2.1. Type 1 VN

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

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

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

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

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

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

```
                +---------------+
       L1 ------|               |------ L4
       L2 ------|     AN 1      |------ L7
       L3 ------|               |------ L8
                +---------------+
```

Figure 2. Abstract Node (One node topology)

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

2.2. Type 2 VN

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

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

```
      +-----------------------------------------------+
      |             S1                S2              |
      |              O---------------O                |
      |     ________/ \______         \               |
      |    /                  \         \             |
```

```
              |S3 /                   \ S4      \ S5         |
        L1----|-O----------------------O---------O-----------|------L4
              |   \                     \         \          |
              |    \                     \         \         |
              |     \ S6                  \ S7      \ S8     |
              |      O     ----------------O---------O-------|------L7
              |     / \   /                 \   ____/        |
              |S9  /   \ /S10                \ /             |
       L2-----|---O-----O---------------------O--------------|------L8
              |  /                          S11              |
       L3-----|--                                            |
              |                                              |
              +----------------------------------------------+
```

Figure 3. Type 2 topology

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

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

3. High-Level Control Flows with Examples

3.1. Type 1 VN Illustration

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

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

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

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

```
                 +---------------+
           L1 ------|               |------ L4
           L2 ------|     AN 1      |------ L7
           L3 ------|               |------ L8
                 +---------------+
```

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

```
               +--------+                                 +--------+
               |  CNC   |                                 |  MDSC  |
               +--------+                                 +--------+
                    |                                          |
                    |                                          |
   CNC POST TE-topo |  POST /nw:networks/nw:network/           |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-    |
   Matrix on one    |  matrices/tet:connectivity-matrix        |
   Abstract node    |----------------------------------------->|
                    |                        HTTP 200          |
                    |<-----------------------------------------|
                    |                                          |
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->| If there is
   AP, VNAP and VN- |                                          | multi-dest'n
   Members and maps |                                          | module, then
   to the TE-topo   |                        HTTP 200          | MDSC selects a
                    |<-----------------------------------------| src or dest'n
                    |                                          | and update
                    |                                          | ACTN VN YANG
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status: selected|
                    |  VN-members in case of multi s-d         |
                    |<-----------------------------------------|
                    |                                          |
```

3.2. Type 2 VN Illustration

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

```
               VN-Member 1       L1-L4

               VN-Member 2       L1-L7 (via S4 and S7)

               VN-Member 3       L2-L4

               VN-Member 4       L3-L8 (via S10)
```

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

```
                            S1               S2
                             O---------------O
                    ________/ \______         \
                   /                 \         \
               S3 /                   \ S4      \ S5
        L1------O----------------------O---------O------------------L4
                  \                     \         \
                   \                     \         \
                    \ S6                  \ S7      \ S8
                     O     ----------------O---------O--------------L5
                    / \   /                 \   ____/ \_____________L6
               S9  /   \ /S10                \ /
       L2---------O-----O---------------------O---------------------L7
                 /                          S11\____________________L8
       L3--------
```

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

```
              +--------+                                +--------+
```

```
               |  CNC   |                                 |  MDSC  |
               +--------+                                 +--------+
                    |                                          |
                    |                                          |
   CNC POST TE-topo |  POST /nw:networks/nw:network/            |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-    |
   Matrix on one    |  matrices/tet:connectivity-matrix        |
   Abstract node and|----------------------------------------->|
   Explicit paths in|                                          |
   The conn. Matrix |                     HTTP 200             |
                    |<-----------------------------------------|
                    |                                          |
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->|
   AP, VNAP and VN- |                                          |
   Members and maps |                                          |
   to the TE-topo   |                       HTTP 200           |
                    |<-----------------------------------------|
                    |                                          |
                    |                                          |
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status)          |
                    |<-----------------------------------------|
                    |                                          |
```

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

```
               +--------+                                 +--------+
               |  CNC   |                                 |  MDSC  |
               +--------+                                 +--------+
                    |                                          |
                    |                                          |
   CNC POST ACTN VN |                                          |
   Identifying AP,  |                                          |
   VNAP and VN-     |  POST /ACTN VN                           | MDSC populates
   Members          |----------------------------------------->| a single Abst.
                    |                     HTTP 200             | node topology
                    |<-----------------------------------------| by itself
                    |                                          |
```

```
   CNC POST the ACTN|  POST /ACTN VN                           |
   VN identifying   |----------------------------------------->|
   AP, VNAP and VN- |                                          |
   Members and maps |                                          |
   to the TE-topo   |                         HTTP 200         |
                    |<-----------------------------------------|
                    |                                          |
                    |                                          |
   CNC GET the ACTN |  GET /ACTN VN                            |
   VN YANG status   |----------------------------------------->|
                    |                                          |
                    |  HTTP 200 (ACTN VN with status)          |
                    |<-----------------------------------------|
                    |                                          |
```

