PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 10, 2021

H. Chen
Futurewei
M. Toy
Verizon
X. Liu
Volta Networks
L. Liu
Fujitsu
Z. Li
China Mobile
January 6, 2021

PCEP Link State Abstraction
draft-chen-pce-h-connect-access-08

Abstract

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for a child PCE to abstract its domain information to its parent for supporting a hierarchical PCE system.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A hierarchical PCE architecture is described in RFC 6805, in which a parent PCE maintains an abstract domain topology, which contains its child domains (seen as vertices in the topology) and the connections among them.

For a domain for which a child PCE is responsible, connections attached to the domain may comprise inter-domain links and Area Border Routers (ABRs). For a parent PCE to have the abstract domain topology, each of its child PCEs needs to advertise its connections to the parent PCE.

In addition to the connections attached to the domain, there may be some access points in the domain, which are the addresses in the domain to be accessible outside of the domain. For example, an address of a server in the domain that provides a number of services to users outside of the domain is an access point.

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for a child PCE to advertise the information about its connections and access points to its parent PCE and for the parent PCE to build and maintain the abstract domain topology based on the information. The extensions may reduce configurations, thus simplify operations on a PCE system.

A child PCE is simply called a child and a parent PCE is called a parent in the following sections.

2. Terminology

ABR: Area Border Router. Router used to connect two IGP areas (Areas in OSPF or levels in IS-IS).

ASBR: Autonomous System (AS) Border Router. Router used to connect together ASes via inter-AS links.

TED: Traffic Engineering Database.

This document uses terminology defined in [RFC5440].

3. Conventions Used in This Document

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

4. Connections and Accesses

A connection is an inter-domain link between two domains in general. An ABR is also a connection, which connects two special domains called areas in a same Autonomous System (AS).

An access point in a domain is an address in the domain to be accessible to the outside of the domain. An access point is simply called an access.

4.1. Information on Inter-domain Link

An inter-domain link connects two domains in two different ASes. Since there is no IGP running over an inter-domain link, we may not obtain the information about the link generated by an IGP. We may suppose that IP addresses are configured on inter-domain links.

For a point-to-point (P2P) link connecting two ABSRs A and B in two different domains, from A's point of view, the following information about the link may be obtained:

1) Link Type: P2P
2) Local IP address
3) Remote IP address
4) Traffic engineering metric
5) Maximum bandwidth
6) Maximum reservable bandwidth
7) Unreserved bandwidth
8) Administrative group
9) SRLG

We will have a link ID if it is configured; otherwise no link ID (i.e., the Router ID of the neighbor) may be obtained since no IGP adjacency over the link is formed.

For a broadcast link connecting multiple ASBRs in a number of domains, on each of the ASBRs X, the same information about the link as above may be obtained except for the followings:

a) Link Type: Multi-access,
b) Local IP address with mask length, and
c) No Remote IP address.

In other words, the information about the broadcast link obtained by ASBR X comprises a), b), 4) to 9), but does not include any remote IP address or link ID. We will have a link ID if it is configured; otherwise no link ID (i.e., the interface address of the designated router for the link) may be obtained since no IGP selects it.

A parent constructs an abstract AS domain topology after receiving the information about each of the inter-domain links described above from its children.

RFC 5392 and RFC 5316 describe the distributions of inter-domain links in OSPF and IS-IS respectively. For each inter-domain link, its neighboring AS number and neighboring ASBR Identity (TE Router ID) need to be configured in IGP (OSPF or IS-IS).

In addition, an IGP adjacency between a network node running IGP and a PCE running IGP as a component needs to be configured and fully established if we want the PCE to obtain the inter-domain link information from IGP.

These configurations and IGP adjacency establishment are not needed if the extensions in this draft are used.

RFC 7752 (BGP-LS) describes the distributions of TE link state information including inter-domain link state. A BGP peer between a network node running BGP and a PCE running BGP as a component needs to be configured and the peer relation must be established before the PCE can obtain the inter-domain link information from BGP. However, some networks may not run BGP.

4.2. Information on ABR

For an AS running IGP and containing multiple areas, an ABR connects two or more areas. For each area connected to the ABR, the PCE as a child responsible for the area sends its parent the information about the ABR, which indicates the identifier (ID) of the ABR.

A parent has the information about each of its children, which includes the domain such as the area for which the child is responsible. The parent knows all the areas to which each ABR connects after receiving the information on the ABR from each of its children.

4.3. Information on Access Point

For an IP address in a domain to be accessible outside of the domain, the PCE as a child responsible for the domain sends its parent the information about the address.

The parent has all the access points (i.e., IP addresses) to be accessible outside of all its children' domains after receiving the information on the access points from each of its children.

5. Extensions to PCEP

This section focuses on procedures for abstracting domain information after briefing messages containing the abstract information.

5.1. Messages for Abstract Information

A child abstracts its domain to its parent through sending its parent a message containing the abstract information on the domain. After the relation between the child and the parent is determined, the parent has some information on the child, which includes the child's ID and domain. The message does not need to contain this information. It comprises the followings:

- For new or updated Connections and Accesses,
  * Indication of Update Connections and Accesses
  * Detail Information about Connections and Accesses
- For Connections and Accesses down,
  * Indication of Withdraw Connections and Accesses
  * ID Information about Connections and Accesses

For a P2P link from ASBR A to B and a broadcast link connecting to A, the detail information on the links includes A's ID, the information on the P2P link and the information on the broadcast link described in Section 4. The ID information on the links includes A's ID, 1) to 3) for the P2P link and a) to b) for the broadcast link described in Section 4. A link ID for a link is included if it is configured.

For an ABR X, the information on X includes X's ID and a flag indicating that X is ABR.

For an Access X (address), the detail information on X includes X and a cost associated with it. The ID information on X is X itself.

There are a few ways to encode the information above into a message. For example, one way is to extend an existing Notification message for including the information. Another way is to use a new message. These are put in Appendix A for your reference.

5.2. Procedures

5.2.1. Child Procedures

5.2.1.1. New or Changed Connections and Accesses

After a child determines its parent, it sends the parent a message containing the information about the connections (i.e., inter-domain links and ABRs) from its domain to its adjacent domains and the access points in its domain.

For any new or changed inter-domain links, ABRs and access points in the domain for which a child is responsible, the child sends its parent a message containing the information about these links, ABRs and access points with indication of Update Connections and Accesses.

For example, for a new inter-domain P2P link from ASBR A in a child's domain to ASBR B in another domain, the child sends its parent a message containing an indication of Update Connections and Accesses, A's ID, and the detail information on the link described in section 4.1.

For multiple new or changed inter-domain links from ASBR A, the child sends its parent a message having an indication of Update Connections and Accesses, and A's ID followed by the detail information about each of the links.

In another example, for a new or changed inter-domain broadcast link connected to ASBR X, an ABR Y and an access point 10.10.10.1/32 with cost 10 in a child's domain, the child sends its parent a message containing an indication of Update Connections and Accesses, and X's ID followed by the detail information about the link attached to X and the detail information about ABR Y, and the information on access 10.10.10.1/32 with cost 10.

For changes on the attributes (such as bandwidth) of an inter-domain link, a threshold may be used to control the frequency of updates that are sent from a child to its parent. At one extreme, the threshold is set to let a child send its parent a update message for any change on the attributes of an inter-domain link. At another extreme, the threshold is set to make a child not to send its parent any update message for any change on the attributes of an inter-domain link. Typically, the threshold is set to allow a child to send its parent a update message for a significant change on the attributes of an inter-domain link.

5.2.1.2. Connections and Accesses Down

For any inter-domain links, ABRs and access points down in the domain for which a child is responsible, the child sends its parent a message containing the information about these links, ABRs and access points with indication of Withdraw Connections and Accesses.

For example, for the inter-domain P2P link from ASBR A down, the child sends its parent a message containing an indication of Withdraw

Connections and Accesses, and A's ID, which is followed by the ID information about the link.

For multiple inter-domain links from ASBR A down, the child sends its parent a message having an indication of Withdraw Connections and Accesses, and A's ID, which is followed by the ID information about each of the links.

5.2.1.3. Child and Parent in Same Organization

If a child and its parent are in a same organization, the child may send its parent the information inside its domain. For a parent, after all its children in its organization send their parent the information in their domains, connections and access points, it has in its TED the detail information inside each of its children's domains and the connections among these domains. The parent can compute a path crossing these domains directly and efficiently without sending any path computation request to its children.

5.2.1.4. Child as a Parent

There are a few ways in which a child as a parent abstracts its domain information to its parent.

One way is that the child sends its parent all its domain information if the child and the parent are in a same organization. The information includes the detail network topology inside each of the child's domains, the inter-domain links connecting the domains that the child's children are responsible and the inter-domain links connecting these domains to other adjacent domains.

In another way, the child abstracts each of the domains that its children are responsible as a cloud (or say abstract node) and these clouds are connected by the inter-domain links attached to the domains. The child sends its parent all the inter-domain links attached to any of the domains.

In a third way, the child abstracts all its domains including the domains for which its children are responsible as a cloud. This abstraction is described below in details.

If a parent P1 is also a child of another parent P2, P1 as a child sends its parent P2 a message containing the information about the connections and access points. P1 as a parent has the connections among its children's domains. But these connections are hidden from its parent P2. P1 may have connections from its children's domains to other domains. P1 as a child sends its parent P2 these connections.

P1 as a parent has the access points in its children's domains to be accessible outside of the domains. P1 as child may not send all of these to its parent P2. It sends its parent some of these access points according to some local policies.

From P2's point of view, its child P1 is responsible for one domain, which has some connections to its adjacent domains and some access points to be accessible.

5.2.2. Parent Procedures

5.2.2.1. Process Connections and Accesses

A parent stores into its TED the connections and accesses for each of its children according to the messages containing connections and accesses received. For a message containing Update Connections and Accesses, it updates the connections and accesses in the TED accordingly. For a message containing Withdraw Connections and Accesses, it removes the connections and accesses from the TED.

After receiving the messages for connections and accesses from its children, the parent builds and maintains the TED for the topology of its children's domains, in which each of the domains is seen as a cloud or an abstract node. The information inside each of the domains is hidden from the parent. There are connections among the domains and the access points in the domains to be accessible in the topology.

For a new P2P link from node A to B with no link ID configured, when receiving a message containing the link from a child, the parent stores the link from A into its TED, where A is attached to the child's domain as a cloud. It finds the link's remote end B using the remote IP address of the link. After finding B, it associates the link attached to A with B and the link attached to B with A. This creates a bidirectional connection between A and B.

For a new P2P link from node A to B with link ID configured, when receiving a message containing the link, the parent stores the link from A into its TED. It finds the link's remote end B using the link ID (i.e., B's ID).

For a new broadcast link connecting multiple nodes with no link ID configured, when the parent receives a message containing the link attached to node X, it stores the link from X into its TED. It finds the link's remote end P using the link's local IP address with network mask. P is a Pseudo node identified by the local IP address of the designated node selected from the nodes connected to the link. After finding P, it associates the link attached to X with P and the link connected to P with X. If P is not found, a new Pseudo node P is created. The parent associates the link attached to X with P and the link attached to P with X. This creates a bidirectional connection between X and P.

The first node and second node from which the parent receives a message containing the link is selected as the designed node and backup designed node respectively. After the designed node is down, the backup designed node becomes the designed node and the node other than the designed node with the largest local IP address connecting to the link is selected as the backup designed node.

When the old designed node is down and the backup designed node becomes the new designed node, the parent updates its TED through removing the link between each of nodes X and old P (the Pseudo node corresponding to the old designed node) and adding a link between each of nodes X (still connecting to the broadcast link) and new P (the Pseudo node corresponding to the new designed node).

5.2.2.2. Detail Topology in a Domain

If a parent is in a same organization as its child, it stores into its TED the detail information inside the child’s domain when receiving a message containing the information from the child; otherwise, it discards the information and issues a warning indicating that the information is sent to a wrong place.

6. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

7. IANA Considerations

This section specifies requests for IANA allocation.

8. Acknowledgement

The authors would like to thank Jescia Chen, Adrian Farrel, and Eric Wu for their valuable comments on this draft.

9. References

9.1. Normative References

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

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

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

[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>.

9.2. Informative References

[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>.

[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>.

[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>.

[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>.

Appendix A. Message Encoding

A.1. Extension to Existing Message

An existing Notification message may be extended to advertise the information about connections and access points. The following new Notification-type (NT) and Notification-value (NV) of a NOTIFICATION object in the message are defined:

- o NT=8 (TBD): Connections and Accesses

  * NV=1: Update Connections and Accesses. A NT=8 and NV=1 indicates that the child sends its parent updates on the information about Connections and Accesses, and TLVs containing the information are in the object.

  * NV=2: Withdraw Connections and Accesses. A NT=8 and NV=2 indicates that the child asks its parent to remove Connections and Accesses indicated by TLVs in the object.

A.1.1. TLVs

Four TLVs are defined for connections and accesses. They are Inter-Domain link TLV, Router-ID TLV, Access IPv4/IPv6 Prefix TLV.

The format of the Inter-Domain link TLV is illustrated 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (tTBD1)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Inter-Domain Link Sub-TLVs                  |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

An Inter-Domain link TLV describes a single inter-domain link. It comprises a number of inter-domain link sub-TLVs for the information described in section 4, which are the sub-TLVs defined in RFC 3630 or their equivalents except for the local IP address with mask length defined below.

The format of the Router-ID TLV is shown below. Undefined flags MUST be set to zero. The ID indicates the ID of a router. For a router running OSPF, the ID may be the 32-bit OSPF router ID of the router. For a router running IS-IS, the ID may be the 48-bit IS-IS router ID of the router. For a router not running OSPF or IS-IS, the ID may be the 32-bit ID of the router configured.

```
 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 (tTBD2)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|E|I|      Flags              |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                           32-bit/48-bit ID                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Flag B:     Set to 1 indicating ABR (B is for Border)
Flag E:     Set to 1 indicating ASBR (E is for External)
Flag I:     Set to 1 indicating ID of local router (I is for ID)
```

The format of the Access IPv4/IPv6 Prefix TLV is shown as follows. The cost is the metric to the prefix. The Prefix Length indicates the length of the prefix. The IPv4/IPv6 Prefix indicates an access IPv4/IPv6 address prefix.

```
 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 (tTBD3/tTDB4)      |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               cost                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length | IPv4/IPv6 Prefix (variable)                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A.1.2. Sub-TLVs

The format of the Sub-TLV for a local IPv4/IPv6 address with mask length is shown as follows.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type (stTBD1/stTBD2)     |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv4/IPv6 Address                       ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mask Length  |
+-+-+-+-+-+-+-+-+
```

The IPv4/IPv6 Address indicates the local IPv4/IPv6 address of a link. The Mask Length indicates the length of the IPv4/IPv6 address mask.

A.2. New Message

A new message may be defined to advertise the connections and accesses from a child to its parent. The format of the message containing Connections and Access (AC for short) is as follows:

```
  <AC Message> ::= <Common Header> <NRP>
                   <Connection-List> [<Access-List>]
  where:
    <Connection-List> ::= <Connection> [<Connection-List>]
    <Connection> ::= [<Inter-Domain-Link> | <ABR>]
    <Access-List> ::= <Access-Address> [<Access-List>]
```

Where the value of the Message-Type in the Common Header indicates the new message type. The exact value is to be assigned by IANA. A new RP (NRP) object will be defined, which follows the Common Header.

A new flag W (Withdraw) in the NRP object is defined to indicate whether the connections and access are withdrawn. When flag W is set to one, the parent removes the connections and accesses contained in the message after receiving it. When flag W is set to zero, the parent adds/updates the connections and accesses in the message after receiving it.

An alternative to flag W in the NRP object is a similar flag in each CONNECTION and ACCESS object such as using one bit in Res flags for flag W. For example, when the flag is set to one in the object, the parent removes the connections and accesses in the object after receiving it. When the flag is set to zero in the object, the parent adds/updates the connections and accesses in the object after receiving it.

In another option, one byte in a CONNECTION and ACCESS Object is defined as flags field and one bit is used as flag W. The other undefined bits in the flags field MUST be set to zero.

The objects in the new message are defined below.

A.2.1. CONNECTION and ACCESS Object

A new object, called CONNECTION and ACCESS Object (CA for short), is defined. It has Object-Class ocTBD1. Four Object-Types are defined under CA object:

- o CA Inter-Domain Link: CA Object-Type is 1.
- o CA ABR: CA Object-Type is 2.
- o CA Access IPv4 Prefix: CA Object-Type is 3.
- o CA Access IPv6 Prefix: CA Object-Type is 4.

Each of these objects are described below.

The format of Inter-Domain Link object body is as follows:

```
     Object-Class = ocTBD1 (Connection and Access)
     Object-Type = 1 (CA Inter-Domain Link)
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |W|    Flags    |             Router-ID TLV                     |
 +-+-+-+-+-+-+-+-+                                               +
 ~                                                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      Inter-Domain Link TLVs                   |
 ~                                                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Router-ID TLV indicates an ASBR in the domain, which is a local end of inter-domain links. Each of the Inter-Domain Link TLVs describes an inter-domain link and comprises a number of inter-domain link Sub-TLVs. Flag W=1 indicates withdraw the links. W=0 indicates new or changed links.

The format of ABR object body is illustrated below:

```
     Object-Class = ocTBD1 (Connection and Access)
     Object-Type = 2 (CA ABR)
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |W|    Flags    |             Router-ID TLVs                    |
 +-+-+-+-+-+-+-+-+                                               +
 ~                                                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Each of the Router-ID TLVs indicates an ABR in the domain. Flag W=1 indicates withdraw the ABRs. W=0 indicates new ABRs.