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

4.1. Customer view of VN

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

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

4.2. Innovative Services

4.2.1. VN Compute

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

4.2.2. Multi-sources and Multi-destinations

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

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

4.2.3. Others

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

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

4.3. Summary

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

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

5. ACTN VN YANG Model (Tree Structure)

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

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

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

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

6. ACTN-VN YANG Code

The YANG code is as follows:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                }
                  leaf if-selected{
                    if-feature multi-src-dest;
                      type boolean;
                      default false;
```

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

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

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

<CODE ENDS>

7. JSON Example

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

The example in this section includes following VN

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

o VN2 (Type 2): Which maps to the single node topology abstract2 (node D2), this topology has an underlay topology (absolute) (see figure in section 3.2). This VN has a single VN member 105 (L1 to L5) and an underlay path (S4 and S7) has been set in the connectivity matrix of abstract2 topology;

o VN3 (Type 1): This VN has a multi-source, multi-destination feature enable for VN Member 104 (L1 to L4)/107 (L1 to L7) [multi-src] and VN Member 204 (L2 to L4)/304 (L3 to L4) [multi-dest] usecase. The selected VN-member is known via the field "if-selected" and the corresponding connectivity-matrix-id.

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

7.1. ACTN VN JSON


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

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

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

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

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

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

             ]
          }

       }
    }
```


7.2. TE-topology JSON


```
   {
        "networks": {
```

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

                        },
                        "connectivity-matrix": [
                          {
                            "id": 104,
                            "from": "1-0-1",
                            "to": "4-4-0"
                          },
                          {
                            "id": 107,
                            "from": "1-0-1",
                            "to": "7-7-0"
                          },
                          {
                            "id": 204,
                            "from": "2-0-2",
                            "to": "4-4-0"
                          },
                          {
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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


8. Security Considerations

TDB

9. IANA Considerations

TDB

10. Acknowledgments

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

11. References

11.1. Normative References

[TE-TOPO] X. Liu, et al., "YANG Data Model for TE Topologies", work in progress: draft-ietf-teas-yang-te-topo.

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

11.2. Informative References

[RFC7926] A. Farrel (Ed.), "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC 7926, July 2016.

[ACTN-REQ] Lee, et al., "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements, work in progress.

[ACTN-FWK] D. Ceccarelli, Y. Lee [Editors], "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ceccarelli-teas-actn-framework, work in progress.

[TE-MAP] Y. Lee, D. Dhody, and D. Ceccarelli, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang, work in progress.

[SERVICE-YANG] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained, work in progress.

[ACTN-PM] Y. Lee, et al., "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.

[OIF-VTNS] Virtual Transport Network Services 1.0 Specification, IA OIF-VTNS-1.0, April 2017.

12. Contributors

Contributor's Addresses

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

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

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

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

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

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

Authors' Addresses

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

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

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

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

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

Teas 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 A.Wang
Internet Draft China Telecom
Quintin Zhao
Boris Khasanov
HuaiMo Chen
Huawei Technologies
Penghui Mi
Tencent Company

Intended status: Experimental Track January 25, 2018
Expires: July 24, 2018

PCE in Native IP Network
draft-wang-teas-pce-native-ip-07.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the framework for CCDR traffic engineering within Native IP network, using Dual/Multi-BGP session strategy and PCE-based central control architecture.

The proposed central mode control framework conforms to the concept that defined in RFC " An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control".

The scenario and simulation results of CCDR traffic engineering is described in draft "CCDR Scenario, Simulation and Suggestion".

Table of Contents

1. Introduction

Draft [I-D.draft-wang-teas-ccdr] describes the scenario and simulation results for the CCDR traffic engineering. In summary, the requirements for CCDR traffic engineering in Native IP network are the following:

1) No complex MPLS signaling procedure.
2) End to End traffic assurance, determined QoS behavior.
3) Identical deployment method for intra- and inter- domain.
4) No influence to existing router forward behavior.
5) Can utilize the power of centrally control(PCE) and flexibility/robustness of distributed control protocol.
6) Coping with the differentiation requirements for large amount traffic and prefixes.
7) Flexible deployment and automation control.

This document defines the framework for CCDR traffic engineering within Native IP network, using Dual/Multi-BGP session strategy and CCDR architecture, to meet the above requirements in dynamical and central control mode. Future PCEP protocol extensions to transfer the key parameters between PCE and the underlying network devices(PCC) are provided in draft [draft-wang-pcep-extension-native-IP]

2. Dual-BGP framework for simple topology.

Dual-BGP framework for simple topology is illustrated in Fig.1, which is comprised by SW1, SW2, R1, R2. There are multiple physical links between R1 and R2. Traffic between IP11 and IP21 is normal traffic, traffic between IP12 and IP22 is priority traffic that should be treated differently.

Only Native IGP/BGP protocol is deployed between R1 and R2. The traffic between each address pair may change timely and the corresponding source/destination addresses of the traffic may also change dynamically.

The key idea of the Dual-BGP framework for this simple topology is the following:

1) Build two BGP sessions between R1 and R2, via the different loopback address lo0, lo1 on these routers.
2) Send different prefixes via the two BGP sessions. (For example, IP11/IP21 via the BGP pair 1 and IP12/IP22 via the BGP pair 2).
3) Set the explicit peer route on R1 and R2 respectively for BGP next hop of lo0, lo1 to different physical link address between R1 and R2.

So, the traffic between the IP11 and IP21, and the traffic between IP12 and IP22 will go through different physical links between R1 and R2, each type of traffic occupy the different dedicated physical links.

If there is more traffic between IP12 and IP22 that needs to be assured , one can add more physical links on R1 and R2 to reach the loopback address lo1(also the next hop for BGP Peer pair2). In this cases the prefixes that advertised by two BGP peer need not be changed.

If, for example, there is traffic from another address pair that needs to be assured (for example IP13/IP23), but the total volume of assured traffic does not exceed the capacity of the previous appointed physical links, then one need only to advertise the newly added source/destination prefixes via the BGP peer pair2, then the traffic between IP13/IP23 will go through the assigned dedicated physical links as the traffic between IP12/IP22.

Such decouple philosophy gives the network operator more flexible control ability on the network traffic, get the determined QoS assurance effect to meet the application’s requirement. No complex MPLS signal procedures is introduced, the router need only support native IP protocol.

```
                   |  BGP Peer Pair2  |
                   +------------------+
                   |lo1           lo1 |
                   |                  |
                   |  BGP Peer Pair1  |
                   +------------------+
       IP12        |lo0           lo0 |       IP22
       IP11        |                  |       IP21
       SW1-------R1-----------------R2-------SW2
                       Links Group
```

Fig.1 Design Philosophy for Dual-BGP Framework

3. Dual-BGP in large Scale Topology

When the assured traffic spans across one large scale network, as that illustrated in Fig.2, the dual BGP sessions cannot be established hop by hop especially for the iBGP within one AS. For such scenario, we should consider to use the Route Reflector (RR) to achieve the similar Dual-BGP effect, select one router which performs the role of RR (for example R3 in Fig.2), every other edge router will establish two BGP peer sessions with the RR, using their different loopback addresses respectively. The other two steps for traffic differentiation are same as one described in the Dual-BGP simple topology usage case.

For the example shown in Fig.2, if we select the R1-R2-R4-R7 as the dedicated path, then we should set the explicit peer routes on these routers respectively, pointing to the BGP next hop (loopback addresses of R1 and R7, which are used to send the prefix of the assured traffic) to the actual address of the physical link

```
              +------------R3--------------+
              |                            |
```

```
SW1-------R1-------R5---------R6-------R7--------SW2
           |        |          |        |
           +-------R2---------R4--------+
```

Fig.2 Dual-BGP Framework for large scale network

4. Multi-BGP for Extended Traffic Differentiation

In general situation, several additional traffic differentiation criteria exist, including:

- o Traffic that requires low latency links and is not sensitive to packet loss
- o Traffic that requires low packet loss but can endure higher latency
- o Traffic that requires lowest jitter path
- o Traffic that requires high bandwidth links

These different traffic requirements can be summarized in the following table:

Flow No.	Latency	Packet Loss	Jitter
1	Low	Normal	Don’t care
2	Normal	Low	Dont’t care
3	Normal	Normal	Low

Table 1. Traffic Requirement Criteria

For Flow No.1, we can select the shortest distance path to carry the traffic; for Flow No.2, we can select the idle links to form its end to end path; for Flow No.3, we can let all the traffic pass one single path, no ECMP distribution on the parallel links is required.

It is difficult and almost impossible to provide an end-to-end (E2E) path with latency, latency variation, packet loss, and bandwidth utilization constraints to meet the above requirements in large scale IP-based network via the traditional distributed routing protocol, but these requirements can be solved using the CCDR architecture since the PCE has the overall network view, can collect real network topology and network performance information about the underlying network, select the appropriate path to meet the various network performance requirements of different traffic type.

5. CCDR based framework for Multi-BGP strategy deployment.

With the advent of SDN concepts towards pure IP networks, it is possible now to accomplish the central and dynamic control of network traffic according to the application's various requirements.

The procedure to implement the dynamic deployment of Multi-BGP strategy is the following:

1) PCE gets topology and link utilization information from the underlying network, calculate the appropriate link path upon application's requirements.
2) PCE sends the key parameters to edge/RR routers(R1, R7 and R3 in Fig.3) to build multi-BGP peer relations and advertise different prefixes via them.
3) PCE sends the route information to the routers (R1,R2,R4,R7 in Fig.3) on forwarding path via PCEP, to build the path to the BGP next-hop of the advertised prefixes.
4) If the assured traffic prefixes were changed but the total volume of assured traffic does not exceed the physical capacity of the previous end-to-end path, then PCE needs only change the related information on edge routers (R1,R7 in Fig.3).
5) If volume of the assured traffic exceeds the capacity of previous calculated path, PCE must recalculate the appropriate path to accommodate the exceeding traffic via some new end-to-end physical link. After that PCE needs to update on-path routers to build such path hop by hop.

```
                               +----+
                     ***********+PCE +*************
                     *         +--*-+            *
                     *           / * \            *
                     *             *              *
                 PCEP*             *BGP-LS/SNMP   *PCEP
                     *             *              *
                     *             *           \  * /
                   \ * /           *            \ */
                    \*/-----------R3--------------*
                     |                            |
                     |                            |
          SW1-------R1-------R5---------R6-------R7--------SW2
```

```
                     |        |          |        |
                     |        |          |        |
                     +-------R2---------R4--------+
```

Fig.3 PCE based framework for Multi-BGP deployment

6. PCEP extension for key parameters delivery.

The PCEP protocol needs to be extended to transfer the following key parameters:
1) BGP peer address and advertised prefixes.
2) Explicit route information to BGP next hop of advertised prefixes.

Once the router receives such information, it should establish the BGP session with the peer appointed in the PCEP message, advertise the prefixes that contained in the corresponding PCEP message, and build the end to end dedicated path hop by hop. Details of communications between PCEP and BGP subsystems in router's control plane are out of scope of this draft and will be described in separate draft.[draft-wang-pce-extension for native IP]

The reason why we selected PCEP as the southbound protocol instead of OpenFlow, is that PCEP is suitable for the changes in control plane of the network devices, there OpenFlow dramatically changes the forwarding plane. We also think that the level of centralization that requires by OpenFlow is hardly achievable in many today's SP networks so hybrid BGP+PCEP approach looks much more interesting.