The format of Access IPv4/IPv6 Prefix object body is as follows:

```
     Object-Class = ocTBD1 (Connection and Access)
     Object-Type = 3/4 (CA Access IPv4/IPv6 Prefix)
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |W|    Flags    |           Access IPv4/IPv6 Prefix TLVs        |
 +-+-+-+-+-+-+-+-+                                               +
 ~                                                               ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Each of the Access IPv4/IPv6 Prefix TLVs describes an access IPv4/IPv6 address prefix in the domain, which is accessible to outside of the domain. Flag W=1 indicates withdraw the address prefixes. W=0 indicates new address prefixes.

The TLVs in the objects are the same as those described above.

Authors' Addresses

Huaimo Chen
Futurewei

Boston, MA
USA

EMail: Huaimo.chen@futurewei.com

Mehmet Toy
Verizon

,
USA

EMail: mehmet.toy@verizon.com

Xufeng Liu
Volta Networks

McLean, VA
USA

EMail: xufeng.liu.ietf@gmail.com

Lei Liu
Fujitsu

USA

EMail: liulei.kddi@gmail.com

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

EMail: li_zhenqiang@hotmail.com

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 10, 2021

H. Chen
Futurewei
M. Toy
Verizon
X. Liu
Volta Networks
L. Liu
Fujitsu
Z. Li
China Mobile

January 6, 2021

Hierarchical PCE Determination
draft-chen-pce-h-discovery-08

Abstract

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for determining parent child relations and exchanging the information between a parent and a child PCE in a hierarchical PCE system.

Table of Contents

1. Introduction

A hierarchical PCE architecture is described in RFC 6805, in which a parent PCE has a number of child PCEs. A child PCE may also be a parent PCE, which has multiple child PCEs.

For a parent PCE, it needs to obtain the information about each of its child PCEs. The information about a child PCE comprises the address or ID of the PCE and the domain for which the PCE is responsible. It may also include the position of the PCE, which indicates whether the PCE is a leaf (i.e., only a child) or branch (i.e., a child and also a parent). In addition, the information may indicate whether the child PCE and its responsible domain is in a same organization as the parent PCE.

For a child PCE, it needs to obtain the information about its parent PCE, which includes the address or ID of the parent PCE. The information may also indicate whether the parent PCE is in a same organization as the child PCE.

After a user configures a parent PCE and a child PCE over a session, this parent child PCE relation needs to be determined in the protocol level. This is similar to OSPF and BGP. After an adjacency between two OSPF routers is configured by a user, the OSPF protocol (refer to RFC 2328, Section 7) will determine whether the adjacency is allowed based on the parameters configured, and forms the OSPF adjacency after the determination. After a peer relation between two BGP routers is configured by a user, the BGP protocol (refer to RFC 4271, Section 8) will determine whether the peer is allowed based on the parameters configured, and forms the BGP peer relation after the determination.

For a parent child PCE relation determination, the PCE protocol needs to check or confirm whether the parent child PCE relation is allowed based on the parameters configured. If so, the child PCE has to send its parent PCE the information about it and vice versa.

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for determining parent child relations and exchanging the information between a parent and a child PCE in a hierarchical PCE system.

2. Terminology

The following terminology is used in this document.

Parent Domain: A domain higher up in a domain hierarchy such that it contains other domains (child domains) and potentially other links and nodes.

Child Domain: A domain lower in a domain hierarchy such that it has a parent domain.

Parent PCE: A PCE responsible for selecting a path across a parent domain and any number of child domains by coordinating with child PCEs and examining a topology map that shows domain inter-connectivity.

Child PCE: A PCE responsible for computing the path across one or more specific (child) domains. A child PCE maintains a relationship with at least one parent PCE.

TED: Traffic Engineering Database.

This document uses terminology defined in [RFC5440].

3. Conventions Used in This Document

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

4. Extensions to PCEP

This section describes the extensions to PCEP for determining the relation between a parent PCE and a child PCE and exchanging the information between a parent and a child PCE in a hierarchical PCE system. A child PCE is simply called a child and a parent PCE is called a parent in the following sections.

4.1. Determination of Parent Child Relation

During a PCEP session establishment between two PCEP speakers, each of them advertises its capabilities for Hierarchical PCE (H-PCE for short) through the Open Message with the Open Object containing a new TLV to indicate its capabilities for H-PCE. This new TLV is called H-PCE capability TLV. It 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type = TBD1       |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|P|C|S|B|                 Capability Flags                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Optional  Sub-TLVs                      |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The type of the TLV is TBD1. It has a length of 4 octets plus the size of optional Sub-TLVs. The value of the TLV comprises a capability flags field of 32 bits, which are numbered from the most significant as bit zero. Some of them are defined as follows. The others are not defined and MUST be set to zero.

- o P (Parent - 1 bit): Bit 0 is used as P flag. It is set to 1 indicating a parent.
- o C (Child - 1 bit): Bit 1 is used as C flag. It is set to 1 indicating a child.
- o S (Same Org - 1 bit): Bit 2 is used as S flag. It is set to 1 indicating a PCE in a same organization as its remote peer.

- B (Both - 1 bit): Bit 3 is used as B flag. It is set to 1 indicating a PCE as both a child and a parent.

The following Sub-TLVs are defined:

- A Domain Sub-TLV containing an AS number and optional area, and

- PCE-ID Sub-TLV containing the ID of a PCE.

4.2. Sub-TLVs

When a child sends its parent a Open message, it places the information about it in the message through using some optional Sub-TLVs. When a parent sends each of its child PCEs a Open message, it puts the information about it in the message.

4.2.1. Domain Sub-TLV

A domain is an AS or an area in an AS. An AS is identified by an AS number. An area in an AS is identified by the combination of the AS and the area. The former is indicated by an AS number and the latter by an area number. A domain is represented by a domain Sub-TLV containing an AS number and a optional area number.

The format of the domain Sub-TLV is shown 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (tTBD1)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       AS Number (4 bytes)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                      Optional  Area Number                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where Length is four plus size of area number.

An AS is represented by a domain Sub-TLV containing only the AS number of the AS. In this case, the Length is four. An area in an AS is represented by a domain Sub-TLV containing the AS number of the AS and the area number of the area. In this case, the Length is eight.

4.2.2. PCE ID Sub-TLV

An Identifier (ID) of a PCE (PCE ID for short) is a 32-bit number that uniquely identifies the PCE among all PCEs. This 32-bit number for PCE ID SHOULD NOT be zero.

The format of the PCE ID Sub-TLV is shown 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (tTBD3)          |          Length (4)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       PCE ID (4 bytes)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The PCE ID Sub-TLV specifies an non zero number as the identifier of the PCE.

Alternatively, an IP address attached to a PCE can also be used as an identifier of the PCE. The format of an IPv4 address Sub-TLV is shown 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (tTBD4)          |          Length (4)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv4 Address (4 bytes)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv4 address Sub-TLV specifies an IPv4 address associated with the PCE, which is used as the identifier of the PCE.

The format of an IPv6 address Sub-TLV is shown 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Type (tTBD5)          |         Length (16)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IPv6 Address (16 bytes)                    |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv6 Sub-TLV specifies an IPv6 address associated with the PCE, which is used as the identifier of the PCE.

4.3. Procedures

For two PCEs A and B configured as parent and child, they determine parent child relation through Open messages in the initialization phase. The following is a sequence of events related.

```
          A                                       B
     Configure B                             Configure A
     as its child                            as its parent
                     Open (P=1, A's ID)
                    -------------------> Same as configured
                                           A is B's parent
                     Open (C=1, B's ID)
 Same as configured <--------------------
   B is A's child
```

A sends B a Open message with P=1 and A's ID after B is configured as its child on it. B sends A a Open message with C=1 and B's ID after A is configured as its parent on it.

When A receives the Open message from B and determines C=1 and the PCE ID of B in the message is the same as the PCE ID of the child locally configured, B is A's child.

When B receives the Open message from A and determines P=1 and the PCE ID of A in the message is the same as the PCE ID of the parent locally configured, A is B's parent.

The Open message from child B to its parent A contains B's domain, which is represented by a domain Sub-TLV in the H-PCE capability TLV. If child B is also a parent, the B flag in the TLV is set to 1.

The PCE ID in a Open message may be represented in one of the following ways:

- o The source IP address of the message (i.e., PCE session).
- o A PCE ID Sub-TLV in the H-PCE capability TLV.
- o An IP address Sub-TLV in the H-PCE capability TLV.

When the IP address Sub-TLV is used, the address in the Sub-TLV MUST be the same as the source IP address of the PCE session.

For a child that is a leaf, it is normally responsible for one domain, which is contained in the message to its parent.

For a child that is a branch (i.e., also a parent of multiple child PCEs), it may be directly responsible for one domain, which is contained in the message to its parent. In addition, it is responsible for the domains of its child PCEs. In other words, it is responsible for computing paths crossing the domains through working together with its child PCEs. If these domains are all areas of an AS, the AS is included in the message to its parent.

A parent stores the information about each of its child PCEs received. When the session to one of them is down, it removes the information about the child on that session.

A child stores the information about its parent received. When the session to the parent is down, it removes the information about the parent.

If there already exists a session between A and B and the configurations on parent and child are issued on them, the procedures above may be executed through bringing down the existing session and establishing a new session between them. Alternatively, they may determine parent child relation through using extended Notification messages in the same procedures as using Open messages described above without bringing down the existing session.

The following new Notification-type and Notification-value are defined for H-PCE:

- Notification-type=5 (TBD): Determination of H-PCE

  * Notification-value=1: The information about a parent PCE or a child PCE. A Notification-type=5, Notification-value=1 indicates that the PCE sends its peer the information about it and a TLV containing the information is in the Notification object. The format and contents of the TLV is the same as the H-PCE capability TLV described above. The only difference may be the type of the TLV.

5. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

6. IANA Considerations

This section specifies requests for IANA allocation.

7. Acknowledgement

The authors would like to Jescia Chen, Adrian Farrel for their valuable comments on this draft.

8. References

8.1. Normative References

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

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

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

8.2. Informative References

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

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

Authors' Addresses

Huaimo Chen
Futurewei
Boston, MA
USA

EMail: Huaimo.chen@futurewei.com

Mehmet Toy
Verizon

USA

EMail: mehmet.toy@verizon.com

Xufeng Liu
Volta Networks

McLean, VA
USA

EMail: xufeng.liu.ietf@gmail.com

Lei Liu
Fujitsu

USA

EMail: liulei.kddi@gmail.com

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

EMail: li_zhenqiang@hotmail.com

PCE Working Group H. Chen
Internet-Draft Futurewei
Intended status: Standards Track M. Toy
Expires: July 10, 2021 Verizon
X. Liu
Volta Networks
L. Liu
Fujitsu
Z. Li
China Mobile
January 6, 2021

Static PCEP Link State
draft-chen-pce-pcc-ted-08

Abstract

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for a PCC to advertise the information about the links without running IGP and for a PCE to build a TED based on the information received.

Table of Contents

1. Introduction

A PCE architecture is described in RFC 4655, in which a Traffic Engineering Database (TED) for a PCE is constructed based on the link information from IGP (OSPF or IS-IS) running in the domain for which the PCE is responsible.

For a domain without running IGP, the PCE responsible for the domain may obtain the link information from a PCC running on each node in the domain.

This document presents extensions to the Path Computation Element Communication Protocol (PCEP) for a PCC to advertise the information about the links attached to the node running the PCC and for a PCE to build the TED based on the information received from the PCC.

2. Terminology

ABR: Area Border Router. Router used to connect two IGP areas (Areas in OSPF or levels in IS-IS).

ASBR: Autonomous System (AS) Border Router. Router used to connect together ASes via inter-AS links.

This document uses terminology defined in [RFC5440].

3. Conventions Used in This Document

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

4. Link Information

Since no IGP runs over any link, we may not obtain any link information via IGP. But links are configured.

For a point-to-point (P2P) link between nodes A and B, from A’s point of view, we have the following link information:

1) Link Type: P2P
2) Local IP address
3) Remote IP address
4) Traffic engineering metric
5) Maximum bandwidth
6) Maximum reservable bandwidth
7) Unreserved bandwidth
8) Administrative group
9) SRLG

A link ID for the link is obtained if a user configures it; otherwise, no link ID (i.e., the Router ID of A’s neighbor) may be obtained since no IGP adjacency over the link is formed.

For a broadcast link connecting multiple nodes, on each of the nodes X, we have the same link information as above except for:

a) Link Type: Multi-access,
b) Local IP address with mask length, and
c) No Remote IP address.

In other words, the information about the broadcast link obtained by node X comprises a), b), 4) to 9), but does not include any remote IP address or link ID. A link ID for the link is obtained if a user configures it; otherwise, no link ID (i.e., the interface address of the designated router for the link) may be obtained since no IGP selects it.

A PCE constructs a TED for its responsible domain after receiving the link information from the PCC running on every node in the domain.

5. Extensions to PCEP

5.1. Extension to Existing Message

An existing Notification message may be extended to advertise the information about links. Alternatively, a new message can be used (refer to Appendix A).

The following new Notification-type (NT) and Notification-value (NV) of a NOTIFICATION object in a Notification message are defined:

- o NT=8 (TBD): Links
  - * NV=1: Update Links. NT=8 and NV=1 indicates that the PCC requests the PCE to update the link information based on the TLVs in the object, which are described below.
  - * NV=2: Withdraw Links. NT=8 and NV=2 indicates that the PCC asks the PCE to remove the Links indicated by the TLVs in the object.

5.1.1. TLVs

A link TLV and a Router-ID TLV are defined. The format of the link TLV is illustrated below. The Type=tTBD1 indicates a link TLV Type. The Length indicates the size of the Link Sub-TLVs.

```
 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 (tTBD1)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Sub-TLVs                         |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A link TLV describes a single link. It comprises a number of link sub-TLVs for the information described in section 4, which are the sub-TLVs defined in RFC 3630 or their equivalents except for the local IP address with mask length defined below.

The format of the Router-ID TLV is shown below. The Type=tTBD2 indicates a Router-ID TLV Type. The Length indicates the size of the ID and flags field.

```
 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 (tTBD2)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|B|E|I|      Flags              |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
|                          32-bit/48-bit ID                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Flag B:     Set to 1 indicating ABR (B is for Border)
Flag E:     Set to 1 indicating ASBR (E is for External)
Flag I:     Set to 1 indicating ID of local router (I is for ID)
```

Undefined flags MUST be set to zero. The ID indicates the ID of a router. For a router not running IGP, the ID may be the 32-bit or 48-bit ID of the router configured.

5.1.2. Sub-TLVs

The format of the Sub-TLV for a local IPv4 address with mask length is shown below. The Type=stTBD1 indicates a local IPv4 Address with mask length. The Length indicates the size of the IPv4 address and Mask Length. The IPv4 Address indicates the local IPv4 address of a link. The Mask Length indicates the length of the IPv4 address mask.

```
 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 (stTBD1)         |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          IPv4 Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Mask Length |
+-+-+-+-+-+-+-+-+
```

The format of the Sub-TLV for a local IPv6 address with mask length is illustrated below. The Type=stTBD2 indicates a local IPv6 Address with mask length. The Length indicates the size of the IPv6 address and Mask Length. The IPv6 Address indicates the local IPv6 address of a link. The Mask Length indicates the length of the IPv6 address mask.

```
 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 (stTBD2)         |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        IPv6 Address (16 bytes)                |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Mask Length  |
+-+-+-+-+-+-+-+-+
```

5.2. Procedures

5.2.1. PCC Procedures

1. New or Changed Links

After the session between a PCC and a PCE is established, the PCC sends the PCE a message containing the information about the links attached to the node running the PCC.

For any new or changed links, the PCC sends the PCE a message containing the information about these links with indication of Update Links.

For example, for a new P2P link from node A, the PCC running on A sends the PCE a Notification message having a NOTIFICATION object with NT=8 and NV=1 (indicating Update Links), which contains a Router-ID TLV, followed by a link TLV. The former comprises A's ID and flag I set to 1. The latter comprises the Sub-TLVs for the information described in section 4.

For multiple new or changed links from node A, the PCC running on A sends the PCE a Notification message having a NOTIFICATION object with NT=8 and NV=1, which contains a Router-ID TLV for A's ID, followed by multiple link TLVs for the links.

2. Links Down

For links down, the PCC sends the PCE a message containing the information about these links with indication of Withdraw Links.

For example, for multiple links from node A down, the PCC running on A sends the PCE a Notification message having a NOTIFICATION object with NT=8 and NV=2 (indicating Withdraw Links), which contains a Router-ID TLV for A's ID, followed by multiple link TLVs for the links. The TLV for a P2P link comprises the Sub-TLVs for the information on 1), 2) and 3) described in section 4. The TLV for a broadcast link comprises the Sub-TLVs for the information on a) and b) described in section 4.

3. Simplified Message

Alternatively, the messages may be simplified. For each node, the source IP address of the PCC running on the node may be used as the ID of the node. The PCE knows the address after the session between the PCE and the PCC is up. Thus, a message containing the information about links does not need include any router-ID TLV.

For example, for a new P2P link attached to node A, the PCC running on A sends the PCE a Notification message having a NOTIFICATION object with NT=8 and NV=1 (indicating Update Links), which contains a link TLV comprising the Sub-TLVs for the information on 1) to 9) described in section 4. The object does not contain any Router-ID TLV for node A.

5.2.2. PCE Procedures

A PCE stores into its TED the links for each node according to the messages for the links received from the PCC running on the node. For a message containing Update Links, it updates the links accordingly. For a message containing Withdraw Links, it removes the links. When a node is down, the PCE removes the links attached to the node.

For a new P2P link between node A and B with no link ID configured, when receiving a message containing the link from the PCC running on A, the PCE stores the link for A (i.e., the link from A) into its TED. It will find the link's remote end B using the remote IP address of the link. After finding B, it associates the link for A with B and the link for B with A. This creates a bidirectional connection between A and B.

For a new broadcast link connecting multiple nodes with no link ID configured, when receiving a message containing the link from the PCC running on each of the nodes X, the PCE stores the link for X (i.e., the link from X) into its TED. It will find the link's remote end P using the link's local IP address with network mask. P is a Pseudo node identified by the local IP address of the designated node selected from the nodes connected to the link. After finding P, it associates the link for X with P and the link for P with X. This creates a bidirectional connection between X and P.

The first node and second node from which the PCE receives a message containing the link is selected as the designed node and backup designed node respectively. After the designed node is down, the backup designed node becomes the designed node and the node other than the designed node with the largest local IP address connecting to the link is selected as the backup designed node.

When the old designed node is down and the backup designed node becomes the new designed node, the PCE updates its TED through removing the link between each of nodes X and old P (the Pseudo node corresponding to the old designed node) and adding a link between each of nodes X (still connecting to the broadcast link) and new P (the Pseudo node corresponding to the new designed node).

6. Security Considerations

The mechanism described in this document does not raise any new security issues for the PCEP protocols.

7. IANA Considerations

This section specifies requests for IANA allocation.

8. Acknowledgement

The authors would like to thank Jescia Chen, and Eric Wu for their valuable comments on this draft.

9. References

9.1. Normative References

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

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

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

[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>.

9.2. Informative References

[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>.

Appendix A. New Message

A new message may be defined to advertise the information on links. The format of the message for the information on Links (IL for short) is as follows:

```
  <IL Message> ::= <Common Header> <NRP> <Link-List>
  where:
    <Link-List> ::= <LINK> [<Link-List>]
```

Where the value of the Message-Type in the Common Header indicates the new message type. The exact value is to be assigned by IANA. A new RP (NRP) object will be defined, which follows the Common Header.

A new flag W (Withdraw) in the NRP object is defined to indicate whether the links are withdrawn. When flag W is set to one, the PCE removes the links in the message after receiving it from the PCC. When flag W is set to zero, the PCE adds/updates the links in the message.

An alternative to flag W in the NRP object is a similar flag W in each LINK object. For example, when the flag is set to one in the LINK object, the PCE removes the links in the object. When the flag is set to zero, the PCE adds/updates the links in the object.

The format of a LINK object body is as follows:

```
     Object-Class = ocTBD1 (LINK)   Object-Type = 1 (Link)
   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
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |W|    Flags    |               (Router-ID TLV)                 |
  +-+-+-+-+-+-+-+-+                                               +
  ~                                                               ~
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                            Link TLVs                          |
  ~                                                               ~
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flag W=1 indicates Withdraw links. W=0 indicates Updated links. Router-ID TLV is optional. Link TLVs are mandatory. They are the same as described in section 5.

Authors' Addresses

Huaimo Chen
Futurewei

Boston, MA
USA

EMail: Huaimo.chen@futurewei.com

Mehmet Toy
Verizon

USA

EMail: mehmet.toy@verizon.com

Xufeng Liu
Volta Networks

McLean, VA
USA

EMail: xufeng.liu.ietf@gmail.com

Lei Liu
Fujitsu

USA

EMail: liulei.kddi@gmail.com

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

EMail: li_zhenqiang@hotmail.com

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

D. Dhody
Y. Lee
Huawei Technologies
D. Ceccarelli
Ericsson
March 13, 2017

Applicability of Path Computation Element (PCE) for Abstraction and Control of TE Networks (ACTN)

draft-dhody-pce-applicability-actn-02

Abstract

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

The Path Computation Element Communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

This document examines the applicability of PCE to the ACTN framework.

Status of This Memo

Table of Contents

1. Introduction

1.1. Path Computation Element (PCE)

The Path Computation Element communication Protocol (PCEP) [RFC5440] provides mechanisms for Path Computation Elements (PCEs) [RFC4655] to perform path computations in response to Path Computation Clients (PCCs) requests.

The ability to compute shortest constrained TE LSPs in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks across multiple domains has been identified as a key motivation for PCE development.

A stateful PCE is capable of considering, for the purposes of path computation, not only the network state in terms of links and nodes (referred to as the Traffic Engineering Database or TED) but also the status of active services (previously computed paths, and currently reserved resources, stored in the Label Switched Paths Database (LSPDB).

[RFC8051] describes general considerations for a stateful PCE deployment and examines its applicability and benefits, as well as its challenges and limitations through a number of use cases.

[I-D.ietf-pce-stateful-pce] describes a set of extensions to PCEP to provide stateful control. A stateful PCE has access to not only the information carried by the network's Interior Gateway Protocol (IGP), but also the set of active paths and their reserved resources for its computations. The additional state allows the PCE to compute constrained paths while considering individual LSPs and their interactions. [I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model.

[I-D.ietf-pce-stateful-pce] also describes the active stateful PCE. The active PCE functionality allows a PCE to reroute an existing LSP or make changes to the attributes of an existing LSP, or a PCC to delegate control of specific LSPs to a new PCE.

1.1.1. Role of PCE in SDN

Software-Defined Networking (SDN) refers to a separation between the control elements and the forwarding components so that software running in a centralized system called a controller, can act to program the devices in the network to behave in specific ways. A required element in an SDN architecture is a component that plans how the network resources will be used and how the devices will be programmed. It is possible to view this component as performing specific computations to place flows within the network given knowledge of the availability of network resources, how other forwarding devices are programmed, and the way that other flows are routed. It is concluded in [RFC7399], that this is the same function that a PCE might offer in a network operated using a dynamic control plane. This is the function and purpose of a PCE, and the way that a PCE integrates into a wider network control system including SDN is presented in Application-Based Network Operation (ABNO) [RFC7491].

1.1.2. PCE in multi-domain and multi-layer deployments

Computing paths across large multi-domain environments require special computational components and cooperation between entities in different domains capable of complex path computation. The PCE provides an architecture and a set of functional components to address this problem space. A PCE may be used to compute end-to-end paths across multi-domain environments using a per-domain path computation technique [RFC5152]. The Backward recursive PCE based path computation (BRPC) mechanism [RFC5441] defines a PCE-based path computation procedure to compute inter-domain constrained MPLS and GMPLS TE networks. However, both per-domain and BRPC techniques assume that the sequence of domains to be crossed from source to destination is known, either fixed by the network operator or obtained by other means.

[RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs) when the domain sequence is not known. Within the Hierarchical PCE (H-PCE) architecture, the Parent PCE (P-PCE) is used to compute a multi-domain path based on the domain connectivity information. A Child PCE (C-PCE) may be responsible for a single domain or multiple domains, it is used to compute the intra-domain path based on its domain topology information.

[I-D.dhodylee-pce-stateful-hpce] state the considerations for stateful PCE(s) in hierarchical PCE architecture. In particular, the behavior changes and additions to the existing stateful PCE mechanisms (including PCE- initiated LSP setup and active PCE usage) in the context of networks using the H-PCE architecture.

[RFC5623] describes a framework for applying the PCE-based architecture to inter-layer to (G)MPLS TE. It provides suggestions for the deployment of PCE in support of multi-layer networks. It also describes the relationship between PCE and a functional component in charge of the control and management of the VNT, called the Virtual Network Topology Manager (VNTM).

1.2. Abstraction and Control of TE Networks (ACTN)

[I-D.ietf-teas-actn-requirements] describes the high-level ACTN requirements. [I-D.ietf-teas-actn-framework] describes the architecture model for ACTN including the entities (Customer Network Controller(CNC), Mult-domain Service Coordinator(MDSC), and Physical Network Controller(PNC)) and their interfaces.

The ACTN reference architecture identified a three-tier control hierarchy as depicted in Figure 1:

```
      +-------+                 +-------+                 +-------+
      | CNC-A |                 | CNC-B |                 | CNC-C |
      +-------+                 +-------+                 +-------+
            \                       |                        /
             ----------             |-CMI I/F     -----------
                        \           |            /
                         +-----------------------+
                         |         MDSC          |
                         +-----------------------+
                         /          |            \
               ----------           |-MMI I/F     -----------
              /                     |                        \
      +----------+              +----------+             +--------+
      |   MDSC   |              |   MDSC   |             |  MDSC  |
      +----------+              +----------+             +--------+
           |                    /     |-MPI I/F            /    \
           |                   /      |                   /      \
        +-----+           +-----+  +-----+           +-----+  +-----+
        | PNC |           | PNC |  | PNC |           | PNC |  | PNC |
        +-----+           +-----+  +-----+           +-----+  +-----+
```

CMI - (CNC-MDSC Interface)
MMI - (MDSC-MDSC Interface)
MPI - (MDSC-PNC Interface)

Figure 1: ACTN Hierarchy

The two interfaces with respect to the MDSC, one north of the MDSC Interface) and MPI (MDSC-PNC Interface), respectively. MMI (MDSC-MDSC interface) is used to support recursion.

[I-D.ietf-teas-actn-info-model] provides an information model for ACTN interfaces.

1.3. PCE and ACTN

This document examines the PCE and ACTN architecture and describes how the PCE architecture is applicable to ACTN. It also lists the PCEP extensions that are needed to use PCEP as an ACTN interface. This document also identifies any gaps in PCEP, that exist at the time of publication of this document.

2. Architectural Considerations

ACTN [I-D.ietf-teas-actn-framework] architecture is based on hierarchy and recursiveness of controllers. It defines three types of controllers (depending on the functionalities they implement). The main functionalities are -

- o Multi domain coordination function
- o Virtualization/Abstraction function
- o Customer mapping/translation function
- o Virtual service coordination function

Section 3 of [I-D.ietf-teas-actn-framework] describes these functions.

It should be noted that, in this document we list all possible ways in which PCEP could be used for each of the above functions, but all functions are not required to be implemented via PCEP. Operator may choose to use the PCEP for multi domain coordination via stateful H-PCE but use RestConf or BGP-LS to get the topology and support virtualization/abstraction function.

2.1. Multi domain coordination via Hierarchy

With the definition of domain being "everything that is under the control of the single logical controller", as per [I-D.ietf-teas-actn-framework], 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.

The MDSC in ACTN framework realizes this function by coordinating the per-domain PNCs in a hierarchy of controllers. It also needs to detach from the underlying network technology and express customer concerns by business needs.

[RFC6805] and [I-D.dhodylee-pce-stateful-hpce] describes a hierarchy of PCE with Parent PCE coordinating multi-domain path computation function between Child PCE(s). It is easy to see how these principles align, and thus how stateful H-PCE architecture can be used to realize ACTN.

The Per domain stitched LSP in the Hierarchical stateful PCE architecture, described in Section 3.3.1 of [I-D.dhodylee-pce-stateful-hpce] is well suited for multi-domain coordination function. This includes domain sequence selection; E2E path computation; Controller (PCE) initiated path setup and reporting. This is also applicable to multi-layer coordination in case of IP+optical networks.

[I-D.litkowski-pce-state-sync]" describes the procedures to allow a stateful communication between PCEs for various use-cases. The procedures and extensions are also applicable to Child and Parent PCE communication and thus useful for ACTN as well.

2.2. Virtualization/Abstraction function

To realize ACTN, an abstracted view of the underlying network resources needs to be built. This includes global network-wide abstracted topology based on the underlying network resources of each domain. This also include abstract topology created as per the customer service connectivity requests and represented as a network slice allocated to each customer.

In order to compute and provide optimal paths, PCEs require an accurate and timely Traffic Engineering Database (TED). Traditionally this TED has been obtained from a link state (LS) routing protocol supporting traffic engineering extensions. PCE may construct its TED by participating in the IGP ([RFC3630] and [RFC5305] for MPLS-TE; [RFC4203] and [RFC5307] for GMPLS). An alternative is offered by BGP-LS [RFC7752].

In case of H-PCE [RFC6805], the parent PCE needs to build the domain topology map of the child domains and their interconnectivity. [RFC6805] and [I-D.ietf-pce-inter-area-as-applicability] suggest that BGP-LS could be used as a "northbound" TE advertisement from the child PCE to the parent PCE.

[I-D.dhodylee-pce-pcep-ls] proposes another approaches for learning and maintaining the Link-State and TE information as an alternative to IGPs and BGP flooding, using PCEP itself. The child PCE can use this mechanism to transport Link-State and TE information from child PCE to a Parent PCE using PCEP.

In ACTN, there is a need to control the level of abstraction based on the deployment scenario and business relationship between the controllers. The mechanism used to disseminate information from PNC (child PCE) to MDSC (parent PCE) should support abstraction. [I-D.lee-teas-actn-abstraction] describes a few alternative approaches of abstraction. The resulting abstracted topology can be encoded using the PCEP-LS mechanisms [I-D.dhodylee-pce-pcep-ls]. PCEP-LS is an attractive option when the operator would wish to have a single control plane protocol (PCEP) to achieve ACTN functions.

2.3. Customer mapping function

In ACTN, there is a need to map customer virtual network (VN) requirements into network provisioning request to the PNC. That is, the customer requests/commands are mapped into network provisioning requests that can be sent to the PNC. Specifically, it provides mapping and translation of a customer's service request into a set of parameters that are specific to a network type and technology such that network configuration process is made possible.

[I-D.ietf-pce-pce-initiated-lsp] describes the setup, maintenance and teardown of PCE-initiated LSPs under the stateful PCE model, without the need for local configuration on the PCC, thus allowing for a dynamic network that is centrally controlled and deployed. To instantiate or delete an LSP, the PCE sends the Path Computation LSP Initiate Request (PCInitiate) message to the PCC. As described in [I-D.dhodylee-pce-stateful-hpce], for inter-domain LSP in Hierarchical PCE architecture, the initiation operations can be carried out at the parent PCE. In which case after parent PCE finishes the E2E path computation, it can send the PCInitiate message to the child PCE, the child PCE further propagates the initiate request to the LSR. The customer request is received by the MDSC (parent PCE) and based on the business logic, global abstracted topology, network conditions and local policy, the MDSC (parent PCE) translates this into per domain LSP initiation request that a PNC (child PCE) can understand and act on. This can be done via the PCInitiate message.

PCEP extensions for associating opaque policy between PCEP peer [I-D.ietf-pce-association-policy] can be used.

2.4. Virtual Network Operations

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

[I-D.leedhody-pce-vn-association] describes the need for associating a set of LSPs with a VN "construct" to facilitate VN operations in PCE architecture. This association allows the PCEs to identify which LSPs belong to a certain VN.

This association based on VN is useful for various optimizations at the VN level which can be applied to all the LSPs that are part of the VN slice. During path computation, the impact of a path for an LSP is compared against the paths of other LSPs in the VN. This is to make sure that the overall optimization and SLA of the VN rather than of a single LSP. Similarly, during re-optimization, advanced path computation algorithm and optimization technique can be considered for all the LSPs belonging to a VN/customer and optimize them all together.

3. Interface Considerations

As per [I-D.ietf-teas-actn-framework], 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. The 3 ACTN interfaces are -

- o The CNC-MDSC Interface (CMI) is an interface between a Customer Network Controller and a Multi Domain Service Coordinator. It requests the creation of the network resources, topology or services for the applications. The MDSC may also report potential network topology availability if queried for current capability from the Customer Network Controller.

- o 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.

- o The MDSC-MDSC Interface (MMI) is a special case of the MPI and behaves similarly to an MPI to support general functions performed by the MDSCs such as abstraction function and provisioning function. From an abstraction point of view, the top level MDSC which interfaces the CNC operates on a higher level of abstraction (i.e., less granular level) than the lower level MSDCs. As such, the MMI carries more abstract TE information than the MPI.

PCEP is especially suitable on the MPI and MMI as it meets the requirement and the functions as set out in the ACTN framework [I-D.ietf-teas-actn-framework]. Its recursive nature is well suited via the multi-level hierarchy of PCE. The Section 4 describe how PCE and PCEP could help realize ACTN.

4. Realizining ACTN with PCE (and PCEP)

As per the example in the Figure 2, there are 4 domains, each with its own PNC and a MDSC at top. The PNC and MDSC need PCE as a important function. The PNC (or child PCE) already uses PCEP to communicate to the network device. It can utilize the PCEP as the MPI to communicate between controllers too.

```
                          ******
                ..........*MDSC*..............................
             .            ****** ..                   MPI    .
          .                .        .                        .
       .                   .          .                      .
     .                    .             .                    .
    .                    .                .                  .
   .                    .                  .                 .
  .                    .                    .                .
  v                    v                    v                .
******               ******               ******             .
*PNC1*               *PNC2*               *PNC4*             .
******               ******               ******             .
+---------------+    +---------------+    +---------------+  .
|A              |----|               |----|              C|  .
|               |    |               |    |               |  .
|DOMAIN 1       |----|DOMAIN 2       |----|DOMAIN 4       |  .
+------------B13+    +---------------+    +B43------------+  .
                \                         /                  .
                 \   ******              /                   .
                  \  *PNC3*<............/.....................
                   \ ******            /
                    \+---------------+/
                     B31           B34
                     |               |
                     |DOMAIN 3      B|
                     +---------------+

MDSC -> Parent PCE
PNC  -> Child  PCE
MPI  -> PCEP
```

Figure 2: ACTN with PCE

o Building Domain Topology at MDSC: PNC (or child PCE) needs to have the TED to compute path in its domain. As described in Section 2.2, it can learn the topology via IGP or BGP-LS. PCEP-LS is also a proposed mechanism to carry link state and traffic engineering information within PCEP. A mechanism to carry abstracted topology while hiding technology specific information between PNC and MDSC is described in [I-D.dhodylee-pce-pcep-ls]. At the end of this step the MDSC (or parent PCE) has the abstracted topology from each of its PNC (or child PCE). This could be as simple as a domain topology map as described in

[RFC6805] or it can have full topology information of all domains. The latter is not scalable and thus an abstracted topology of each domain interconnected by inter-domain links is the most common case.

* Topology Change: When the PNC learns of any topology change, the PNC needs to decide if the change needs to be notified to the MDSC. This is dependent on the level of abstraction between the MDSC and the PNC.

o VN Instantiate: MDSC is requested to instantiate a VN, the minimal information that is required would be a VN identifier and a set of end points. Various path computation, setup constraints and objective functions may also be provided. In PCE terms, a VN Instantiate can be considered as a set of paths belonging to the same VN. As described in Section 2.4 and [I-D.leedhody-pce-vn-association] the VN association can help in identifying the set of paths that belong to a VN. The rest of the information like the endpoints, constraints and objective function is already defined in PCEP in terms of a single path.

* Path Computation: As per the example in the Figure 2, the VN instantiate requires two end to end paths between (A in Domain 1 to B in Domain 3) and (A in Domain 1 to C in Domain 4). The MDSC (or parent PCE) triggers the end to end path computation for these two paths. MDSC can do path computation based on the abstracted domain topology that it already has or it may use the H-PCE procedures (Section 2.1) using the PCReq and PCRep messages to get the end to end path with the help of PNC. Either way, the resulted E2E paths may be broken into per-domain paths.

* A-B: (A-B13,B13-B31,B31-B)

* A-C: (A-B13,B13-B31,B34-B43,B43-C)

* Per Domain Path Instantiation: Based on the above path computation, MDSC can issue the path instantiation request to each PNC via PCInitiate message (see [I-D.dhodylee-pce-stateful-hpce] and [I-D.leedhody-pce-vn-association]). A suitable stitching mechanism would be use to stitch these per domain LSPs.

* Per Domain Path Report: Each PNC should report the status of the per-domain LSP to the MDSC via PCRpt message, as per the Hierarchy of stateful PCE ([I-D.dhodylee-pce-stateful-hpce]). The status of the end to end LSP (A-B and A-C) is made up when all the per domain LSP are reported up by the PNCs.

  * Delegation: It is suggested that the per domain LSPs are delegated to respective PNC, so that they can control the path and attributes based on each domain network conditions.

  * State Synchronization: The state needs to be synchronized between the parent PCE and child PCE. The mechanism described in [I-D.litkowski-pce-state-sync] can be used.

- VN Modify: MDSC is requested to modify a VN, for example the bandwidth for VN is increased. This may trigger path computation at MDSC as described in the previous step and can trigger an update to existing per-intra-domain path (via PCUpd message) or creation (or deletion) of a per-domain path (via PCInitiate message). This should be done in make-before-break fashion.

- VN Delete: MDSC is requested to delete a VN, in this case, based on the E2E paths and the resulting per-domain paths need to be removed (via PCInitiate message).

- VN Update (based on network changes): Any change in the per-domain LSP are reported to the MDSC (via PCRpt message) as per [I-D.dhodylee-pce-stateful-hpce]. This may result in changes in the E2E path or VN status. This may also trigger a re-optimization leading to a new per-domain path, update to existing path, or deletion of the path.

- VN Protection: The VN protection/restoration requirements, need to applied to each E2E path as well as each per domain path. The MDSC needs to play a crucial role in coordinating the right protection/restoration policy across each PNC. The existing protection/restoration mechanism of PCEP can be applied on each path.

- In case PNC generates an abstract topology to the MDSC, the PCInitiate/PCUpd messages from the MDSC to a PNC will contain a path with abstract nodes and links. PNC would need to take that as an input for path computation to get a path with physical nodes and links. Similarly PNC would convert the path received from the device (with physical nodes and links) into abstract path (based on the abstract topology generated before with abstract nodes and links) and reported to the MDSC.

5. Relationship to PCE based central control

[I-D.ietf-teas-pce-central-control] introduces the architecture for PCE as a central controller (PCECC), it further examines the motivations and applicability for PCEP as a southbound interface, and introduces the implications for the protocol. The section 2.1.3 of

[I-D.ietf-teas-pce-central-control] describe an hierarchy of PCE-based controller as per the Hierarchy of PCE framework defined in [RFC6805]. Both ACTN and PCECC is based on the same basic framework and thus compatible with each other.

6. IANA Considerations

This is an informational document and thus does not have any IANA allocations to be made.

7. Security Considerations

The ACTN framework described in [I-D.ietf-teas-actn-framework] defines key components and interfaces for managed traffic engineered networks. It also list various security considerations such as request and control of resources, confidentially of the information, and availability of function which should be taken into consideration.

When PCEP is used on the MPI/MMI, this interface needs to be secured, use of [I-D.ietf-pce-pceps] is RECOMENDED. Each PCEP extension listed in this document, presents its individual security considerations, which continue to apply.

8. Acknowledgments

The authors would like to thank Jonathan Hardwick for the inspiration behind this document. Further thanks to Avantika for her comments with suggested text.

9. References

9.1. Normative References

[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>.

9.2. Informative References

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

[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, <http://www.rfc-editor.org/info/rfc4203>.

[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>.

[RFC5152] Vasseur, JP., Ed., Ayyangar, A., Ed., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, DOI 10.17487/RFC5152, February 2008, <http://www.rfc-editor.org/info/rfc5152>.

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

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

[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>.

[RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, DOI 10.17487/RFC5623, September 2009, <http://www.rfc-editor.org/info/rfc5623>.

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <http://www.rfc-editor.org/info/rfc6805>.

[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>.

[RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <http://www.rfc-editor.org/info/rfc7491>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <http://www.rfc-editor.org/info/rfc7752>.

[RFC8051] Zhang, X., Ed. and I. Minei, Ed., "Applicability of a Stateful Path Computation Element (PCE)", RFC 8051, DOI 10.17487/RFC8051, January 2017, <http://www.rfc-editor.org/info/rfc8051>.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-18 (work in progress), December 2016.

[I-D.ietf-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-09 (work in progress), March 2017.

[I-D.dhodylee-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., King, D., and O. Dios, "Hierarchical Stateful Path Computation Element (PCE).", draft-dhodylee-pce-stateful-hpce-03 (work in progress), March 2017.

[I-D.ietf-teas-pce-central-control]
Farrel, A., Zhao, Q., Li, Z., and C. Zhou, "An Architecture for Use of PCE and PCEP in a Network with Central Control", draft-ietf-teas-pce-central-control-01 (work in progress), December 2016.

[I-D.ietf-teas-actn-requirements]
Lee, Y., Dhody, D., Belotti, S., Pithewan, K., and D. Ceccarelli, "Requirements for Abstraction and Control of TE Networks", draft-ietf-teas-actn-requirements-04 (work in progress), January 2017.

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-04 (work in progress), February 2017.

[I-D.ietf-teas-actn-info-model]
Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", draft-ietf-teas-actn-info-model-00 (work in progress), February 2017.

[I-D.ietf-pce-inter-area-as-applicability]
King, D., Meuric, J., Dugeon, O., Zhao, Q., Dhody, D., and O. Dios, "Applicability of the Path Computation Element to Inter-Area and Inter-AS MPLS and GMPLS Traffic Engineering", draft-ietf-pce-inter-area-as-applicability-06 (work in progress), July 2016.

[I-D.dhodylee-pce-pcep-ls]
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information.", draft-dhodylee-pce-pcep-ls-07 (work in progress), March 2017.

[I-D.leedhody-pce-vn-association]
Lee, Y., Dhody, D., Zhang, X., and D. Ceccarelli, "PCEP Extensions for Establishing Relationships Between Sets of LSPs and Virtual Networks", draft-leedhody-pce-vn-association-02 (work in progress), March 2017.

[I-D.litkowski-pce-state-sync]
Litkowski, S., Sivabalan, S., and D. Dhody, "Inter Stateful Path Computation Element communication procedures", draft-litkowski-pce-state-sync-01 (work in progress), February 2017.

[I-D.ietf-pce-association-policy]
Dhody, D., Sivabalan, S., Litkowski, S., Tantsura, J., and J. Hardwick, "Path Computation Element communication Protocol extension for associating Policies and LSPs", draft-ietf-pce-association-policy-00 (work in progress), December 2016.

[I-D.ietf-pce-pceps]
Lopez, D., Dios, O., Wu, W., and D. Dhody, "Secure Transport for PCEP", draft-ietf-pce-pceps-11 (work in progress), January 2017.

[I-D.lee-teas-actn-abstraction]
Lee, Y., Dhody, D., Ceccarelli, D., and O. Dios, "Abstraction and Control of TE Networks (ACTN) Abstraction Methods", draft-lee-teas-actn-abstraction-00 (work in progress), October 2016.

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023
USA

EMail: leeyoung@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm
Sweden

EMail: daniele.ceccarelli@ericsson.com

PCE Working Group D. Dhody
Internet-Draft Huawei Technologies
Intended status: Standards Track D. King
Expires: September 28, 2017 Lancaster University
March 27, 2017

Experimental Codepoint Allocation for Path Computation Element communication Protocol (PCEP)

draft-dhody-pce-pcep-exp-codepoints-03

Abstract

IANA assigns values to the Path Computation Element (PCE) communication Protocol (PCEP) parameters (messages, objects, TLVs). IANA established a new top-level registry to contain all PCEP codepoints and sub-registries. The allocation policy for each new registry is by IETF Consensus.

This document seeks to mark some codepoints for experimental usage of 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].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Path Computation Element communication Protocol (PCEP) provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Clients (PCCs) requests.

In section 9 of [RFC5440], IANA assigns values to the PCEP protocol parameters (messages, objects, TLVs). IANA established a new top-level registry to contain all PCEP codepoints and sub-registries. The allocation policy for each new registry is by IETF Consensus as described in [RFC5226]. Specifically, new assignments are made via RFCs approved by the IESG. Typically, the IESG will seek input on prospective assignments from appropriate persons (e.g., a relevant

Working Group if one exists). Early allocation [RFC7120] provides some latitude for allocation of these code points, but is reserved for features that are considered appropriately stable.

With some recent advancement, there is an enhanced need to experiment with PCEP. It is often necessary to use some sort of number or constant in order to actually test or experiment with the new function, even when testing in a closed environment. In order to run experiment, it is important that the value won't collide not only with existing codepoints but any future allocation.

This document thus set apart some codepoints in PCEP registry and subregistries for experimental usage.

2. PCEP Messages

Some codepoints are requested to be set aside for experimentation with new PCEP messages. The suggested range is 246-255.

3. PCEP Objects

Some codepoints are requested to be set aside for experimentation with new PCEP objects. The suggested range is 224-255.

4. PCEP TLVs

Some codepoints are requested to be set aside for experimentation with new PCEP TLVs. The suggested range is 65280-65535.

5. Handling of unknown experimentation

A PCEP implementation that receives an experimental PCEP message, that it does not recognize, would react as per section 6.9 of [RFC5440] by sending a PCErr message with Error-value=2 (capability not supported).

A PCE that does not recognize an experimental PCEP object, MUST reject the entire PCEP message and MUST send a PCE error message with Error- Type="Unknown Object" or "Not supported object", defined as per [RFC5440].

As per section 7.1 of [RFC5440], unknown experimental PCEP TLV would be ignored.

6. IANA Considerations

IANA maintains the "Path Computation Element Protocol (PCEP) Numbers" at <http://www.iana.org/assignments/pcep>.

6.1. New PCEP Messages

Within this registry IANA maintains a sub-registry for PCEP Messages (see PCEP Messages at <http://www.iana.org/assignments/pcep>).

Upon approval of this document, IANA is requested to make the following allocations:

Type	Description	Allocation Policy
246-255	Unassigned	Experimental Use

6.2. New PCEP Objects

Within this registry IANA maintains a sub-registry for PCEP Objects (see PCEP Objects at <http://www.iana.org/assignments/pcep>).

Upon approval of this document, IANA is requested to make the following allocations:

Type	Description	Allocation Policy
224-255	Unassigned	Experimental Use

6.3. New PCEP TLVs

Within this registry IANA maintains a sub-registry for PCEP TLVs (see PCEP TLV Type Indicators at <http://www.iana.org/assignments/pcep>).

Upon approval of this document, IANA is requested to make the following allocations:

Type	Description	Allocation Policy
65280-65535	Unassigned	Experimental Use

7. Allocation Policy

The allocation policy for the IANA request in Section 6 is "Experimental". As per [RFC5226], IANA does not record specific assignments for any particular use for this policy.

As the experiment/standard progress and an early IANA allocation or RFC publication happens, the IANA defined codepoints are used and experimental code points are freed up.

8. Security Considerations

This document does not introduce any new security considerations to the existing protocol. Refer to [RFC5440] for further details of the specific security measures.

9. Acknowledgments

The authors would like to thank Ramon Casellas, Jeff Tantsura, Adrian Farrel, Jonathan Hardwick, Julien Mueric, Lou Berger, Michael Shroff, and Andrew Dolganow for their feedback and suggestions.

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, <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>.

10.2. Informative References

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC7120] Cotton, M., "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 7120, DOI 10.17487/RFC7120, January 2014, <http://www.rfc-editor.org/info/rfc7120>.

Appendix A. Other Codepoints

Based on the feedback from the WG, it was decided to focus only on the essentials in the scope of this documents. For others, Experiments can use a new experimental TLV/Object instead.

Authors' Addresses

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Daniel King
Lancaster University
UK

EMail: d.king@lancaster.ac.uk

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2018

J. Dong
M. Chen
D. Dhody
Huawei Technologies
J. Tantsura
Individual
K. Kumaki
KDDI Corporation
T. Murai
Furukawa Network Solution Corp.

July 3, 2017

BGP Extensions for Path Computation Element (PCE) Discovery
draft-dong-pce-discovery-proto-bgp-07

Abstract

In networks where a Path Computation Element (PCE) is used for path computation, it is desirable for the Path Computation Clients (PCCs) to discover dynamically and automatically a set of PCEs along with certain information relevant for PCE selection. RFC 5088 and RFC 5089 define the PCE discovery mechanisms based on Interior Gateway Protocols (IGP). This document defines extensions to BGP for the advertisement of PCE Discovery information. The BGP based PCE discovery mechanism is complementary to the existing IGP based mechanisms.

Requirements Language

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

Status of This Memo

This Internet-Draft will expire on January 4, 2018.

Copyright Notice

Table of Contents

1. Introduction

In networks where a Path Computation Element (PCE) is used for path computation, it is desirable for the Path Computation Clients (PCCs) to discover dynamically and automatically a set of PCEs along with certain information relevant for PCE selection. [RFC5088] and [RFC5089] define the PCE discovery mechanisms based on Interior Gateway Protocols (IGP). When PCCs are LSRs participating in the IGP (OSPF or IS-IS), and PCEs are either LSRs or servers also participating in the IGP, an effective mechanism for PCE discovery within an IGP routing domain consists of utilizing IGP advertisements.

[RFC4674] presents a set of requirements for a PCE discovery mechanism. This includes the discovery by a PCC of a set of one or more PCEs which may potentially be in some other domains. This is a desirable function in the case of inter-domain path computation. For example, Backward Recursive Path Computation (BRPC) [RFC5441] can be used by cooperating PCEs to compute an inter-AS path, in which case the discovery of PCE as well as the domain information is useful.

BGP has been extended for north-bound distribution of routing and TE information to PCE [RFC7752] and [I-D.ietf-idr-te-pm-bgp]. Similary this document extends BGP to also carry the PCE discovery information.

This document defines extensions to BGP to allow a PCE to advertise its location, along with some information useful to a PCC for the PCE selection, so as to satisfy dynamic PCE discovery requirements set forth in [RFC4674].

This specification contains two parts: definition of a new BGP-LS NLRI [RFC7752] that describes PCE information and definition of PCE Attribute TLVs as part of BGP-LS attributes.

2. Carrying PCE Discovery Information in BGP

2.1. PCE NLRI

The PCE discovery information is advertised in BGP UPDATE messages using the MP_REACH_NLRI and MP_UNREACH_NLRI attributes [RFC4760]. The "Link- State NLRI" defined in [RFC7752] is extended to carry the PCE information. BGP speakers that wish to exchange PCE discovery information MUST use the BGP Multiprotocol Extensions Capability Code (1) to advertise the corresponding (AFI, SAFI) pair, as specified in [RFC4760].

The format of "Link-State NLRI" is defined in [RFC7752]. A new "NLRI Type" is defined for PCE Information as following:

o Type = TBD1: PCE NLRI

The format of PCE NLRI is shown in the following figure:

```
     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
    +-+-+-+-+-+-+-+-+
    |  Protocol-ID  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Identifier                          |
    |                            (64 bits)                          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                   PCE Descriptors (variable)                  ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. PCE NLRI

The 'Protocol-ID' field is defined in [RFC7752], to be set to the appropriate value that indicates the source of the PCE information. If BGP speaker and PCE are co-located, the Protocol-ID SHOULD be set to "Direct". If PCE information to advertise is configured at the BGP speaker, the Protocol-ID SHOULD be set to "Static configuration".

As defined in [RFC7752], the 64-Bit 'Identifier' field is used to identify the "routing universe" where the PCE belongs.

2.1.1. PCE Descriptors

The PCE Descriptor field is a set of Type/Length/Value (TLV) triplets. The format of each TLV is as per Section 3.1 of [RFC7752]. The PCE Descriptor TLVs uniquely identify a PCE. The following PCE descriptor are defined -

Codepoint	Descriptor TLV	Length
TBD2	IPv4 PCE Address	4
TBD3	IPv6 PCE Address	16

Table 1: PCE Descriptors

The PCE address TLVs specifies an IP address that can be used to reach the PCE. The PCE-ADDRESS Sub-TLV defined in [RFC5088] and [RFC5089] is used in the OSPF and IS-IS respectively. The format of the PCE address TLV are -

```
        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=TBD2        |             Length=4          |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                       IPv4 PCE Address                        |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

        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=TBD3        |             Length=16         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                                                               |
       |                       IPv6 PCE Address                        |
       |                                                               |
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. PCE Address TLVs

When the PCE has both an IPv4 and IPv6 address, both the TLVs MAY be included.

2.2. PCE Attribute TLVs

PCE Attribute TLVs are TLVs that may be encoded in the BGP-LS attribute [RFC7752] with a PCE NLRI. The format of each TLV is as per Section 3.1 of [RFC7752]. The format and semantics of the Value fields in some PCE Attribute TLVs correspond to the format and semantics of the Value fields in IS-IS PCED Sub-TLV, defined in [RFC5089]. Other PCE Attribute TLVs are defined in this document.

The following PCE Attribute TLVs are valid in the BGP-LS attribute with a PCE NLRI:

TLV Code Point	Description	IS-IS TLV /Sub-TLV	Reference (RFC/Section)
TBD4	Path Scope	5/2	[RFC5089]/4.2
TBD5	PCE Domain	-	-
TBD6	Neighbor PCE Domain	-	-
TBD7	PCE Capability	5/5	[RFC5089]/4.5

Table 2: PCE Attribute TLVs

The format and semantics of Path Scope and PCE capability is as per [RFC5089]. The Path Scope TLV is mandatory.

2.2.1. PCE Domain TLV

The PCE Domain TLV specifies a PCE-Domain (IGP area and/or AS) where the PCE has topology visibility and through which the PCE can compute paths.

```
 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=TBD5        |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        Domain Sub-TLVs (variable)           //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The length of this TLV is variable. The value contains one or more domain sub-TLVs as listed below -

Sub-TLV Code Point	Description	Length
512	Autonomous System	4
514	OSPF Area-ID	4
1027	IS-IS Area Identifier	Variable

Multiple sub-TLVs MAY be included, when the PCE has visibility into multiple PCE-Domains.

2.2.2. Neighbor PCE Domain TLV

The Neighbor PCE Domain TLV specifies a neighbor PCE-Domain (IGP area and/or AS) toward which a PCE can compute paths.

```
 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=TBD6        |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        Domain Sub-TLVs (variable)           //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The length of this TLV is variable. The value contains one or more domain sub-TLVs as listed above. Multiple sub-TLVs MAY be included, when the PCE can compute paths towards several neighbor PCE-Domains.

3. Operational Considerations

Existing BGP-LS operational procedures apply to the advertisement of PCE information as per [RFC7752]. This information is treated as pure application level data which has no immediate impact on forwarding states. The PCE information SHOULD be advertised only to the domains where such information is allowed to be used. This can be achieved by policy control on the ASBRs.

The PCE information is considered relatively stable and does not change frequently, thus this information will not bring significant impact on the amount of BGP updates in the network.

4. IANA Considerations

IANA needs to assign a new NLRI Type for 'PCE NLRI' from the "BGP-LS NLRI-Types" registry.

IANA needs to assign new TLV code point as per Table 1 and 2 from the "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, and Attribute TLVs" registry.

[Editor's Note - Check if name of the registry should be changes with following instructions - Further IANA is requested to rename the registry as "BGP-LS Node Descriptor, Link Descriptor, Prefix Descriptor, PCE Descriptor, and Attribute TLVs".]

5. Security Considerations

Procedures and protocol extensions defined in this document do not affect the BGP security model. See the 'Security Considerations' section of [RFC4271] for a discussion of BGP security. Also refer to [RFC4272] and [RFC6952] for analysis of security issues for BGP.

Existing BGP-LS security considerations as per [RFC7752] continue to apply.

6. Contributors

The following individuals gave significant contributions to this document:

Takuya Miyasaka
KDDI Corporation
ta-miyasaka@kddi.com

7. Acknowledgements

The authors would like to thank Zhenbin Li, Hannes Gredler, Jan Medved, Adrian Farrel, Julien Meuric and Jonathan Hardwick for the valuable discussion and comments.

8. References

8.1. Normative References

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

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

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <http://www.rfc-editor.org/info/rfc4760>.

[RFC5088] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, DOI 10.17487/RFC5088, January 2008, <http://www.rfc-editor.org/info/rfc5088>.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, DOI 10.17487/RFC5089, January 2008, <http://www.rfc-editor.org/info/rfc5089>.

[RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <http://www.rfc-editor.org/info/rfc7752>.

8.2. Informative References

[I-D.ietf-idr-te-pm-bgp]
Previdi, S., Wu, Q., Gredler, H., Ray, S., jefftant@gmail.com, j., Filsfils, C., and L. Ginsberg, "BGP-LS Advertisement of IGP Traffic Engineering Performance Metric Extensions", draft-ietf-idr-te-pm-bgp-06 (work in progress), June 2017.

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <http://www.rfc-editor.org/info/rfc4272>.

[RFC4674] Le Roux, J., Ed., "Requirements for Path Computation Element (PCE) Discovery", RFC 4674, DOI 10.17487/RFC4674, October 2006, <http://www.rfc-editor.org/info/rfc4674>.

[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>.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <http://www.rfc-editor.org/info/rfc6952>.

Authors' Addresses

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: jie.dong@huawei.com

Mach(Guoyi) Chen
Huawei Technologies
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: mach.chen@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Jeff Tantsura
Individual
US

Email: jefftant.ietf@gmail.com

Kenji Kumaki
KDDI Corporation
Garden Air Tower, Iidabashi, Chiyoda-ku
Tokyo 102-8460
Japan

Email: ke-kumaki@kddi.com

Tomoki Murai
Furukawa Network Solution Corp.
5-1-9, Higashi-Yawata, Hiratsuka
Kanagawa 254-0016
Japan

Email: murai@fnsc.co.jp

MPLS Working Group T. Saad
Internet-Draft Juniper Networks
Intended status: Standards Track R. Gandhi
Expires: May 5, 2021 Cisco Systems, Inc.
X. Liu
Volta Networks
V. Beeram
Juniper Networks
I. Bryskin
Huawei Technologies
November 01, 2020

A YANG Data Model for MPLS Static LSPs
draft-ietf-mpls-static-yang-12

Abstract

This document contains the specification for the MPLS Static Label Switched Paths (LSPs) YANG model. The model allows for the provisioning of static LSP(s) on Label Edge Router(s) LER(s) and Label Switched Router(s) LSR(s) devices along a LSP path without the dependency on any signaling protocol. The MPLS Static LSP model augments the MPLS base YANG model with specific data to configure and manage MPLS Static LSP(s).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a YANG [RFC7950] data model for configuring and managing the Multiprotocol Label Switching (MPLS) [RFC3031] Static LSPs. The model allows the configuration of LER and LSR devices with the necessary MPLS cross-connects or bindings to realize an end-to-end LSP service.

A static LSP is established by manually specifying incoming and outgoing MPLS label(s) and necessary forwarding information on each of the traversed LER and LSR devices (ingress, transit, or egress nodes) of the forwarding path.

For example, on an ingress LER device, the model is used to associate a specific Forwarding Equivalence Class (FEC) of packets- e.g. matching a specific IP prefix in a Virtual Routing or Forwarding (VRF) instance- to an MPLS outgoing label imposition, next-hop(s) and respective outgoing interface(s) to forward the packet. On an LSR device, the model is used to create a binding that swaps the incoming label with an outgoing label and forwards the packet on one or multiple egress path(s). On an egress LER, it is used to create a binding that decapsulates the incoming MPLS label and performs forwarding based on the inner MPLS label (if present) or IP forwarding in the packet.

The MPLS Static LSP YANG model is broken into two modules "ietf-mpls-static" and "ietf-mpls-static-extended". The "ietf-mpls-static" module covers basic features for the configuration and management of unidirectional Static LSP(s), while "ietf-mpls-static-extended" covers extended features like the configuration and management of bidirectional Static LSP(s) and LSP admission control.

The module "ietf-mpls-static" augments the MPLS Base YANG model defined in module "ietf-mpls" in [I-D.ietf-mpls-base-yang].

1.1. Terminology

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

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

1.2. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

LSR: Label Switching Router

LER: Label Edge Router

FEC: Forwarding Equivalence Class

NHLFE: Next Hop Label Forwarding Entry

ILM: Incoming Label Map

2. MPLS Static LSP Model

2.1. Model Organization

The base MPLS Static LSP model covers the core features with the minimal set of configuration parameters needed to manage and operate MPLS Static LSPs.

Additional MPLS Static LSP parameters as well as optional feature(s) are grouped in a separate MPLS Static LSP extended model. The relationship between the MPLS base and other MPLS modules are shown in Figure 1.

```
     Routing module   +----------------+    v: import
                      | ietf-routing   |    o: augment
                      +----------------+
                          o
                          |
                          v
     MPLS base        +------------+    v: import
     module           | ietf-mpls  |    o: augment
                      +------------+
                         o          o
                         |           \
                         v            v
                 +-------------------+ +---------------------+
     MPLS Static | ietf-mpls-static  | | ietf-mpls-ldp.yang  | . . .
     LSP module  +-------------------+ +---------------------+
                             o
                             |
                             v
                    +----------------------------+
    Extended MPLS   | ietf-mpls-static-extended  |
    Static LSP      +----------------------------+
    module
```

Figure 1: Relationship between MPLS modules

2.2. Model Tree Diagram

The MPLS Static and extended LSP tree diagram as per [RFC8340] is shown in Figure 2.