7. CCDR Deployment Consideration

CCDR framework requires the parallel work of 2 subsystems in router's control plane: PCE (PCEP) and BGP as well as coordination between them, so it might require additional planning work before deployment.

8.1 Scalability

In CCDR framework, PCE needs only to influence the edge routers for the prefixes differentiation via the multi-BGP deployment. The route information for these prefixes within the on-path routers were distributed via the traditional BGP protocol. Unlike the solution from BGP Flowspec, the on-path router need only keep the specific policy routes to the BGP next-hop of the differentiate prefixes, not the specific routes to the prefixes themselves. This can lessen the burden from the table size of policy based routes for the on-path routers, and has more scalability when comparing with the solution from BGP flowspec or Openflow.

8.2 High Availability

CCDR framework is based on the traditional distributed IP protocol. If the PCE failed, the forwarding plane will not be impacted, as the BGP session between all devices will not flap, and the forwarding table will remain the same. If one node on the optimal path is failed, the assurance traffic will fall over to the best-effort forwarding path. One can even design several assurance paths to load balance/hot standby the assurance traffic to meet the path failure situation, as done in MPLS FRR.
From PCE/SDN-controller HA side we will rely on existing HA solutions of SDN controllers such as clustering.

8.3 Incremental deployment

Not every router within the network support will support the PCEP extension that defined in [draft-wang-pce-extension-native-IP] simultaneously. For such situations, router on the edge of sub domain can be upgraded first, and then the traffic can be assured between different sub domains. Within each sub domain, the traffic will be forwarded along the best-effort path. Service provider can selectively upgrade the routers on each sub-domain in sequence.

8. Security Considerations

TBD

9. IANA Considerations

TBD

10. Conclusions

TBD

11. References

11.1. Normative References

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

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

[RFC8283] A.Farrel, Q.Zhao et al.," An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a Network with Central Control", [RFC8283], December 2017

11.2. Informative References

[I-D.draft-wang-teas-ccdr]

A.Wang, X.Huang et al. "CCDR Scenario, Simulation and Suggestion"

https://datatracker.ietf.org/doc/draft-wang-teas-ccdr/

[I-D. draft-ietf-teas-pcecc-use-cases]

Quintin Zhao, Robin Li, Boris Khasanov et al. "The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs

https://tools.ietf.org/html/draft-ietf-teas-pcecc-use-cases-00

March,2017

[draft-wang-pcep-extension for native IP]

Aijun Wang, Boris Khasanov et al. "PCEP Extension for Native IP Network" https://datatracker.ietf.org/doc/draft-wang-pce-extension-native-ip/

12. Acknowledgments

The authors would like to thank George Swallow, Xia Chen, Jeff Tantsura,Scharf Michael,Daniele Ceccarelli and Dhruv Dhody for their valuable comments and suggestions.

The authors would also like to thank Lou Berger, Adrian Farrel, Vishnu Pavan Beeram, Deborah Brungard and King Daniel for their suggestions to put forward this draft.

Authors’ Addresses

Aijun Wang
China Telecom
Beiqijia Town, Changping District
Beijing,China

Email: wangaj.bri@chinatelecom.cn

Quintin Zhao
Huawei Technologies
125 Nagog Technology Park
Acton, MA 01719
USA

EMail: quintin.zhao@huawei.com

Boris Khasanov
Huawei Technologies
Moskovskiy Prospekt 97A
St.Petersburg 196084
Russia

EMail: khasanov.boris@huawei.com

Huaimo Chen
Huawei Technologies
Boston, MA,
USA

EMail: huaimo.chen@huawei.com

Penghui Mi
Tencent
Tencent Building, Kejizhongyi Avenue,
Hi-techPark, Nanshan District,Shenzhen 518057, P.R.China

Email kevinmi@tencent.com

Raghavendra Mallya
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089 USA

Email: rmallya@juniper.net

Shaofu Peng
ZTE Corporation
No.68 Zijinghua Road,Yuhuatai District
Nanjing 210012
China

Email: peng.shaofu@zte.com.cn

TEAS Working Group   Quintin Zhao
Internet-Draft   Robin Li
Intended status: Experimental   Boris Khasanov, Ed.
Expires: May 25, 2017   Huawei Technologies
                                                                 King Ke
                                                 Tencent Holdings Ltd.
                                                            Luyuan Fang
                                                               Microsoft
                                                              Chao Zhou
                                                           Cisco Systems
                                                             Boris Zhang
                                                 Telus Communications
                                                       Artem Rachitskiy
                                                           Anton Gulida
                                                  Mobile TeleSystems JLLC
                                                        October 26, 2016

The Use Cases for Using PCE as the Central Controller(PCECC) of LSPs
draft-zhao-teas-pcecc-use-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 reducing existing complexity.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.

Requirements Language

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

This Internet-Draft will expire on May 25, 2017.

Copyright Notice

Table of Contents

1. Introduction

1.1. Background

In many 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 provides such flexibility and programmability for that case.

By migrating to the SDN enabled network from the existing network, service providers and network operators must have a solution which they can easily evolve from the existing network into the fully SDN enabled network while keeping scalability of the network services, guarantee robustness, availability, flexibility etc.