```
  module: ietf-mpls-static
    augment /rt:routing/mpls:mpls:
      +--rw static-lsps
         +--rw static-lsp* [name]
         |  +--rw name           string
         |  +--rw operation?     mpls:mpls-operations-type
```

```
         |  +--rw in-segment
         |  |  +--rw fec
         |  |     +--rw (type)?
         |  |     |  +--:(ip-prefix)
         |  |     |  |  +--rw ip-prefix?        inet:ip-prefix
         |  |     |  +--:(mpls-label)
         |  |     |     +--rw incoming-label?   rt-types:mpls-label
         |  |     +--rw incoming-interface?     if:interface-ref
         |  +--rw out-segment
         |     +--rw (out-segment)?
         |        +--:(nhlfe-single)
         |        |  +--rw nhlfe-single
         |        |     +--rw mpls-label-stack
         |        |     |  +--rw entry* [id]
         |        |     |     +--rw id               uint8
         |        |     |     +--rw label?           rt-types:mpls-label
         |        |     |     +--rw ttl?             uint8
         |        |     |     +--rw traffic-class?   uint8
         |        |     +--rw outgoing-interface?   if:interface-ref
         |        +--:(nhlfe-multiple)
         |           +--rw nhlfe-multiple
         |              +--rw nhlfe* [index]
         |                 +--rw index                 string
         |                 +--rw backup-index?         string
         |                 +--rw loadshare?            uint16
         |                 +--rw role?                 nhlfe-role
         |                 +--rw mpls-label-stack
         |                 |  +--rw entry* [id]
         |                 |     +--rw id               uint8
         |                 |     +--rw label?
         |                 |     |       rt-types:mpls-label
         |                 |     +--rw ttl?             uint8
         |                 |     +--rw traffic-class?   uint8
         |                 +--rw outgoing-interface?   if:interface-ref
         +--rw mpls-static-ext:bandwidth?            uint32
         +--rw mpls-static-ext:lsp-priority-setup?   uint8
         +--rw mpls-static-ext:lsp-priority-hold?    uint8

  module: ietf-mpls-static-extended
    augment /rt:routing/mpls:mpls:
      +--rw bidir-static-lsps
         +--rw bidir-static-lsp* [name]
            +--rw name           string
            +--rw forward-lsp?   mpls-static:static-lsp-ref
            +--rw reverse-lsp?   mpls-static:static-lsp-ref
```

Figure 2: MPLS Static LSP tree diagram

2.3. Model Overview

This document defines two YANG modules for MPLS Static LSP(s) configuration and management: ietf-mpls-static.yang and ietf-mpls-static-extended.yang.

The ietf-mpls-static module contains the following high-level types and groupings:

static-lsp-ref:

A YANG reference type for a static LSP that can be used by data models to reference a configured static LSP.

in-segment:

A YANG grouping that describes parameters of an incoming class of FEC associated with a specific LSP as described in the MPLS architecture document [RFC3031]. The model allows the following types of traffic to be mapped onto the static LSP on an ingress LER:

- o Unlabeled traffic destined to a specific prefix
- o Labeled traffic arriving with a specific label

out-segment:

A YANG grouping that describes parameters for the forwarding path(s) and their associated attributes for an LSP. The model allows for the following cases:

- o single forwarding path or NHLFE
- o multiple forwarding path(s) or NHLFE(s), each of which can serve a primary, backup or both role(s).

The ietf-mpls-static-extended module contains the following high-level types and groupings:

bidir-static-lsp:

A YANG grouping that describes list of static bidirectional LSPs

The ietf-mpls-static-extended augments the ietf-mpls-static model with additional parameters to configure and manage:

- o Bidirectional Static LSP(s)

- o Defining Static LSP bandwidth allocation

- Defining Static LSP preemption priorities

2.4. Model YANG Module(s)

Configuring LSPs through an LSR/LER involves the following steps:

- Enabling MPLS on MPLS capable interfaces.

- Configuring in-segments and out-segments on LER(s) and LSR(s) traversed by the LSP.

- Setting up the cross-connect per LSP to associate segments and/or to indicate connection origination and termination.

- Optionally specifying label stack actions.

- Optionally specifying segment traffic parameters.

The objects covered by this model are derived from the Incoming Label Map (ILM) and Next Hop Label Forwarding Entry (NHLFE) as specified in the MPLS architecture document [RFC3031].

The ietf-mpls-static module imports the followinig modules:

- ietf-inet-types defined in [RFC6991]

- ietf-routing defined in [RFC8349]

- ietf-routing-types defined in [RFC8294]

- ietf-interfaces defined in [RFC8343]

- ietf-mpls defined in [I-D.ietf-mpls-base-yang]

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

The ietf-mpls-static module is shown below:

```
<CODE BEGINS> file "ietf-mpls-static@2019-09-12.yang"
module ietf-mpls-static {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-static";
  prefix "mpls-static";

  import ietf-mpls {
    prefix "mpls";
    reference "draft-ietf-mpls-base-yang: MPLS Base YANG Data Model";
  }
```

```
   import ietf-routing {
     prefix "rt";
     reference "RFC8349: A YANG Data Model for Routing Management";
   }

   import ietf-routing-types {
     prefix "rt-types";
     reference "RFC8294: Common YANG Data Types for the Routing Area";
   }

   import ietf-inet-types {
     prefix inet;
     reference "RFC6991: Common YANG Data Types";
   }

   import ietf-interfaces {
     prefix "if";
     reference "RFC7223: A YANG Data Model for Interface Management";
   }

   organization "IETF MPLS Working Group";

   contact
     "WG Web:   <http://tools.ietf.org/wg/mpls/>

      WG List:  <mailto:mpls@ietf.org>

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

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

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

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

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

   description
      "This YANG module augments the 'ietf-routing' module with basic
       configuration and operational state data for MPLS static
       The model fully conforms to the Network Management Datastore
       Architecture (NMDA).
```

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

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

   // RFC Ed.: replace XXXX with actual RFC number and remove this
   // note.

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

   revision "2019-09-12" {
     description
       "Latest revision of MPLS Static LSP YANG module";
     reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";
   }

   typedef static-lsp-ref {
     type leafref {
       path "/rt:routing/mpls:mpls/mpls-static:static-lsps/" +
            "mpls-static:static-lsp/mpls-static:name";
     }
     description
       "This type is used by data models that need to reference
        configured static LSP.";
   }

   grouping in-segment {
     description "In-segment grouping";
     container in-segment {
       description "MPLS incoming segment";
       container fec {
         description "Forwarding Equivalence Class grouping";
         choice type {
           description "FEC type choices";
           case ip-prefix {
             leaf ip-prefix {
               type inet:ip-prefix;
               description "An IP prefix";
             }
           }
```

```
            case mpls-label {
              leaf incoming-label {
                type rt-types:mpls-label;
                description "label value on the incoming packet";
              }
            }
          }
          leaf incoming-interface {
            type if:interface-ref;
            description
              "Optional incoming interface if FEC is restricted
               to traffic incoming on a specific interface";
          }
        }
      }
    }

    grouping out-segment {
      description "Out-segment grouping";
      container out-segment {
        description "MPLS outgoing segment";
        choice out-segment {
          description "The MPLS out-segment type choice";
          case nhlfe-single {
            container nhlfe-single {
              description "Container for single NHLFE entry";
              uses mpls:nhlfe-single-contents;
              leaf outgoing-interface {
                type if:interface-ref;
                description
                  "The outgoing interface";
              }
            }
          }
          case nhlfe-multiple {
            container nhlfe-multiple {
              description "Container for multiple NHLFE entries";
              list nhlfe {
                key index;
                description "MPLS NHLFE entry";
                uses mpls:nhlfe-multiple-contents;
                leaf outgoing-interface {
                  type if:interface-ref;
                  description
                    "The outgoing interface";
                }
              }
            }
```

```
         }
       }
     }
   }

   augment "/rt:routing/mpls:mpls" {
     description "Augmentations for MPLS Static LSPs";
     container static-lsps {
       description
         "Statically configured LSPs, without dynamic signaling";
       list static-lsp {
         key name;
         description "list of defined static LSPs";
         leaf name {
           type string;
           description "name to identify the LSP";
         }
         leaf operation {
           type mpls:mpls-operations-type;
           description
             "The MPLS operation to be executed on the incoming packet";
         }
         uses in-segment;
         uses out-segment;
       }
     }
   }
 }
 <CODE ENDS>
```

The ietf-mpls-static-extended module imports the followinig modules:

- ietf-mpls defined in [I-D.ietf-mpls-base-yang]

- ietf-mpls-static defined in this document

- ietf-routing defined in [RFC8349]

The ietf-mpls-static-extended module is shown below:

```
   <CODE BEGINS> file "ietf-mpls-static-extended@2019-09-12.yang"
   module ietf-mpls-static-extended {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended";
     prefix "mpls-static-ext";

     import ietf-mpls {
       prefix "mpls";
```

```
       reference "draft-ietf-mpls-base-yang: MPLS Base YANG Data Model";
     }

     import ietf-routing {
       prefix "rt";
       reference "RFC8349: A YANG Data Model for Routing Management";
     }

     import ietf-routing-types {
       prefix "rt-types";
       reference "RFC8294: Common YANG Data Types for the Routing Area";
     }

     import ietf-mpls-static {
       prefix "mpls-static";
       reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";
     }

     organization "IETF MPLS Working Group";

     contact
       "WG Web:   <http://tools.ietf.org/wg/mpls/>

        WG List:  <mailto:mpls@ietf.org>

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

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

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

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

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

     description
       "This YANG module contains the Extended MPLS Static LSP YANG
        data model. The model fully conforms to the Network Management
        Datastore Architecture (NMDA).

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

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

     // RFC Ed.: replace XXXX with actual RFC number and remove this
     // note.

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

     revision "2019-09-12" {
       description
         "Latest revision of MPLS Static LSP Extended YANG module";
       reference "RFC XXXX: A YANG Data Model for MPLS Static LSPs";
     }

     grouping bidir-static-lsp {
       description
         "grouping for top level list of static bidirectional LSPs";
       leaf forward-lsp {
         type mpls-static:static-lsp-ref;
         description
           "Reference to a configured static forward LSP";
       }
       leaf reverse-lsp {
         type mpls-static:static-lsp-ref;
         description
           "Reference to a configured static reverse LSP";
       }
     }

     augment "/rt:routing/mpls:mpls/mpls-static:static-lsps" {
       description
         "Augmentation for static MPLS LSPs";

       leaf bandwidth {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Bandwidth using offline calculation";
       }
       leaf lsp-priority-setup {
         type uint8 {
```

```
            range "0..7";
          }
          description "LSP setup priority";
        }
        leaf lsp-priority-hold {
          type uint8 {
            range "0..7";
          }
          description "LSP hold priority";
        }
      }

      augment "/rt:routing/mpls:mpls" {
        description "Augmentations for MPLS Static LSPs";
        container bidir-static-lsps {
          description
            "Statically configured bidirectional LSPs";
          list bidir-static-lsp {
            key name;
            description "List of static bidirectional LSPs";

            leaf name {
              type string;
              description "Name that identifies the bidirectional LSP";
            }
            uses bidir-static-lsp;
          }
        }
      }
    }
    <CODE ENDS>
```

3. IANA Considerations

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

URI: urn:ietf:params:xml:ns:yang:ietf-mpls-static
Registrant Contact: The MPLS WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended
Registrant Contact: The MPLS WG of the IETF.
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-mpls-static
namespace:  urn:ietf:params:xml:ns:yang:ietf-mpls-static
prefix:     ietf-mpls-static
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:  RFCXXXX

name:       ietf-mpls-static-extended
namespace:  urn:ietf:params:xml:ns:yang:ietf-mpls-static-extended
prefix:     ietf-mpls-static-extended
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:  RFCXXXX
```

4. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

All nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default) 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 /ietf-routing:routing/ietf-mpls:mpls:/ietf-mpls:static-lsps: This entire subtree is related to security.

An administrator needs to restrict write access to all configurable objects within this data model.

5. Contributors

Himanshu Shah
Ciena
email: hshah@ciena.com

Kamran Raza
Cisco Systems, Inc.
email: skraza@cisco.com

6. References

6.1. Normative References

[I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-17 (work in progress), October 2020.

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

[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>.

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

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

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

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

6.2. Informative References

[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>.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad.net@gmail.com

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

PCE Working Group H. Pouyllau
Internet-Draft Alcatel-Lucent
Updates: 5440 (if approved) R. Theillaud
Intended status: Standards Track Marben Products
Expires: August 22, 2021 J. Meuric
Orange
H. Zheng (Editor)
X. Zhang
Huawei Technologies
February 18, 2021

Extensions to the Path Computation Element Communication Protocol for Enhanced Errors and Notifications
draft-ietf-pce-enhanced-errors-09

Abstract

This document defines new error and notification TLVs for the PCE Communication Protocol (PCEP) specified in RFC5440, and will update it. It identifies the possible PCEP behaviors in case of error or notification. Thus, this draft defines types of errors and how they are disclosed to other PCEs in order to support predefined PCEP behaviors.

Status of This Memo

Copyright Notice

Table of Contents

1. Terminology

PCE terminology is defined in [RFC4655].

PCEP Peer: An element involved in a PCEP session (i.e. a PCC or a PCE).

Source PCC: the PCC, for a given path computation query, initiating the first PCEP request, which may then trigger a chain of successive requests.

Target PCE: the PCE that can compute a path to the destination without having to query any other PCE.

2. Conventions used in this document

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

3. Introduction

The PCE Communication Protocol [RFC5440] is designed to be flexible and extensible in order to allow future evolutions or specific constraint support such as proposed in [RFC7470]. Crossing different PCE implementations (e.g. from different providers or due to different releases), a PCEP request may encounter unknown errors or notification messages. In such a case, the PCEP RFC [RFC5440] specifies to send a specific error code to the PCEP peer. This document updates [RFC5440] by introducing mechanism to propagate the error message, with specifying error and notification TLVs.

In the context of path computation crossing different routing domains or autonomous systems, the number of different PCE system specificities is potentially high, thus possibly leading to divergent and unstable situations. Such phenomenon can also occur in homogeneous cases since PCE systems have their own policies that can introduce differences in requests treatment even for requests having the same destination. In order to generalize PCEP behaviors in the case of heterogeneous PCE systems, new objects have to be defined. Dealing with heterogeneity is a major challenge considering PCE applicability, particularly in multi-layer, multi-domain and H-PCE contexts [RFC8751]. Thus, extending such error codes and PCEP behaviors accordingly would improve interoperability among different PCEP implementations and would solve some of these issues. However, some of them would still remain (e.g. the divergences in request treatment introduced by different policies).

The purpose of this draft is to identify and specify new optional TLVs and objects in order to generalize PCEP behaviors.

3.1. Examples

The two following scenarios underline the need for a normalization of the PCEP behaviors according to existing error or notification types.

3.1.1. Error use-case

PCE(i-1) has sent a request to PCE(i) which has also sent a request to PCE(i+1). PCE(i-1) and PCE(i+1) have the same error semantic but not PCE(i). If PCE(i+1) throws an error type and value unknown by PCE(i). PCE(i) could then adopt any other behaviors and sends back to PCE(i-1) an error of type 2 (Capability not supported), 3 (Unknown Object) or 4 (Not supported Object) for instance. As a consequence, the path request would be cancelled but the error has no meaning for PCE(i-1) whereas if PCE(i) had simply forwarded the error sent by PCE(i+1), it would have been understood by PCE(i-1).

3.1.2. Notification use-case

PCE(i-1) has sent a request to PCE(i) which has also sent a request to PCE(i+1) but PCE(i+1) is overloaded. Without extensions, PCE(i+1) should send a notification of type 2 and a value flag giving its estimated congestion duration. PCE(i) can choose to stop the path computation and send a NO_PATH reply to PCE(i-1). Hence, PCE(i-1) ignores the congestion duration on PCE(i+1) and could seek it for further requests.

4. PCEP Behaviors

One of the purposes of the PCE architecture is to compute paths across networks, but an added value is to compute such paths in inter-area/layer/domain environments. The PCE Communication Protocol [RFC5440] is based on the Transport Communication Protocol (TCP). Thus, to compute a path within the PCE architecture, several TCP/PCEP sessions have to be set up, in a peer-to-peer manner, along a set of identified PCEs.

When the PCEP session is up for two PCEP peers, the PCC of the first PCE System (the source PCC) sends a PCReq message. If the PCC does not receive any reply before the dead timer is out, then it goes back to the idle state. A PCC can expect two kinds of replies: a PCRep message containing one or more valid paths (EROs) or a negative PCRep message containing a NO-PATH object.

Beside PCReq and PCRep messages, notification and error messages, named respectively PCNtf and PCErr, can be sent. There are two types of notification messages: type 1 is for cancelling pending requests and type 2 for signaling a congestion of the PCE. Several error values are described in [RFC5440]. The error types concerning the session phase begin at 2, error type 1 values are dedicated to the initialization phase.

As the PCE Communication Protocol is built to work in a peer-to-peer manner (i.e. supported by a TCP Connection), it supposes that the "deadtimer" of the source PCC is long enough to support the end-to-end distributed path computation process.

The exchange of messages in the PCE Communication Protocol is described in details when PCEP is in states OpenWait and KeepWait in [RFC5440]. When the session is up, message exchange is defined in [RFC5440]. [RFC5441] describes the Backward Recursive Path Computation (BRPC) procedure, and, because it considers an inter-domain path computation, gives a bigger picture of the possible behaviors when the session is up. Detailed behavior is mostly let free to any specific implementation. The following sections identifies the PCEP behaviors in case of error or notification and also introduce the requirement of PCEP peer identification in both cases.

4.1. PCEP Behaviors in Case of Error

[RFC5440] specifies that "a PCEP Error message is sent in several situations: when a protocol error condition is met or the request is not compliant with the PCEP specification". On this basis, and according to the other RFCs, the identified PCEP behaviors are the followings:

o "Propagation": the received message requires to be propagated forwardly or backwardly (depending on which PCEP peer has sent the message) to a set of PCEP peers;

o "Criticality level": in different RFCs, error-types affects the state of the PCEP request or session in different manners; hence, different level of criticality can be observed:

o

   * Low-level of criticality: the received message does not affect the PCEP connection and further answer can still be expected;

   * Medium-level of criticality: the received message does not affect the PCEP connection but the request(s) is(are) cancelled;

   * High-level of criticality: the received message indicates that the PCEP peer will close the session with its peer (and so

pending requests associated by the error, if any, are cancelled.)

The high-level of criticality has been extracted from [RFC5440] which associates such a behavior to error-type of 1 (errors raised during the PCEP session establishment). Hence, such errors are quite specific. For the sake of completeness, they have been included in this document.

4.2. PCEP Behaviors in Case of Notification

Notification messages can be employed in two different manners: during the treatment of a PCEP request, or independently from it to advertise information (in [RFC5440], the request ID list within a PCNtf message is optional). Hence, three different types of behaviors can be identified:

- o "Local": the notification does not imply any forward or backward propagation of the message;

- o "Request-specific propagation": the received message requires to be propagated forwardly or backwardly (depending on which peer has sent the message) to the PCEP peers;

- o "Non request-specific propagation": the received message must be propagated to any known peers (e.g. if PCE discovery is activated) or to a list of identified peers.

4.3. PCE Peer Identification

The propagation of errors and notifications affects the state of the PCEP peers along the chain. In some cases, for instance a notification that a PCE is overloaded, the identification of the PCEP peer - or that the sender PCE is not the direct neighbor - might be an important information for the PCEP peers receiving the message. The ID of sender PCE is not carried in the error TLVs, but can be achieved via the speaker entity ID TLV during state synchronization. An example can be found in [RFC8232].

5. PCEP Extensions for Error and Notification Handling

This section describes extensions to support error and notification with respect to the PCEP behavior description defined in Section 4. This document does not intend to modify errors and notification types previously defined in existing documents (e.g. [RFC5440], [RFC5441], etc.). Error related TLVs have been specified in this section, while the notification functionality can be achieved via using PCNtf message with RP object with no need to extend further notification type.

5.1. Propagation TLV

To support the propagation behavior mentioned in Section 4.1 and Section 4.2, a new optional TLV is defined, which can be carried in PCEP-ERROR and NOTIFICATION objects, to indicate whether a message has to be propagateed or not. The allocation from the "PCEP TLV Type Indicators" sub-registry will be assigned by IANA and the request is documented in Section 10.

The description is "Propagation", the length value is 2 bytes and the value field is 1 byte. The value field is set to 0 meaning that the message MUST NOT be propagated. If the value field is set to 1, the message MUST be propagated. Section 5.4 specifies the destination and to limit the number of messages.

5.2. Error-criticality TLV

To support the shutdown behavior mentioned in Section 4.1, we extend the PCEP-ERROR object by creating a new optional TLV to indicate whether an error is recoverable or not. The allocation from the "PCEP TLV Type Indicators" sub-registry will be assigned by IANA and the request is documented in Section 10.

The description is "Error-criticality", the length value is 2 bytes and the value field is 1 byte. The value field is set to 0 meaning that the error has a low-level of criticality (so further messages can be expected for this request). If the value field is set to 1, the error has a medium-level of criticality and requests whose identifiers appear in the same message MUST be cancelled (so no further messages can be expected for these requests). If the value field is set to 2, the error has a high-level of criticality, the connection for this PCEP session is closed by the sender PCE peer.

5.3. Behaviors and TLV combinations

The propagation behavior MAY be combined with all criticality levels, thus leading to 6 different behaviors. In the case of a criticality level of 2, the session is closed by the PCE peer which sends the message. Hence, the criticality level is purely informative for the PCE peer which receives the message. If it is combined with a propagation behavior, then the PCE propagating the message MUST indicate the same level of criticality if it closes the session. Otherwise, it MUST use a criticality level of 1 if it does not close the session.

For a PCErr message, all the possible behaviors described in Section 4.1 can be covered with TLVs included in a PCEP-ERROR object. The following table captures all combinations of error behaviors:

Error criticallity value \ Propogation Value	0 (No Propagation)	1 (Propogation Required)
0 (low)	Type 1	Type 4
1 (medium)	Type 2	Type 5
2 (high)	Type 3	Type 6

- "Error Behavior Type 1" : Local Error with a low level of criticality;
- "Error Behavior Type 2": Local Error with a medium level of criticality;
- "Error Behavior Type 3": Local Error with a high level of criticality;
- "Error Behavior Type 4": Propagated Error with a low level of criticality;
- "Error Behavior Type 5": Propagated Error with a medium level of criticality;
- "Error Behavior Type 6": Propagated Error with a high level of criticality;

5.4. Propagation Restrictions TLVs

In order to limit the propagation of errors and notifications, the following mechanisms SHOULD be used:

A Time-To-Live(TTL) RLV: to limit the number of PCEP peers that will recursively receive the message;

A DIFFUSION-LIST TLV: to specify the PCEP peer addresses or domains of PCEP peers the message must be propagate to;

History mechanism: if a PCEP peer keeps track of the messages it has relayed, it could avoid propagating an error or notification it has already received.

Such mechanisms SHOULD be used jointly or independently depending the error or notification behaviors they are associated to. The conditions of use for the TTL and DIFFUSION-LIST TLVs are described in sections below.

5.4.1. Time-To-Live (TTL) TLV

The TTL value is set to any integer value to indicate the number of PCEP peers that will recursively receive the message. The TTL TLV SHOULD be used with propagated errors or notifications ("Propagation" TLV with value 1 in PCEP-ERROR or NOTIFICATION objects). Each PCEP peer MUST decrement the TTL value before propagating the message. When the TTL value becomes 0, the message is no more propagated.

If the message to be propagated is request-specific and there is no TTL or DIFFUSION-LIST TLVs included, the message MUST reach the source PCC (or alternatively the target PCE).

5.4.2. DIFFUSION-LIST TLV

The DIFFUSION-LIST TLV can be carried within either the error object of a PCErr message, or the notification object of a PCNtf message. It can either be used in a message sent by a PCC to a PCE or vice versa. The DIFFUSION-LIST MAY be used with propagated errors (TLV "Propagation"at value 1 in PCEP-ERROR object).

The format of the DIFFUSION-LIST object body is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |         Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                          (Sub-objects)                      //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type (16 bits): restricts the diffusion to certain peers. The following values are currently defined:

0: Any PCEP peer indicated in the list must be reached.

1: Only PCEs must be reached (and not PCC).

2: All PCEP peers with which a session is still opened must be reached.

The value of DIFFUSION-LIST is made of sub-objects similar to the IRO defined in [RFC5440]. The following sub-object types are supported.

Type	Sub-object
1	IPv4 address
2	IPv6 address
4	Unnumbered Interface ID
5	4-byte AS number
6	OSPF area ID
7	IS-IS Area ID
32	Autonomous System number
33	Explicit eXclusion Route Sub-object (EXRS)

If the error or notification codes target specific PCEP peers, a DIFFUSION-LIST TLV avoids partially flooding all PCEP peers. Any PCEP peer receiving a PCErr or PCNTf message containing a PCEP-ERROR or a NOTIFICATION object with a TLV "Propagation" at value 1 and where a DIFFUSION-LIST appears, MUST remove the addresses of the PCEP peers from the DIFFUSION-LIST, before sending the message to any other PCEP peers. This is performed by adding the PCEP peer addresses to the Explicit eXclusion Route Sub-object of the DIFFUSION-LIST. If a DIFFUSION-LIST value is empty, the PCEP peer MUST NOT propagate the message to any peer.

Note that, a Diffusion-List could contain strict or loose addresses to refer to a network domain (e.g. an Autonomous System number, an OSPF area, an IP address). Hence, the PCEP peers targeted by the message would be the PCEP peers covering the corresponding domain. If an address is loose, each time a PCEP peer forwards a message to another PCEP peer of this address, it MUST add it own address to the Explicit eXclusion Route Sub-object (EXRS) of the Diffusion-List for any forwarded messages. Hence, a PCE SHOULD avoid forwarding the same message repeated to the same set of peers. Finally, when an address is loose, the forwarding SHOULD be restrained indicating what type of PCEP peers are targeted (i.e. PCE and/or PCC).

5.4.3. Rules Applied to Existing Errors and Notifications

Many existing normative references states on error definitions (see for instance [RFC5440], [RFC5441],[RFC5455], [RFC5521], [RFC5557], [RFC5886], [RFC8231], [RFC8232],[RFC8253], [RFC8281], [RFC8306], [RFC8408], [RFC8697]). This section provides processing rules for existing error types handling, as a recommendation. According to the definitions provided in this document, the follwoing rules are applicable:

Error-type 1, described in [RFC5440], relates to PCEP session establishement failures. All errors of this type are local and not propagated. Hence, if a "Propagation" TLV is added to the error message it is recommended to be set to value 0. Error-values 1,2,6,7 have a high level of criticality. Hence, if the "Error-criticality" TLV is included within a PCErr message of type 1 and value 1,2,6 or 7, it is recommended to have a value of 2.

Error-type 2,3,4, "Capability not supported", "Unknown object" and "Not supported object" respectively, described in [RFC5440]: errors of this type MAY be propagated using the TLV "Propagation". Their level of criticality is defined as leading to cancel the path computation request [RFC5440]. Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. The error-value 4 of error-type 4 ("Unsupported parameter") associated to the BRPC procedure [RFC5441] is suggested to contain the "Propagation" TLV with a DIFFUSION-LIST requesting a propagation to the PCC at the origin of the request.

Error-type 5 refers to "Policy violation", error values for this type have been defined in [RFC5440], [RFC5541], [RFC5557], [RFC5886] and [RFC8306]. In [RFC5440], it is specified that the path computation request MUST be cancelled when an error of type 5 occurs. Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. As such errors might be conveyed to several PCEs, the "Propagation" TLV MAY be used.

Error-type 6 described as "Mandatory object missing" in [RFC5440], leads to the cancellation of the path computation request. Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. The "Propagation" TLV MAY be used with such errors. The error-value of 4 for Monitoring object missing defined in [RFC5886] is no exception to the rule.

Error-type 7 is described as "synchronized path computation request missing". In [RFC5440], it is specified that the reffered synchronized path computation request MUST be cancelled when an error of type 5 occurs. Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. The "Propagation" TLV MAY be used with such errors.

Error-type 8 is raised when a PCE receives a PCRep with an unknown request reference. If the "Propagation" TLV is used with error-type 8, it is recommended to be set at a value of 0. The "Error-criticality" TLV is not particularly relevant for error-type 8. Hence, it usually have the value of 0 if used.

Error-type 9 is raised when a PCE attempts to establish a second PCEP session. The existing session must be preserved. Hence, if the "Error-criticality" TLV is included, it usually have a value of 0. By definition, such an error message SHOULD NOT be propagated. Thus, if the "Propagation" TLV is used with error-type 9, it is usually set to a value of 0.

Error-type 10 which refers to the reception of an invalid object as described in [RFC5440] no indication is provided on the cancellation of the path computation request. Hence, if the "Error-criticality" TLV is included, it usually have a value of 0. The "Propagation" TLV MAY be used with such errors with any value depending on the expected behavior.

Error-type 11 relates to "Unrecognized EXRS subobject" and is described in [RFC5521]. No path computation request cancellation is required by [RFC5521]. Hence, if the "Error-criticality" TLV is included, it usually have a value of 0. The "Propagation" TLV MAY be used with such errors with any value depending on the expected behavior.

Error-type 12 refers to "Diffserv-aware TE error" and is described in [RFC5455]. Such errors are raised when the CLASSTYPE object of a PCReq is recognized but not supported by a PCE. [RFC5455] does not state about the path computation request when such errors are met. Hence, both "Propagation" and "Error-criticality" TLVs COULD be used within such error-types’ messages and set to any specified values.

Error-type 13 on "BRPC procedure completion failure" is described in [RFC5441]. [RFC5441] states that in such cases, the PCErr message MUST be relayed to the PCC. Hence, such messages SHOULD contain a "Propagation" TLV and a DIFFUSION-LIST with a Target-Type of 0 and corresponding addresses or with a Target-Type of 2. It is not specified in [RFC5441] whether the path computation request should be canceled or not. If the procedure is not supported, it does not necessarily imply to cancel the path computation request if another procedure is able to read and write VSPT objects. Thus, the "Error-criticality" TLV MAY be used with any value depending on the expected behavior.

Error-type 15 refers to "Global Concurrent Optimization Error" defined in [RFC5557]. [RFC5557] states that the corresponding global concurrent path optimization MUST be cancelled at the PCC.

Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. The "Propagation" TLV MAY be used with such errors.

Error-type 16 relates to "P2MP Capability Error" defined in [RFC8306]. Such errors lead to the cancellation of the path computation request. Hence, if the "Error-criticality" TLV is included, it usually have a value of 1. The "Propagation" TLV MAY be used with such errors.

Error-type 17, titled "P2MP END-POINTS Error" is defined [RFC8306]. Such errors are thrown when a PCE tries to add or prune nodes to or from a P2MP Tree. [RFC8306] does not specify if such errors lead to cancel the path computation request. Hence, the "Error-criticality" and "Propagation" TLVs MAY be used with this type of error with any value depending on the expected behavior.

Error-type 18 of "P2MP Fragmentation Error" is described [RFC8306] which does not specify whether the path computation request should be cancelled. But, as messages are fragmented, it is natural to think that the PCE should wait at least a bit for further messages. The "Error-criticality" TLV MAY be included in such error messages and is particularly adapted to differ the semantic of the same error-type message: if it is included with a value of 0 then the PCE will still wait for further fragmented messages, when this waiting time ends, the TLV can be included with a value of 1 in order to finally cancel the request. The "Propagation" TLV MAY also be used with such errors.

Error-type 19 of "Invalid Operation" is described in [RFC8231] and [RFC8281], which implies a wrong capability description for PCEP session. In this case, the PCErr message MUST be returned to PCC, and this message usually contain a "Propagation" TLV and a DIFFUSION-LIST with a Target-Type of 0 or 2. The "Error-criticality" TLV is recommended be set to 2 in order to guanrantee the termination of PCEP session.

Error-type 20 of "LSP State Synchronization Error" is described in [RFC8231] and [RFC8232], which cannot successfully sync up the LSP states. In this case, the "Error-criticality" TLV should be set to 2 in order to guanrantee the termination of PCEP session. The "Propagation" TLV MAY also be used with such errors.

Error-type 21 of "Invalid traffic engineering path setup type" is described in [RFC8408] . Such errors failed to find a matched path setup type and the PCEP sessions MUST be closed. In this case, the "Error-criticality" TLV is usually set to 2 in order to guanrantee the termination of PCEP session. The "Propagation" TLV MAY also be used with such errors.

Error-type 23 of "Bad parameter value" is described in [RFC8281] . Such errors occur when there is a conflict in path name of C flag not set for PCE initiation. In this case, the "Error-criticality" TLV may be set to either 0 or 1 to indicate whether the request is still valid, with the PCEP session open. The "Propagation" TLV MAY also be used with such errors.

Error-type 24 of "LSP instantiation error" is described in [RFC8281] . Such errors occur when PCC detects problems when establishing the path, the message MUST relay to the PCE, therefore the "Propogation" TLV is usually contained. The "Error-criticality" TLV may be set to either 0 or 1 to indicate whether the request is still valid, with the PCEP session open.

Error-type 25 of "PCEP StartTLS failure" is described in [RFC8253]. Such errors indicate the security issue in transport layer. In this case, the "Error-criticality" TLV is usually set to 2 in order to close the PCEP session. The "Propagation" TLV MAY also be used with such errors, depending on the detailed security conditions.

Error-type 26 of "Association Error " is described in [RFC8697] . Such errors occur when there is problem for LSP association. In this case, the "Error-criticality" TLV should be set to either 0 or 1 to indicate whether the request is still valid, with the PCEP session open. The "Propagation" TLV MAY also be used with such errors.

6. Error Handling Guidelines for Future PCEP Extension

Error and Notification handling in this document should be considered in PCE documents that include new errors and notifications. A requirement for the authors of these drafts is to evaluate the applicability of the procedure in this document and provide details about the "Error-criticality" TLV and "Propagation" TLV for errors and notifications defined in the draft. Example text is provided as follow.

Error-type XX (fill in value of the Error-type) of " XXXX " (fill in name of the Error-type) is described in [RFCYYYY] (fill in the document reference of the Error-type). Such errors occur when ZZZZ (fill in typical scenario). In this case, the "Error-criticality" TLV should be set to X (fill in the recommended value) to indicate whether the request is still valid, with the PCEP session open. The error messages SHOULD/MAY (select the mandatory level) contain a

"Propagation" TLV and a DIFFUSION-LIST with a Target-Type of A(fill in the recommended value).

7. Backward Compatibility Consideration

There would be backward compatibility issue if there are multiple PCEs with different level understanding of error message. In a scenario that PCE(i) propagate the error message to PCE (i+1), it is possible that PCE (i+1) is not capable to extract the message correctly, then such error message would be ignored and not be further propagated.

There can be potential approach to avoid these problem, such as recognizing the incapable PCE and avoiding propagation. However, these approach is not in the scope of this document.

8. Implementation Status

[NOTE TO RFC EDITOR : This whole section and the reference to [RFC7942] is to be removed before publication as an RFC]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

At the time of posting the -08 version of this document, there are no known implementations of this mechanism. It is believed that two vendors are considering prototype implementations, but these plans are too vague to make any further assertions.

9. Security Considerations

Within the introduced set of TLVs, the "Propagation" TLV affects PCEP security considerations since it forces propagation behaviors. Thus, a PCEP implementation SHOULD activate stateful mechanism when receiving PCEP-ERROR or NOTIFICATION object including this TLV in order to avoid DoS attacks.

10. IANA Considerations

IANA maintains a registry of PCEP parameters. This includes a sub-registry for PCEP Objects.

IANA is requested to make an allocation from the sub-registry as follows. The values here are suggested for use by IANA.

10.1. PCEP TLV Type Indicators

As described in Section 5.4 the newly defined TLVs allows a PCE to enforce specific error and notification behaviors within PCEP-ERROR and NOTIFICATION objects. IANA is requested to make the following allocations from the "PCEP TLV Type Indicators" sub-registry.

Value	Description	Reference
TBD	Propagation	this document
TBD	Error-criticality	this document

10.2. New DIFFUSION-LIST TLV

Type Value	Meaning	Reference
0	Any PCEP peers	this document
1	PCEs but excludes PCC-only peers	this document
2	PCEs and PCCs with which a session is still opened	this document

Subobjects	Reference
1: IPv4 prefix	this document
2: IPv6 prefix	this document
4: Unnumbered Interface ID	this document
5: 4-byte AS number	this document
6 OSPF area ID	this document
7 IS-IS Area ID	this document
32: Autonomous system number	this document
33: Explicit Exclusion Route subobject (EXRS)	this document

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>.

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

[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, <https://www.rfc-editor.org/info/rfc5441>.

[RFC5455] Sivabalan, S., Ed., Parker, J., Boutros, S., and K. Kumaki, "Diffserv-Aware Class-Type Object for the Path Computation Element Communication Protocol", RFC 5455, DOI 10.17487/RFC5455, March 2009, <https://www.rfc-editor.org/info/rfc5455>.

[RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, DOI 10.17487/RFC5521, April 2009, <https://www.rfc-editor.org/info/rfc5521>.

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

[RFC5557] Lee, Y., Le Roux, JL., King, D., and E. Oki, "Path Computation Element Communication Protocol (PCEP) Requirements and Protocol Extensions in Support of Global Concurrent Optimization", RFC 5557, DOI 10.17487/RFC5557, July 2009, <https://www.rfc-editor.org/info/rfc5557>.

[RFC5886] Vasseur, JP., Ed., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path Computation Element (PCE)-Based Architecture", RFC 5886, DOI 10.17487/RFC5886, June 2010, <https://www.rfc-editor.org/info/rfc5886>.

[RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, DOI 10.17487/RFC8231, September 2017, <https://www.rfc-editor.org/info/rfc8231>.

[RFC8232] Crabbe, E., Minei, I., Medved, J., Varga, R., Zhang, X., and D. Dhody, "Optimizations of Label Switched Path State Synchronization Procedures for a Stateful PCE", RFC 8232, DOI 10.17487/RFC8232, September 2017, <https://www.rfc-editor.org/info/rfc8232>.

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <https://www.rfc-editor.org/info/rfc8253>.

[RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <https://www.rfc-editor.org/info/rfc8281>.

[RFC8306] Zhao, Q., Dhody, D., Ed., Palleti, R., and D. King, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths", RFC 8306, DOI 10.17487/RFC8306, November 2017, <https://www.rfc-editor.org/info/rfc8306>.

[RFC8408] Sivabalan, S., Tantsura, J., Minei, I., Varga, R., and J. Hardwick, "Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages", RFC 8408, DOI 10.17487/RFC8408, July 2018, <https://www.rfc-editor.org/info/rfc8408>.

[RFC8697] Minei, I., Crabbe, E., Sivabalan, S., Ananthakrishnan, H., Dhody, D., and Y. Tanaka, "Path Computation Element Communication Protocol (PCEP) Extensions for Establishing Relationships between Sets of Label Switched Paths (LSPs)", RFC 8697, DOI 10.17487/RFC8697, January 2020, <https://www.rfc-editor.org/info/rfc8697>.

11.2. Informational References

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

[RFC7470] Zhang, F. and A. Farrel, "Conveying Vendor-Specific Constraints in the Path Computation Element Communication Protocol", RFC 7470, DOI 10.17487/RFC7470, March 2015, <https://www.rfc-editor.org/info/rfc7470>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[RFC8751] Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., and D. King, "Hierarchical Stateful Path Computation Element (PCE)", RFC 8751, DOI 10.17487/RFC8751, March 2020, <https://www.rfc-editor.org/info/rfc8751>.

Appendix A. Error and Notification Scenarios

This section provides some examples depicting how the error described above can be used in a PCEP session. The origin of the errors or notifications is only illustrative and has no normative purpose. Sometimes the PCE features behind may be implementation-specific (e.g. detection of flooding). This section does not provide scenarios for errors with a high-level of critcity (i.e., Error behaviors 3 and 6) since such errors are very specific and until now have been normalized only during the session establishment (error-type of 1).

A.1. Error Behavior Type 1

In this example, a PCC attempts to establish a second PCEP session with the same PCE for another request. Consequently the PCE sends back an error message with error-type 9. This error stays local and does not affect the former session. The second session is ignored. If the "Propagation" TLV and "Error-criticality" TLV are used, they should be both set to value 0.

```
                     +-+-+                   +-+-+
                     |PCC|                   |PCE|
                     +-+-+                   +-+-+
1) Path computation    |                       |
 event                 |                       |
2) PCE selection       |----- Open Message--->|
                       |<--- Open message ----|
3) Path computation    |---- PCReq message--->|
 request X sent to     |                      |4) Path computation
 the selected PCE      |                      | request queued
                       |                      |
5) Path computation    |                      |
 event                 |                      |
6) PCE selection       |                      |
                       |----- Open Message--->|8) Session already
                       |                      |opened
                       |<--- PCErr message----| Error-type=9
                       |                      |
```

A.2. Error Behavior Type 2

In this example, the PCC sends a DiffServ-aware path computation request. If the PCE receiving the request does not support the indicated class-type, it thus sends back a PCErr message with error-type=12 and error-value=1. If the "Propagation" TLV and "Error-criticality" TLV are present, they should carry value 0 and value 1 respectively. Consequently, the request is cancelled.

```
                     +-+-+                   +-+-+
                     |PCC|                   |PCE|
                     +-+-+                   +-+-+
   1) Path computation |                       |
    event              |                       |
   2) PCE selection    |                       |
   3) Path computation |---- PCReq message--->|
    request X sent to  |                       |4) Path computation
    the selected PCE   |                       | request queued
                       |                       |
                       |                       |5) DiffServ class-type
                       |                       | not supported
                       |                       |6) Path computation
                       |                       | request X
                       |                       | cancelled
                       |<--- PCErr message----| Error-type=12
                       |                       |
```

A.3. Error Behavior Type 4

In this example, a PCC sends a path computation requests with no P flag set (e.g. END-POINT object with P-flag cleared). This is detected by another PCE in the sequence. The path computation request can thus be treated but the P-Flag will be ignored. Hence, this error is not critical but the source PCC should be informed of this fact. So, a PCErr message with error-type 10 ("Reception of an invalid object"). The PCEP-ERROR object of the message contains a "Propagation" TLV at value 1 and a "Error-criticality" TLV at value 0. It is hence propagated backwardly to the source PCC.