Taking into account the smooth transition from existing network to the new SDN enabled network with optimal cost, re-usage of the existing PCE components in network to be function of the central (SDN) controller is one choice, that not only achieves the goal of having centralized control 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.

Segment Routing (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 MPLS network.

The PCECC solution proposed in this document allows creation of dynamic MPLS network that is eventually controlled and deployed without the RSVP-TE protocol or extended IGP protocol with node/adjacency segment identifiers 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 way for users.

1.2. Using the PCE as the Central Controller (PCECC) Approach

PCECC not only can remove the existing MPLS signaling totally from the control plane without losing any MPLS functionalities, but also will achieve 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  | |
| +--------+   +--------+   +--------+   +--------+   +--------+ |
|                                                                  |
+------------------------------------------------------------------+
```

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 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 to 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 providers networks, 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 | |
   | +--------+        +--------+ |    | +--------+        +--------+ |
   |                              |    |                              |
   +------------------------------+    +------------------------------+
```

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
                        +-----------+
```

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:

- o 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.

- o 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:

- o 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.

- o 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
                        +-----------+
```

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 example, 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
                        +-----------+
```

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)
                       +-----------+      +-----------+
```

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)
                       +-----------+      +-----------+
                             .                  .
                             .                  .
```

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                          |       |
|    +-----------------------------------------------------+       |
|     ^                                     ^            ^         |
|     | PCEP                                |   PCEP     |         |
|     V    <-----RSVP------>                V            V         |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
| | NODE 1 |   | NODE 2 |   | NODE 3 |   | NODE 4 |   | NODE 5 |   |
| |        |...|        |...|        |...|        |...|        |   |
| | Legacy |if1| Legacy |if2|Legacy  |if3| PCECC  |if4| PCECC  |   |
| |  Node  |   |  Node  |   |Enabled |   |Enabled |   | Enabled|   |
| +--------+   +--------+   +--------+   +--------+   +--------+   |
|                                                                   |
+------------------------------------------------------------------+
```

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 Node4 and Node5 are centrally controlled and other nodes are legacy nodes.

- o Node1 sends a path request message towards PCECC for the setup of LSP destinating to Node5.

- o PCECC sends to node1 a reply message for LSP setup with the path: (Node1, if1),(Node2, if2), (Node3, if3), (Node4, if4), Node5.

- o Node1, Node2, Node3 will setup the LSP to Node5 using the local labels as usual.

- o Then the PCECC will program the outsegment of Node3, the insegment/ ousegment of Node4, and the insegment for Node5.

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.

Example described in this section is based on network configurations illustrated using the following figure:

```
               +-------------------------------------------+
               |                   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  |
    +--------+ |     +--------+   +--------+   +--------+  |  +--------+
               |                                           |
               +-------------------------------------------+
```

Example: 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. Using PCECC for Traffic Classification Information

When a TE-LSP is set up, the head end needs to know:

o how to use it

o What traffic to send on the LSP

o Whether it is a virtual link

o Whether to advertise it in the IGP

o What bits of this information to signal to the tail end

PCEP allows an Active PCE to set up or modify LSPs. But we have no way to tell the head end how to use the LSP. This is because of history. It used to be the LER that made the request of the PCE, so it knew why it wanted the LSP.

With the PCECC architecture by extending the PCEP protocols, it is easy to carry this information such as how to use the LSP, how to advertise the LSP and other extra signaling information.

11. PCECC Load Balancing (LB) Use Case

Very often many service providers use TE tunnels for solving issues with non-deterministic paths in their networks. One example of such applications is usage of TEs in the mobile backhaul (MBH).
Let’s consider the following typicall topology.

```
                               TE1 --------------->
 +---------+    +--------+    +---------+    +---------+    +-------+  +---+
 | Access  |----| Access |----| AGG 1   |----| AGG N-1|----|Core 1|--|SR1|
 | SubNode1|    | Node 1 |    +---------+    +---------+    +-------+  +---+
 +---------+    +--------+         | |            | ^           |
      |   Access     |    Access    | AGG Ring 1   | |           |
      |  SubRing 1   |    Ring 1    | |            | |           |
 +---------+    +--------+    +---------+           | |           |
 | Access  |    | Access |    | AGG 2   |           | |           |
 | SubNode2|    | Node 2 |    +---------+           | |           |
 +---------+    +--------+         | |            | |           |
      |             |              | |            | |           |
      |             |              | +----TE2----|-+           |
 +---------+    +--------+    +---------+    +---------+    +-------+  +---+
 | Access  |    | Access |----| AGG 3   |----| AGG N   |----|Core N|--|SRn|
 | SubNodeN|----| Node N |    +---------+    +---------+    +-------+  +---+
 +---------+    +--------+
```

This MBH architecture uses L2 access rings and subrings. L3 starts at aggregation. For the sake of simplicity here we have only one access subring,access ring and aggregation ring (AGG1...AGGN), connected by Nx10GE interfaces. Aggregation domain runs its own IGP. There are two Egress routers (AGG N-1,AGG N) that are connected to the Core domain via L2 interfaces. Core also have connections to service routers, RSVP TEs are used for MPLS transport inside the ring. There could be at least 2 tunnels (one way) from each AGG router to egress AGG routers. There are also many L2 access rings connected to AGG routers.

Service deployment made by means of either L2VPNs (VPLS) or L3VPNs. Those services use MPLS TE as transport towards egress AGG routers. TE tunnels could be also used as transport towards service routers in case of seamless MPLS based architecture in the future.

There is a need to solve the following tasks:
o Perform automatic LB amongst TE tunnels according to current traffic load
o TE bandwidth (BW) management: Provide guaranteed BW for specific service: HSI,IPTV, etc., provide time-based BW reservation (BoD)
o Simplify development of TE tunnels (go away from manual provisioning)
o Provide flexibility for Service Router placement (anywhere in the network by creation of transport LSPs to them)

Since other tasks are considered in other PCECC use cases above, hereafter we will focus only on load balancing (LB) task. LB task could be solved by means of PCECC in the following way:

o After application or network service or operator will ask SDN controller (PCECC) for LSP based LB between AGG X and AGG N/AGG N-1 (egress AGG routers which have connections to core) via North Bound Interface (NBI such as REST API), PCECC SHOULD ask for constrains for that particular calculation (i.e. LSP type: traditional CR-LSP or SR-TE LSP, bandwidth, inclusion or exclusion specific links or nodes, number of paths, shortest path or minimum cost tree, need for disjoint LSP paths etc.).
o PCECC MUST calculate N P2P LSPs according to given constrains, calculation is based on results of Objective Function (OF), that includes same source and destination routers IDs, same or different bandwidth (BW) , different links (in case of disjoint paths) and other constrains from Step 1.
o Depending on given LSP type (CR-LSP or SR-TE), PCECC SHOULD create different labels (aka different label spaces, it MAY also require label space negotiation procedure between PCECC and PCCs) for calculated LSPs from egress nodes AGG N-1 and AGG N towards ingress AGG X node.
o PCECC SHOULD send PCInitiate PCEP message [I-D.crabbe-pce-pce-initiated-lsp] towards ingress AGG X router(PCC) for each of N LSPs and receives PCRpt PCEP message [I-D.ietf-pce-stateful-pce] back from him.
o If LSP type is CR-LSP, PCECC MUST send PCLabelUpd [I-D.zhao-pce-pcep-extension-for-pce-controller] PCEP message to each node along the path with label information for each of N LSPs. If LSP type is SR-TE, PCECC also MUST send PCLabelUpd PCEP message

to each node along the path with label information (Node-ID and Adjacency-ID segment (label) list) specific to that node. Then PCECC SHOULD send PCUpd PCEP message to the ingress AGG X router with information about new LSP and AGG X(PCC) SHOULD send PCEP PCRpt back with LSP status:Up.

o Now each router along the LSP has corresponding label forwarding state for each of N LSPs.

o AGG X as ingress router now have N LSPs towards AGG N and AGG N-1 which are available for installing to router’s RIB and LB of traffic between them. Traffic distribution between those LSPs depends on particular realization of hash-function on that router.

o Since PCECC MUST know as LSDB as TEDB (TE state) he can manage and prevent possible oversubscriptions and limit number of available LB states.

12. Using reliable P2MP TE based multicast delivery for distributed computations (MapReduce-Hadoop)

MapReduce model of distributed computations in computing clusters is widely deployed. In Hadoop 1.0 architecture MapReduce operations on big data performs by means of Master-Slave architecture in the Hadoop Distributed File System (HDFS),where NameNode has the knowledge about resources of the cluster and where actual data (chunks) for particular task are located (which DataNode). Each chunk of data (64MB or more) should have 3 saved copies in different DataNodes based on their proximity.

Proximity level currently has semi-manual allocation and based on Rack IDs (Assumption is that closer data are better because of access speed/smaller latency).

JobTracker node is responsible for computation tasks, scheduling across DataNodes and also have Rack-awareness. Currently transport protocols between NameNode/JobTracker and DataNodes are based on IP unicast. It has simplicity as pros but has numerous drawbacks related with its flat approach.

It is clear that we should go beyond of one DC for Hadoop cluster creation and move towards distributed clusters. In that case we need to handle performance and latency issues.
Latency depends on speed of light in fiber links and also latency introduced by intermediate devices in between. The last one is closely correlated with network device architecture and performance. Current performance of NPU based routers should be enough for creating distribute Hadoop clusters with predicted latency. Performance of SW based routers (mainly as VNF) together with additional HW features such as DPDK are promising but require additional research and testing.

Main question is how can we create simple but effective architecture for distributed Hadoop cluster?

There are number of researches [Multicast Tree Map-Reduce...] which show how usage of multicast tree could improve speed of resource or cluster members discovery inside the cluster as well as increase redundancy in communications between cluster nodes.

Is traditional IP based multicast enough for that? We doubt it because it requires additional control plane (IGMP, PIM) and a lot of signaling, that is not suitable for high performance computations, that are very sensitive to latency.

P2MP TE tunnels looks much more suitable as potential solution for creation of multicast based communications between Master and Slave nodes inside cluster. Obviously these P2MP tunnels should be dynamically created and turned down (no manual intervention). Here is there PCECC comes to play. His main task is to create optimal topology of each partucular request for MapReduce computation and also create P2MP tunnels with needed parameters such as badnwidth and delay.

This solution would require to use MPLS label based forwarding inside the cluster. Usage of label based forwarding inside DC was proposed by Yandex [MPLS in DC...] Technically it is already possible because mpls on switches is already supported by some vendors, mpls aslo exists on Linux and OVS.

The following framework can make this task:

```
                      +--------+
                      |  APP   |
                      +--------+
                           | NBI (REST API,...)
                           |
               PCEP       +----------+  REST API
        +---------+   +---|  PCECC   |----------+
        | Client  |---|---|          |          |
        +---------+   |   +----------+          |
                |     |       | |  |            |
                +-----|---+   |PCEP|            |
             +--------+   |   | |  |            |
             |            |   | |  |            |
             | REST API   |   | |  |            |
             |            |   | |  |            |
   +-------------+        |   | |  |           +----------+
   | Job Tracker |        |   | |  |           | NameNode |
   |             |        |   | |  |           |          |
   +-------------+        |   | |  |           +----------+
           +------------------+ |  +-----------+
           |              |     |              |
       |---+-----P2MP TE--+-----|-----------|  |
   +-----------+       +-----------+      +-----------+
   | DataNode1 |       | DataNode2 |      | DataNodeN |
   |TaskTracker|       |TaskTracker| .... |TaskTracker|
   +-----------+       +-----------+      +-----------+
```

Communication between Master nodes (JobTracker and NameNode) and PCECC via REST API MAY be either done directly or via cluster manager such as Mesos.

Phase 1: Distributed cluster resources discovery
During this phase Master Nodes SHOULD identify and find available Slave nodes according to computing request from application (APP). NameNode SHOULD query PCECC about available DataNodes, NameNode MAY provide additional constrains to PCECC such as topological proximity, redundancy level.

PCECC SHOULD analyze the topology of distributed cluster and perform constrain based path calculation [RFC7334] from client towards most suitable NameNodes. PCECC SHOULD reply to NameNode the list of most suitable DataNodes and their resource capabilities. Topology discovery mechanism for PCECC will be added later to that framework.

Phase 2: PCECC SHOULD create P2MP LSP from client towards those DataNodes by means of PCLabelUpd [I-D.zhao-pce-pcep-extension-for-pce -controller] PCEP messages following previously calculated path.

Phase 3. NameNode SHOULD send this information to client, PCECC informs client about optimal P2MP path towards DataNodes via PCEP PCUpd message.

Phase 4. Client sends data blocks to those DataNodes for writing via created P2MP tunnel.

When this task will be finished, P2MP tunnel MAY be turned down.

13. PCECC and Inter-AS TE

There are three signalling options for establishing Inter-AS TE LSP: contiguous TE LSP [RFC5151], stitched inter-AS TE LSP [RFC5150], nested TE LSP [RFC4206].

Requirements for PCE-based Inter-AS setup [RFC5376] describe the approach and PCEP fucntionality that are needed for establishing Inter-AS TE LSPs.

[RFC5376] also gives Inter- and Intra-AS PCE Reference Model that is provided below in shorten form for the sake of simplicity.

```
     Inter-AS       Inter-AS
     PCC <-->PCE1<--------->PCE2
      ::      ::            ::
      ::      ::            ::
      R1----ASBR1====ASBR3---R3---ASBR5
      |   AS1 |        |    PCC     |
      |       |        |    AS2     |
      R2----ASBR2====ASBR4---R4---ASBR6
      ::                    ::
      ::                    ::
   Intra-AS              Intra-AS
      PCE                   PCE
```

Shorten form of Inter- and Intra-AS PCE Reference Model [RFC5376]

Hereatfter we will discuss a simplified Inter-AS case when both AS1 and AS2 belong to the same service provider administration. In that case Inter and Intra-AS PCEs could be combined in one single PCE if such combined PCE performance is enough for handling all Path Computation Requests. Even more in that particular case we potentially could use single PCE for both ASes if his scalability and performance are enough, we just will need interfaces (PCEP and BGP-LS) to both domains. SDN controller’s redundancy mechanisms are out of scope in our case. Thus routers in AS1 and AS2 (PCCs) will send Path Computation Requests towards same PCE.

```
             +----BGP-LS------+ +------BGP-LS-----+
             |                | |                 |
      +-PCEP-|----++-+-------PCECC-----PCEP--++-+-|-------+
    +-:------|----::-:-+                 +--::-:-|-------:---+
    | :      v    :: : |                 |  :: : v       :   |
    | :     RR1   :: : |                 |  :: : RR2     :   |
    | v          v: : |      LSP1        |  :: v         v   |
    | R1---------ASBR1=======================ASBR3--------R3  |
    | |           v : |                  |  :v           |   |
    | +----------ASBR2=======================ASBR4---------+  |
    | |   Region 1   : |                 |  : Region 1    |   |
    |----------------:-|                 |--:-------------|--|
    | |              v |       LSP2      |  v             |   |
    | +----------ASBR5=======================ASBR6---------+  |
    |     Region 2     |                 |       Region 2     |
    +------------------+ <--------------> +------------------+
       MPLS Domain 1         Inter-AS         MPLS Domain 2
    <=======AS1=======>                  <========AS2=======>
```

Particular case of Inter-AS PCE Reference Model

In one particular case of PCECC Inter-AS TE scenario service provider controls both domains (AS1 and AS2), each of them have own IGP and MPLS transport. The need is to setup Inter-AS LSPs for transporting different services on top of them (Voice,L3 VPN etc.) Inter-AS links with different capacity exist in several regions. The task is not only to provision those Inter-AS LSPs with given constrains but also calculate the path and pre-setup the backup Inter-AS LSPs that will be used if main LSP fails.

For the figure above it would be that LSP1 from R1 to R3 SHOULD go via ASBR1 and ASBR3, and it is the main Inter-AS LSP. R1-R3 LSP2 that SHOULD go via ASBR5 and ASBR6 is the backup one. Depending on Inter-AS TE type, backup LSP could be used either by head-end R1 or ASBR1.

After the addition of PCECC functionality to PCE (SDN controller), PCECC based Inter-AS TE model SHOULD follow as PCECC usecase for TE LSP (case 6 above) as requirements of [RFC5376] with the following details:
o Since PCECC MUST know the topology of both domains AS1 and AS2, PCECC MUST establish BGP-LS peering with routers (or RRs) in both domains
o PCECC MUST have SBI (PCEP) connectivity towards all routers in both domains (see also section 4 in [RFC5376])
o After operator’s application or service orchetsrator will create request for topology of specific service, PCECC SHOULD receive that request via NBI (NBI type is implementation dependent, MAY be NETCONF/Yang, REST etc.). Then PCECC SHOULD calculate Objective Function (OF) for optimal path with given constrains (i.e. LSP type, bandwidth etc.), including those from [RFC5376]: priority, AS sequence, preffered ASBR, disjoint paths, protection. On this step we would have two paths: R1-ASBR1-ASBR3-R3, R1-ASBR5-ASBR6-R3
o Depending on given LSP type (CR-LSP or SR-TE), PCECC SHOULD create different labels (aka different label spaces, it MAY also require label space negotiation procedure between PCECC and PCCs) for calculated LSPs from egress node in one AS towards ingress in another AS.
o PCECC SHOULD send PCInitiate PCEP message [I-D.crabbe-pce-pce-initiated-lsp] towards ingress router R1 (PCC) in AS1 and receive PCRpt PCEP message [I-D.ietf-pce-stateful-pce] back from him.
o If LSP type is CR-LSP, PCECC MUST send PCLabelUpd [I-D.zhao-pce-pcep-extension-for-pce-controller] PCEP message to each node along the path (ASBR1-ASBR3-R3, ASBR5-ASBR6-R3) in both ASes with label information for that LSP.
If LSP type is SR-TE, PCECC also MUST send PCLabelUpd PCEP message to each node along the path in aboth Ases with label information (Node-ID and Adjacency-ID segment (label) list) specific to that node.
o Then PCECC SHOULD send PCUpd PCEP message to the ingress router R1 in AS1 with information about new LSP and the R1 router SHOULD send PCEP PCRpt back

with LSP1 and LSP2 status:Up.

o After that step R1 SHOULD have main and backup TEs (LSP1 and LSP2) towards R3 up. It is up to implementation how to put this TEs to R1's RIB and how to make switchover to backup LSP2 if LSP1 fails.

14. 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].

15. IANA Considerations

This document does not require any action from IANA.

16. Security Considerations

TBD.

17. Acknowledgments

We would like to thank Robert Tao, Changjiang Yan, Tieying Huang, Adrian Farrel, Sergio Belotti and Dieter Beller, Andrey Elperin and Evgeniy Brodskiy for their useful comments and suggestions.

18. References

18.1. 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>.

[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>.

18.2. Informative References

[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>.

[RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <http://www.rfc-editor.org/info/rfc5541>.

[RFC5376] N. Bitar, R. Zhang, K. Kumaki "Inter-AS Requirements for the Path Computation Element Communication Protocol (PCECP)", RFC 5376, DOI 10.17487/RFC5376, November 2008 <http://www.rfc-editor.org/info/rfc5376>.

[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-14 (work in progress), May 2016.

[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-05 (work in progress), October 2015.

[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-06 (work in progress), December 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., Raszuk, R., and L. Fang, "Usecases of MPLS Global Label", draft-li-mpls-global-label-usecases-03 (work in progress), October 2015.

[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., Li, Z., Dhody, D., and C. Zhou, "PCEP Procedures and Protocol Extensions for Using PCE as a Central Controller (PCECC) of LSPs", draft-zhao-pce-pcep-extension-for-pce-controller-03 (work in progress), March 2016.

[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-02 (work in progress), December 2015.

[MPLS in DC...]
Afanasiev, D., Ginsburg, D., "MPLS in DC and inter-DC networks: the unified forwarding mechanism for network programmability at scale "

[Multicast Tree Map-Reduce...]
Lee, Kyungyong., Dr. Boykin, P. Oscar., Dr.Figueiredo, Renato J., "Multicast Tree Map-Reduce: Self-organizing Resource Discovery and Monitoring using Structured P2P Systems"

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

Boris Khasanov
Huawei Technologies
Moskovskiy Prospekt 97A
St.Petersburg 196084
Russia

EMail: khasanov.boris@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

EMail: Boris.zhang@telus.com

Artem Rachitskiy
Mobile TeleSystems JLLC
Nezavisimosti ave., 95
Minsk 220043
Belarus

EMail: arachitskiy@mts.by

Anton Gulida
Mobile TeleSystems JLLC
Nezavisimosti ave., 95
Minsk 220043
Belarus

EMail: agulida@mts.by

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 9, 2017
Y. Zhuang, Ed.
Q. Wu
H. Chen
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.

Status of This Memo

Copyright Notice

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:

- In the network nodes where the resources exist;
- 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.

- 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.

- If other sources of planned LSPs are allowed, they can request path computation and resource reservation from the centralized PCE using PCEP [RFC5440].

- 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.

- 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