```
    +-+-+              +-+-+-+-+              +-+-+
    |PCC|              |PCE|PCC|              |PCE|
    +-+-+              +-+-+-+-+              +-+-+
      |---- PCReq message-->|                       |
      |                     |                       |
      |                     |---- PCReq message--->|
      |                     |                       |
      |                     |                       |1) Parameter is
      |                     |                       | not supported
      |                     |                       |
      |                     |<--- PCErr message----| Error-type=10
      |<--- PCErr message---|                       |
      |                     |                       |
```

A.4. Error Behavior Type 5

In this example, PCEs are using the BRPC procedure to treat a path computation request [RFC5441]. However, one of the PCEs does not support a parameter of the request. Hence, a PCErr message with error-type 4 and error-value 4 is sent by this PCE and has to be forwarded to the source PCC. The PCEP-ERROR object includes a "Propagation" TLV at value 1 and "Error-criticality" TLV at value 1 and the message is propagated backwardly to the source PCC. Consequently, the request is cancelled.

```
 +-+-+               +-+-+-+-+                +-+-+
 |PCC|               |PCE|PCC|                |PCE|
 +-+-+               +-+-+-+-+                +-+-+
   |---- PCReq message-->|                      |
   |                     |                      |
   |                     |---- PCReq message--->|
   |                     |                      |
   |                     |                      |1) Unsupported
   |                     |                      | Parameter BRPC
   |                     |                      |2) Path
   |                     |                      | computation
   |                     |                      | request X
   |                     |                      | cancelled
   |                     |<--- PCErr message----| Error-type=4
   |<--- PCErr message---|                      |
   |                     |                      |
```

Authors' Addresses

Helia Pouyllau
Alcatel-Lucent
Route de Villejust
NOZAY 91620
FRANCE

Phone: + 33 (0)1 30 77 63 11
Email: helia.pouyllau@alcatel-lucent.com

Remi Theillaud
Marben Products
176 rue Jean Jaures
Puteaux 92800
FRANCE

Phone: + 33 (0)1 79 62 10 22
Email: remi.theillaud@marben-products.com

Julien Meuric
Orange
2, avenue Pierre Marzin
Lannion 22307
FRANCE

Email: julien.meuric@orange.com

Haomian Zheng (Editor)
Huawei Technologies
H1, Xiliu Beipo Village, Songshan Lake,
Dongguan, Guangdong 523808
China

Email: zhenghaomian@huawei.com

Xian Zhang
Huawei Technologies
A10, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: zhang.xian@huawei.com

PCE Working Group F. Zhang
Internet-Draft Q. Zhao
Intended status: Standards Track Huawei
Expires: December 2, 2019 O. Gonzalez de Dios
Telefonica I+D
R. Casellas
CTTC
D. King
Old Dog Consulting
June 1, 2019

Extensions to Path Computation Element Communication Protocol (PCEP) for Hierarchical Path Computation Elements (PCE)

draft-ietf-pce-hierarchy-extensions-11

Abstract

The Hierarchical Path Computation Element (H-PCE) architecture is defined in RFC 6805. It provides a mechanism to derive an optimum end-to-end path in a multi-domain environment by using a hierarchical relationship between domains to select the optimum sequence of domains and optimum paths across those domains.

This document defines extensions to the Path Computation Element Protocol (PCEP) to support Hierarchical PCE procedures.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Path Computation Element communication Protocol (PCEP) provides a mechanism for Path Computation Elements (PCEs) and Path Computation Clients (PCCs) to exchange requests for path computation and responses that provide computed paths.

The capability to compute the routes of end-to-end inter-domain MPLS Traffic Engineering (MPLS-TE) and GMPLS Label Switched Paths (LSPs) is expressed as requirements in [RFC4105] and [RFC4216]. This capability may be realized by a PCE [RFC4655]. The methods for establishing and controlling inter-domain MPLS-TE and GMPLS LSPs are documented in [RFC4726].

[RFC6805] describes a Hierarchical PCE (H-PCE) architecture which can be used for computing end-to-end paths for inter-domain MPLS Traffic Engineering (TE) and GMPLS Label Switched Paths (LSPs).

Within the hierarchical PCE architecture, the parent PCE is used to compute a multi-domain path based on the domain connectivity information. A child PCE may be responsible for single or multiple domains and is used to compute the intra-domain path based on its own domain topology information.

The H-PCE end-to-end domain path computation procedure is described below:

- o A path computation client (PCC) sends the inter-domain path computation requests to the child PCE responsible for its domain;
- o The child PCE forwards the request to the parent PCE;
- o The parent PCE computes the likely domain paths from the ingress domain to the egress domain;
- o The parent PCE sends the intra-domain path computation requests (between the domain border nodes) to the child PCEs which are responsible for the domains along the domain path;
- o The child PCEs return the intra-domain paths to the parent PCE;
- o The parent PCE constructs the end-to-end inter-domain path based on the intra-domain paths;
- o The parent PCE returns the inter-domain path to the child PCE;
- o The child PCE forwards the inter-domain path to the PCC.

The parent PCE may be requested to provide only the sequence of domains to a child PCE so that alternative inter-domain path computation procedures, including Per Domain (PD) [RFC5152] and Backwards Recursive Path Computation (BRPC) [RFC5441], may be used.

This document defines the PCEP extensions for the purpose of implementing Hierarchical PCE procedures, which are described in [RFC6805].

1.1. Scope

The following functions are out of scope of this document:

- o Determination of Destination Domain (section 4.5 of [RFC6805]):
  - * via a collection of reachability information from child domain;
  - * via requests to the child PCEs to discover if they contain the

destination node;

* or any other methods.

o Parent Traffic Engineering Database (TED) methods (section 4.4 of [RFC6805]), although suitable mechanisms include:

  * YANG-based management interfaces;

  * BGP-LS [RFC7752];

  * Future extension to PCEP (such as [I-D.dhodylee-pce-pcep-ls]).

o Learning of Domain connectivity and boundary nodes (BN) addresses, methods to achieve this function include:

  * YANG-based management interfaces;

  * BGP-LS [RFC7752];

  * Future extension to PCEP (such as [I-D.dhodylee-pce-pcep-ls]).

o Stateful PCE Operations (Refer [I-D.ietf-pce-stateful-hpce])

o Applicability of hierarchical PCE to large multi-domain environments.

  * The hierarchical relationship model is described in [RFC6805]. It is applicable to environments with small groups of domains where visibility from the ingress LSRs is limited. As highlighted in [RFC7399] applying the hierarchical PCE model to very large groups of domains, such as the Internet, is not considered feasible or desirable.

1.2. Terminology

This document uses the terminology defined in [RFC4655], [RFC5440] and the additional terms defined in Section 1.4 of [RFC6805].

1.3. 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. Requirements for H-PCE

This section compiles the set of requirements to the PCEP extensions to support the H-PCE architecture and procedures. [RFC6805] identifies high-level requirements of PCEP extensions required to support the hierarchical PCE model.

2.1. Path Computation Request

The Path Computation Request (PCReq) [RFC5440] messages are used by a PCC or a PCE to make a path computation request to a PCE. In order to achieve the full functionality of the H-PCE procedures, the PCReq message needs to include:

- Qualification of PCE Requests (Section 4.8.1. of [RFC6805]);
- Multi-domain Objective Functions (OF);
- Multi-domain Metrics.

2.1.1. Qualification of PCEP Requests

As described in Section 4.8.1 of [RFC6805], the H-PCE architecture introduces new request qualifications, which are:

- The ability for a child PCE to indicate that a path computation request sent to a parent PCE should be satisfied by a domain sequence only, that is, not by a full end-to-end path. This allows the child PCE to initiate a per-domain (PD) [RFC5152] or a backward recursive path computation (BRPC) [RFC5441].
- As stated in [RFC6805], Section 4.5, if a PCC knows the egress domain, it can supply this information as the path computation request. The PCC may also want to specify the destination domain information in a PCEP request, if it is known.
- An inter domain path computed by parent PCE should be capable of disallowing specific domain re-entry.

2.1.2. Multi-domain Objective Functions

For H-PCE inter-domain path computation, there are three new Objective Functions defined in this document:

- Minimize the number of Transit Domains (MTD)
- Minimize the number of border nodes (MBN)
- Minimize the number of Common Transit Domains (MCTD)

The PCC may specify the multi-domain Objective Function code to use when requesting inter-domain path computation, it may also include intra-domain OFs, such as Minimum Cost Path (MCP) [RFC5441], which must be considered by participating child PCEs.

2.1.3. Multi-domain Metrics

For inter-domain path computation, there are several path metrics of interest.

- o Domain count (number of domains crossed);

- o Border Node count.

A PCC may be able to limit the number of domains crossed by applying a limit on these metrics. Details in Section 3.4.

2.2. Parent PCE Capability Advertisement

A PCEP Speaker (Parent PCE or Child PCE) that supports and wishes to use the procedures described in this document must advertise the fact and negotiate its role with its PCEP peers. It does this using the "H-PCE Capability" TLV, described in Section 3.2.1, in the OPEN Object to advertise its support for PCEP extensions for H-PCE Capability.

During the PCEP session establishment procedure, the child PCE needs to be capable of indicating to the parent PCE whether it requests the parent PCE capability or not.

2.3. PCE Domain Identification

A PCE domain is a single domain with an associated PCE. Although it is possible for a PCE to manage multiple domains simultaneously. The PCE domain could be an IGP area or AS.

The PCE domain identifiers MAY be provided during the PCEP session establishment procedure.

2.4. Domain Diversity

In a multi-domain environment, Domain Diversity is defined in [RFC6805] and described as "A pair of paths are domain-diverse if they do not transit any of the same domains. A pair of paths that share a common ingress and egress are domain-diverse if they only share the same domains at the ingress and egress (the ingress and egress domains). Domain diversity may be maximized for a pair of paths by selecting paths that have the smallest number of shared domains."

The main motivation behind domain diversity is to avoid fate sharing, but it can also be because of some geo-political reasons and commercial relationships that would require domain diversity. For example, a pair of paths should choose different transit Autonomous System (AS) because of some policy considerations.

In the case when full domain diversity could not be achieved, it is helpful to minimize the commonly shared domains. Also, it is interesting to note that other scope of diversity (node, link, SRLG etc.) can still be applied inside the commonly shared domains.

3. PCEP Extensions

This section defines extensions to PCEP [RFC5440] to support the H-PCE procedures.

3.1 Applicability to PCC-PCE Communications

Although the extensions defined in this document are intended primarily for use between a child PCE and a parent PCE, they are also applicable for communications between a PCC and its PCE.

Thus, the information that may be encoded in a PCReq can be sent from a PCC towards the child PCE. This includes the RP object (Section 3.3) and the Objective Function (OF) codes and objects (Section 3.4). A PCC and a child PCE could also exchange the capability (Section 3.2.1) during its session.

This allows a PCC to request paths that transit multiple domains utilizing the capabilities defined in this document.

3.2. OPEN Object

Two new TLVs are defined in this document to be carried within an OPEN object. This way, during the PCEP session establishment, the H-PCE capability and Domain information can be advertised.

3.2.1. H-PCE Capability TLV

The H-PCE-CAPABILITY TLV is an optional TLV associated with the OPEN Object [RFC5440] to exchange H-PCE capability of PCEP speakers.

Its format is shown in the following figure:

```
 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= TBD1        |            Length=4         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Flags                            |P|
+---------------------------------------------------------------+
```

Figure 1: H-PCE-CAPABILITY TLV format

The type of the TLV is TBD1 (to be assigned by IANA), and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits):

P (Parent PCE Request bit): if set, will signal that the child PCE wishes to use the peer PCE as a parent PCE.

Unassigned bits MUST be set to 0 on transmission and MUST be ignored on receipt.

The inclusion of this TLV in an OPEN object indicates that the H-PCE extensions are supported by the PCEP speaker. The child PCE MUST include this TLV and set the P flag. The parent PCE MUST include this TLV and unset the P flag.

The setting of the P flag (parent PCE request bit) would mean that the PCEP speaker wants the peer to be a parent PCE, so in the case of a PCC to Child-PCE relationship, neither entity would set the P flag.

If both peers attempt to set the P flag then the session establishment MUST fail, and the PCEP speaker MUST respond with PCErr message using Error-Type 1: "PCEP Session Establishment Failure" as per [RFC5440].

If the PCE understands the H-PCE path computation request but did not advertise its H-PCE capability, it MUST send a PCErr message with Error-Type=TBD8 ("H-PCE error") and Error-Value=1 ("H-PCE Capability not advertised").

3.2.1.1 Backwards Compatibility

Section 7.1 of [RFC5440] requires that "Unrecognized TLVs MUST be ignored.

That means that a PCE that does not support this document but that receives an Open Message containing an Open Object that includes an H-PCE-CAPABILITIES TLV will ignore that TLV and will continue to attempt to establish a PCEP session. It will, however, not include the TLV in the Open message that it sends, so the H-PCE relationship will not be created.

If a PCE does not support the extensions defined in this document but receives them in a PCEP message (notwithstanding the fact that the session was not established as supporting a H-PCE relationship), the receiving PCE will ignore the H-PCE related parameters because they are all encoded in TLVs within standard PCEP objects.

3.2.2. Domain-ID TLV

The Domain-ID TLV, when used in the OPEN object, identifies the domains served by the PCE. The child PCE uses this mechanism to inform the domain information to the parent PCE.

The Domain-ID TLV is defined 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type= TBD2      |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Domain Type   |                 Reserved                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                           Domain ID                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Domain-ID TLV format

The type of the TLV is TBD2 (to be assigned by IANA), and it has a variable Length of the value portion. The value part comprises:

Domain Type (8 bits): Indicates the domain type. Four types of domain are currently defined:

* Type=1: the Domain ID field carries a 2-byte AS number. Padded with trailing zeros to a 4-byte boundary.

* Type=2: the Domain ID field carries a 4-byte AS number.

* Type=3: the Domain ID field carries a 4-byte OSPF area ID.

* Type=4: the Domain ID field carries (2-byte Area-Len, variable length IS-IS area ID). Padded with trailing zeros to a 4-byte

boundary.

Reserved: Zero at transmission; ignored at the receipt.

Domain ID (variable): Indicates an IGP Area ID or AS number as per the Domain Type field. It can be 2 bytes, 4 bytes or variable length depending on the domain identifier used. It is padded with trailing zeros to a 4-byte boundary. In case of IS-IS it includes the Area-Len as well.

In the case a PCE serves more than one domain, multiple Domain-ID TLVs are included for each domain it serves.

3.3. RP Object

3.3.1. H-PCE-FLAG TLV

The H-PCE-FLAG TLV is an optional TLV associated with the RP Object [RFC5440] to indicate the H-PCE path computation request and options.

Its format is shown in the following figure:

```
 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= TBD3      |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Flags                          |D|S|
+---------------------------------------------------------------+
```

Figure 3: H-PCE-FLAG TLV format

The type of the TLV is TBD3 (to be assigned by IANA), and it has a fixed length of 4 octets.

The value comprises a single field - Flags (32 bits):

S (Domain Sequence bit): if set, will signal that the child PCE wishes to get only the domain sequence in the path computation reply. Refer to Section 3.7 of [RFC7897] for details.

D (Disallow Domain Re-entry bit): if set, will signal that the computed path does not enter a domain more than once.

Unassigned bits MUST be set to 0 on transmission and MUST be ignored on receipt.

The presence of the TLV indicates that the H-PCE based path computation is requested as per this document.

3.3.2. Domain-ID TLV

The Domain-ID TLV, carried in an OPEN object, is used to indicate a (list of) managed domains and is described in Section 3.3.1. This TLV, when carried in an RP object, indicates the destination domain ID. If a PCC knows the egress domain, it can supply this information in the PCReq message. The format and procedure of this TLV are defined in Section 3.2.2.

If a Domain-id TLV is used in the RP object, and the destination is not actually in the indicated domain, then the parent PCE should respond with a NO-PATH object and NO-PATH VECTOR TLV should be used. A new bit number is assigned to indicate "Destination not found in the indicated domain" (see Section 3.7).

3.4. Objective Functions

3.4.1. OF Codes

[RFC5541] defines a mechanism to specify an Objective Function that is used by a PCE when it computes a path. Three new Objective Functions are defined for H-PCE, these are:

- o MTD
  - * Name: Minimize the number of Transit Domains (MTD)
  - * Objective Function Code - TBD4 (to be assigned by IANA)
  - * Description: Find a path P such that it passes through the least number of transit domains.
  - * Objective functions are formulated using the following terminology:
    - \+ A network comprises a set of N domains {Di, (i=1...N)}.
    - \+ A path P passes through K unique domains {Dpi,(i=1...K)}.
    - \+ Find a path P such that the value of K is minimized.
- o MBN
  - * Name: Minimize the number of border nodes.
  - * Objective Function Code - TBD5 (to be assigned by IANA)

   * Description: Find a path P such that it passes through the least number of border nodes.

   * Objective functions are formulated using the following terminology:

      + A network comprises a set of N links {Li, (i=1...N)}.

      + A path P is a list of K links {Lpi,(i=1...K)}.

      + D(Lpi) if a function that determines if the links Lpi and Lpi+1 belong to different domains, D(Li) = 1 if link Li and Li+1 belong to different domains, D(Lk) = 0 if link Lk and Lk+1 belong to the same domain.

      + The number of border node in a path P is denoted by B(P), where B(P) = sum{D(Lpi),(i=1...K-1)}.

      + Find a path P such that B(P) is minimized.

There is one objective function that applies to a set of synchronized path computation requests to increase the domain diversity:

o MCTD

   * Name: Minimize the number of Common Transit Domains

   * Objective Function Code - TBD13 (to be assigned by IANA)

   * Description: Find a set of paths such that it passes through the least number of common transit domains.

      + A network comprises a set of N domains {Di, (i=1...N)}.

      + A path P passes through K unique domains {Dpi,(i=1...K)}.

      + A set of paths {P1...Pm} have L transit domains that are common to more than one path {Dpi,(i=1...L)}.

      + Find a set of paths such that the value of L is minimized.

3.4.2. OF Object

The OF (Objective Function) object [RFC5541] is carried within a PCReq message so as to indicate the desired/required objective function to be applied by the PCE during path computation. As per Section 3.2 of [RFC5541] a single OF object may be included in a path computation request.

The new OF codes described in Section 3.4.1 are applicable at the inter-domain path computation performed by the parent PCE, it is also necessary to specify the OF code that may be applied for the intra-domain path computation performed by the child PCE. To accommodate this, the OF-List TLV (described in Section 2.1. of [RFC5541]) is included in the OF object as an optional TLV.

The OF-List TLV allows encoding of multiple OF codes. When this TLV is included inside the OF object, only the first OF-code in the OF-LIST TLV is considered. The parent PCE MUST use this OF code in the OF object when sending the intra domain path computation request to the child PCE. If the OF list TLV is included in the OF Object, the OF Code inside the OF Object MUST include one of the H-PCE Objective Functions defined in this document, the OF Code inside the OF List TLV MUST NOT include an H-PCE Objective Function. If this condition is not met, the PCEP speaker MUST respond with a PCErr message with Error-Type=10 (Reception of an invalid object) and Error-Value=TBD15 (Incompatible OF codes in H-PCE).

If the Objective Functions defined in this document are unknown or unsupported by a PCE, then the procedure as defined in [RFC5541] is followed.

3.5. Metric Object

The METRIC object is defined in Section 7.8 of [RFC5440], comprising of metric-value, metric-type (T field) and flags. This document defines the following types for the METRIC object for H-PCE:

- o T=TBD6: Domain count metric (number of domains crossed);
- o T=TBD7: Border Node count metric (number of border nodes crossed).

The domain count metric type of the METRIC object encodes the number of domains crossed in the path. The border node count metric type of the METRIC object encodes the number of border nodes in the path. If a domain is re-entered, then domain should be double counted.

A PCC or child PCE MAY use the metric in a PCReq message for an inter-domain path computation, meeting the number of domain or border nodes crossing requirement. As per [RFC5440], in this case, the B bit is set to suggest a bound (a maximum) for the metric that must not be exceeded for the PCC to consider the computed path as acceptable.

A PCC or child PCE MAY also use this metric to ask the PCE to optimize the metric during inter-domain path computation. In this case, the B flag is cleared, and the C flag is set.

The Parent PCE MAY use the metric in a PCRep message along with a NO-PATH object in the case where the PCE cannot compute a path meeting this constraint. A PCE MAY also use this metric to send the computed end to end metric value in a reply message.

3.6. SVEC Object

[RFC5440] defines SVEC object which includes flags for the potential dependency between the set of path computation requests (Link, Node and SRLG diverse). This document defines a new flag O for domain diversity.

The following new bit is added to the Flags field:

- o Domain Diverse O-bit - TBD14 : when set, this indicates that the computed paths corresponding to the requests specified by the following RP objects MUST NOT have any transit domains in common.

The Domain Diverse O-bit can be used in Hierarchical PCE path computation to compute synchronized domain diverse end to end path or diverse domain sequences.

When domain diverse O bit is set, it is applied to the transit domains. The other bit in SVEC object (N, L, S etc.) MAY be set and MUST still be applied in the ingress and egress shared domain.

3.7. PCEP-ERROR Object

3.7.1. Hierarchy PCE Error-Type

A new PCEP Error-Type [RFC5440] is used for the H-PCE extension as defined below:

Error-Type	Meaning
TBD8	H-PCE error Error-value=1: H-PCE capability was not advertised Error-value=2: parent PCE capability cannot be provided

Figure 4: H-PCE error

3.8. NO-PATH Object

To communicate the reason(s) for not being able to find a multi-domain path or domain sequence, the NO-PATH object can be used in the PCRep message. [RFC5440] defines the format of the NO-PATH object. The object may contain a NO-PATH-VECTOR TLV to provide additional information about why a path computation has failed.

Three new bit flags are defined to be carried in the Flags field in the NO-PATH-VECTOR TLV carried in the NO-PATH Object.

- o Bit number TBD9: When set, the parent PCE indicates that destination domain unknown;

- o Bit number TBD10: When set, the parent PCE indicates unresponsive child PCE(s);

- o Bit number TBD11: When set, the parent PCE indicates no available resource available in one or more domains.

- o Bit number TBD12: When set, the parent PCE indicates that the destination is not found in the indicated domain.

4. H-PCE Procedures

The H-PCE path computation procedure is described in [RFC6805].

4.1. OPEN Procedure between Child PCE and Parent PCE

If a child PCE wants to use the peer PCE as a parent, it MUST set the P (parent PCE request flag) in the H-PCE-CAPABILITY TLV inside the OPEN object carried in the Open message during the PCEP session initialization procedure.

The child PCE MAY also report its list of domain IDs, to the parent PCE, by specifying them in the Domain-ID TLVs in the OPEN object. This object is carried in the OPEN message during the PCEP session initialization procedure

The OF codes defined in this document can be carried in the OF-list TLV of the OPEN object. If the OF-list TLV carries the OF codes, it means that the PCE is capable of implementing the corresponding objective functions. This information can be used for selecting a proper parent PCE when a child PCE wants to get a path that satisfies a certain Objective Function.

When a child PCE sends a PCReq to a peer PCE, which requires parental activity and H-PCE capability flags TLV but which were not included in the session establishment procedure described above, the peer PCE SHOULD send a PCErr message to the child PCE and MUST specify the error-type=TBD8 (H-PCE error) and error-value=1 (H-PCE capability was not advertised) in the PCEP-ERROR object.

When a specific child PCE sends a PCReq to a peer PCE, that requires parental activity and the peer PCE does not want to act as the parent for it, the peer PCE SHOULD send a PCErr message to the child PCE and MUST specify the error-type=TBD8 (H-PCE error) and error-value=2 (Parent PCE capability cannot be provided) in the PCEP-ERROR object.

4.2. Procedure to Obtain Domain Sequence

If a child PCE only wants to get the domain sequence for a multi-domain path computation from a parent PCE, it can set the Domain Path Request bit in the H-PCE-FLAG TLV in the RP object carried in a PCReq message. The parent PCE which receives the PCReq message tries to compute a domain sequence for it (instead of the E2E path). If the domain path computation succeeds the parent PCE sends a PCRep message which carries the domain sequence in the Explicit Route Object (ERO) to the child PCE. Refer to [RFC7897] for more details about domain sub-objects in the ERO. Otherwise, it sends a PCReq message which carries the NO-PATH object to the child PCE.

5. Error Handling

A PCE that is capable of acting as a parent PCE might not be configured or willing to act as the parent for a specific child PCE. This fact could be determined when the child sends a PCReq that requires parental activity, and could result in a negative response in a PCEP Error (PCErr) message and indicate the hierarchy PCE error-type=TBD8 (H-PCE error) and suitable error-value. (Section 3.7)

Additionally, the parent PCE may fail to find the multi-domain path or domain sequence due to one or more of the following reasons:

- o A child PCE cannot find a suitable path to the egress;
- o The parent PCE does not hear from a child PCE for a specified time;
- o The Objective Functions specified in the path request cannot be met.

In this case, the parent PCE MAY need to send a negative path computation reply specifying the reason. This can be achieved by including NO-PATH object in the PCRep message. Extension to NO-PATH object is needed to include the aforementioned reasons described in Section 3.7.

6. Manageability Considerations

General PCE and PCEP management considerations are discussed in [RFC4655] and [RFC5440]. There are additional management considerations for H-PCE which are described in [RFC6805], and repeated in this section.

The administrative entity responsible for the management of the parent PCEs must be determined for the following cases:

o multi-domains (e.g., IGP areas or multiple ASes) within a single service provider network, the management responsibility for the parent PCE would most likely be handled by the service provider,

o multiple ASes within different service provider networks, it may be necessary for a third party to manage the parent PCEs according to commercial and policy agreements from each of the participating service providers.

6.1. Control of Function and Policy

Control and function will need to be carefully managed in an H-PCE network. A child PCE will need to be configured with the address of its parent PCE. It is expected that there will only be one or two parents of any child.

The parent PCE also needs to be aware of the child PCEs for all child domains that it can see. This information is most likely to be configured (as part of the administrative definition of each domain).

Discovery of the relationships between parent PCEs and child PCEs do not form part of the hierarchical PCE architecture. Mechanisms that rely on advertising or querying PCE locations across domain or provider boundaries are undesirable for security, scaling, commercial, and confidentiality reasons. The specific behaviour of the child and parent PCE are described in the following sub-sections.

6.1.1. Child PCE

Support of the hierarchical procedure will be controlled by the management organization responsible for each child PCE. A child PCE must be configured with the address of its parent PCE in order for it to interact with its parent PCE. The child PCE must also be authorized to peer with the parent PCE.

6.1.2. Parent PCE

The parent PCE MUST only accept path computation requests from authorized child PCEs. If a parent PCE receives requests from an unauthorized child PCE, the request SHOULD be dropped. This means that a parent PCE MUST be able to cryptographically authenticate requests from child PCEs.

Multi-party shared key authentication schemes are not recommended for inter-domain relationships because of the potential for impersonation and repudiation and for the operational difficulties should revocation be required.

The choice of authentication schemes to employ may be left to implementers of H-PCE and are not discussed further in this document.

6.1.3. Policy Control

It may be necessary to maintain a policy module on the parent PCE [RFC5394]. This would allow the parent PCE to apply commercially relevant constraints such as SLAs, security, peering preferences, and monetary costs.

It may also be necessary for the parent PCE to limit the end-to-end path selection by including or excluding specific domains based on commercial relationships, security implications, and reliability.

6.2. Information and Data Models

A MIB module for PCEP was published as RFC 7420 [RFC7420] that describes managed objects for modelling of PCEP communication. A YANG module for PCEP has also been proposed [I-D.ietf-pce-pcep-yang].

Additionally, H-PCE MIB module, or additional data model, will be required to report parent PCE and child PCE information, including:

- o parent PCE configuration and status,
- o child PCE configuration and information,
- o notifications to indicate session changes between parent PCEs and child PCEs, and
- o notification of parent PCE TED updates and changes.

6.3. Liveness Detection and Monitoring

The hierarchical procedure requires interaction with multiple PCEs. Once a child PCE requests an end-to-end path, a sequence of events occurs that requires interaction between the parent PCE and each child PCE. If a child PCE is not operational, and an alternate transit domain is not available, then the failure must be reported.

6.4. Verify Correct Operations

Verifying the correct operation of a parent PCE can be performed by monitoring a set of parameters. The parent PCE implementation should provide the following parameters monitored at the parent PCE:

- o number of child PCE requests,
- o number of successful hierarchical PCE procedures completions on a per-PCE-peer basis,
- o number of hierarchical PCE procedure completion failures on a per-PCE-peer basis, and
- o number of hierarchical PCE procedure requests from unauthorized child PCEs.

6.5. Requirements On Other Protocols

Mechanisms defined in this document do not imply any new requirements on other protocols.

6.6. Impact On Network Operations

The hierarchical PCE procedure is a multiple-PCE path computation scheme. Subsequent requests to and from the child and parent PCEs do not differ from other path computation requests and should not have any significant impact on network operations.

7. IANA Considerations

IANA maintains the "Path Computation Element Protocol (PCEP) Numbers" registry. This document requests IANA actions to allocate code points for the protocol elements defined in this document.

7.1. PCEP TLV Type Indicators

IANA Manages the PCEP TLV code point registry (see [RFC5440]). This is maintained as the "PCEP TLV Type Indicators" sub-registry of the

"Path Computation Element Protocol (PCEP) Numbers" registry.

This document defines three new PCEP TLVs. IANA is requested to make the following allocation:

Type	TLV name	References
TBD1	H-PCE-CAPABILITY TLV	This I-D
TBD2	Domain-ID TLV	This I-D
TBD3	H-PCE-FLAG TLV	This I-D

7.2. H-PCE-CAPABILITY TLV Flags

This document requests that a new sub-registry, named "H-PCE-CAPABILITY TLV Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field in the H-PCE-CAPABILITY TLV of the PCEP OPEN object.

New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

o Bit number (counting from bit 0 as the most significant bit)

o Capability description

o Defining RFC

The following values are defined in this document:

Bit	Description	Reference
31	P (Parent PCE Request bit)	This I.D.

7.3. Domain-ID TLV Domain type

This document requests that a new sub-registry, named "Domain-ID TLV Domain type", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Domain-Type field of the Domain-ID TLV. The allocation policy for this new sub-registry is IETF Review [RFC8126].

Value	Meaning
1	2-byte AS number
2	4-byte AS number
3	4-byte OSPF area ID
4	Variable length IS-IS area ID

7.4. H-PCE-FLAG TLV Flags

This document requests that a new sub-registry, named "H-PCE-FLAG TLV Flag Field", is created within the "Path Computation Element Protocol (PCEP) Numbers" registry to manage the Flag field in the H-PCE-FLAGS TLV of the PCEP RP object. New values are to be assigned by Standards Action [RFC8126]. Each bit should be tracked with the following qualities:

- o Bit number (counting from bit 0 as the most significant bit)
- o Capability description
- o Defining RFC

The following values are defined in this document:

Bit	Description	Reference
31	S (Domain Sequence bit)	This I.D.
30	D (Disallow Domain Re-entry bit)	This I.D.

7.5. OF Codes

IANA maintains a registry of Objective Function (described in [RFC5541]) at the sub-registry "Objective Function". Three new Objective Functions have been defined in this document.

IANA is requested to make the following allocations:

Code Point	Name	Reference
TBD4	Minimum number of Transit Domains (MTD)	This I.D.
TBD5	Minimize number of Border Nodes (MBN)	This I.D.
TBD13	Minimize the number of Common Transit Domains (MCTD)	This I.D.

7.6. METRIC Types

IANA maintains one sub-registry for "METRIC object T field". Two new metric types are defined in this document for the METRIC object (specified in [RFC5440]).

IANA is requested to make the following allocations:

Value	Description	Reference
TBD6	Domain Count metric	This I.D.
TBD7	Border Node Count metric	This I.D.

7.7. New PCEP Error-Types and Values

IANA maintains a registry of Error-Types and Error-values for use in PCEP messages. This is maintained as the "PCEP-ERROR Object Error Types and Values" sub-registry of the "Path Computation Element Protocol (PCEP) Numbers" registry.

IANA is requested to make the following allocations:

Error-Type	Meaning and error values	Reference
TBD8	H-PCE Error	This I.D.
	Error-value=1 H-PCE Capability not advertised	
	Error-value=2 Parent PCE Capability cannot be provided	
10	Reception of an invalid object	[RFC5440]
	Error-value=TBD15: Incompatible OF codes in H-PCE	This I.D.

7.8. New NO-PATH-VECTOR TLV Bit Flag

IANA maintains a sub-registry "NO-PATH-VECTOR TLV Flag Field" of bit flags carried in the PCEP NO-PATH-VECTOR TLV in the PCEP NO-PATH object as defined in [RFC5440]. IANA is requested to assign three new bit flag as follows:

Bit Number	Name Flag	Reference
TBD9	Destination Domain unknown	This I.D.
TBD10	Unresponsive child PCE(s)	This I.D.
TBD11	No available resource in one or more domain	This I.D.
TBD12	Destination is not found in the indicated domain.	This I.D.

7.9. SVEC Flag

IANA maintains a sub-registry "SVEC Object Flag Field" of bit flags carried in the PCEP SVEC object as defined in [RFC5440]. IANA is requested to assign one new bit flag as follows:

Bit Number	Name Flag	Reference
TBD14	Domain Diverse O-bit	This I.D.

8. Security Considerations

The hierarchical PCE procedure relies on PCEP and inherits the security considerations defined in [RFC5440]. As PCEP operates over TCP, it may also make use of TCP security mechanisms, such as TCP Authentication Option (TCP-AO) [RFC5925] or Transport Layer Security (TLS) [RFC8253].

Any multi-domain operation necessarily involves the exchange of information across domain boundaries. This may represent a significant security and confidentiality risk especially when the child domains are controlled by different commercial concerns. PCEP allows individual PCEs to maintain the confidentiality of their domain path information using path-keys [RFC5520], and the H-PCE architecture is specifically designed to enable as much isolation of domain topology and capabilities information as is possible.

For further considerations of the security issues related to inter-AS path computation, see [RFC5376].

9. Contributing Authors

Xian Zhang
Huawei
EMail: zhang.xian@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

10.Acknowledgements

The authors would like to thank Mike McBride, Kyle Rose, Roni Even for their detailed review, comments and suggestions which helped improve this document.

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>.

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

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

[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>.

11.2. Informative References

[RFC4105] Le Roux, J., Ed., Vasseur, J., Ed., and J. Boyle, Ed., "Requirements for Inter-Area MPLS Traffic Engineering", RFC 4105, DOI 10.17487/RFC4105, June 2005, <https://www.rfc-editor.org/info/rfc4105>.

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

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

[RFC4726] Farrel, A., Vasseur, J., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, DOI 10.17487/RFC4726, November 2006, <https://www.rfc-editor.org/info/rfc4726>.

[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>.

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

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, DOI 10.17487/RFC5394, December 2008, <https://www.rfc-editor.org/info/rfc5394>.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, DOI 10.17487/RFC5520, April 2009, <https://www.rfc-editor.org/info/rfc5520>.

[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, <https://www.rfc-editor.org/info/rfc5441>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <https://www.rfc-editor.org/info/rfc6805>.

[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>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <https://www.rfc-editor.org/info/rfc7420>.

[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>.

[RFC7897] Dhody, D., Palle, U., and R. Casellas, "Domain Subobjects for the Path Computation Element Communication Protocol (PCEP)", RFC 7897, DOI 10.17487/RFC7897, June 2016, <https://www.rfc-editor.org/info/rfc7897>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, DOI 10.17487/RFC8253, October 2017, <https://www.rfc-editor.org/info/rfc8253>.

[I-D.ietf-pce-pcep-yang]
Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-ietf-pce-pcep-yang-11 (work in progress), March 2019.

[I-D.ietf-pce-stateful-hpce]
Dhody, D., Lee, Y., Ceccarelli, D., Shin, J., King, D., and O. Dios, "Hierarchical Stateful Path Computation Element (PCE).", draft-ietf-pce-stateful-hpce-07 (work in progress), April 2019.

[I-D.dhodylee-pce-pcep-ls]
Dhody, D., Lee, Y., and D. Ceccarelli, "PCEP Extension for Distribution of Link-State and TE Information.", draft-dhodylee-pce-pcep-ls-13 (work in progress), February 2019.

Authors' Addresses

Fatai Zhang
Huawei
Huawei Base, Bantian, Longgang District
Shenzhen 518129
China

EMail: zhangfatai@huawei.com

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

EMail: quintin.zhao@huawei.com

Oscar Gonzalez de Dios
Telefonica
Don Ramon de la Cruz 82-84
Madrid 28045
Spain

EMail: oscar.gonzalezdedios@telefonica.com

Ramon Casellas
CTTC
Av. Carl Friedrich Gauss n.7
Barcelona, Castelldefels
Spain

EMail: ramon.casellas@cttc.es

Daniel King
Old Dog Consulting
UK

EMail: daniel@olddog.co.uk

Appendix

A1. Implementation Status

The H-PCE architecture and protocol procedures describe in this I-D were implemented and tested for a variety of optical research applications.

The Appendix should be removed before publication.

A1.1. Inter-layer traffic engineering with H-PCE

This work was led by:

o Ramon Casellas [ramon.casellas@cttc.es]

o Centre Tecnologic de Telecomunicacions de Catalunya (CTTC)

The H-PCE instances (parent and child) were multi-threaded asynchronous processes. Implemented in C++11, using C++ Boost Libraries. The targeted system used to deploy and run H-PCE applications was a POSIX system (Debian GNU/Linux operating system).

Some parts of the software may require a Linux Kernel, the availability of a Routing Controller running collocated in the same host and the usage of libnetfilter / libipq and GNU/Linux firewalling capabilities. Most of the functionality, including algorithms is done by means of plugins (e.g., as shared libraries or .so files in Unix systems).

The CTTC PCE supports the H-PCE architecture, but also supports stateful PCE with active capabilities, as an OpenFlow controller, and has dedicated plugins to support monitoring, BRPC, P2MP, path keys, back end PCEs. Management of the H-PCE entities was supported via HTTP and CLI via Telnet.

Further details of the H-PCE prototyping and experimentation can be found in the following scientific papers:

R. Casellas, R. Martinez, R. Munoz, L. Liu, T. Tsuritani, I. Morita, "Inter-layer traffic engineering with hierarchical-PCE in MPLS-TP over wavelength switched optical networks" , Optics Express, Vol. 20, No. 28, December 2012.

R. Casellas, R. Martinez, R. Munoz, L. Liu, T. Tsuritani, I. Morita, M. Msurusawa, "Dynamic virtual link mesh topology aggregation in multi-domain translucent WSON with hierarchical-PCE", Optics Express Journal, Vol. 19, No. 26, December 2011.

R. Casellas, R. Munoz, R. Martinez, R. Vilalta, L. Liu, T. Tsuritani, I. Morita, V. Lopez, O. Gonzalez de Dios, J. P. Fernandez-Palacios, "SDN based Provisioning Orchestration of

OpenFlow/GMPLS Flexi-grid Networks with a Stateful Hierarchical PCE", in Proceedings of Optical Fiber Communication Conference and Exposition (OFC), 9-13 March, 2014, San Francisco (EEUU). Extended Version to appear in Journal Of Optical Communications and Networking January 2015

F. Paolucci, O. Gonzalez de Dios, R. Casellas, S. Duhovnikov, P. Castoldi, R. Munoz, R. Martinez, "Experimenting Hierarchical PCE Architecture in a Distributed Multi-Platform Control Plane Testbed" , in Proceedings of Optical Fiber Communication Conference and Exposition (OFC) and The National Fiber Optic Engineers Conference (NFOEC), 4-8 March, 2012, Los Angeles, California (USA).

R. Casellas, R. Martinez, R. Munoz, L. Liu, T. Tsuritani, I. Morita, M. Tsurusawa, "Dynamic Virtual Link Mesh Topology Aggregation in Multi-Domain Translucent WSON with Hierarchical-PCE", in Proceedings of 37th European Conference and Exhibition on Optical Communication (ECOC 2011), 18-22 September 2011, Geneve (Switzerland).

R. Casellas, R. Munoz, R. Martinez, "Lab Trial of Multi-Domain Path Computation in GMPLS Controlled WSON Using a Hierarchical PCE", in Proceedings of OFC/NFOEC Conference (OFC2011), 10 March 2011, Los Angeles (USA).

A1.2. Telefonica Netphony (Open Source PCE)

The Telefonica Netphony PCE is an open source Java-based implementation of a Path Computation Element, with several flavours, and a Path Computation Client. The PCE follows a modular architecture and allows to add customized algorithms. The PCE has also stateful and remote initiation capabilities. In current version, three components can be built, a domain PCE (aka child PCE), a parent PCE (ready for the H-PCE architecture) and a PCC (path computation client).

This work was led by:

o Oscar Gonzalez de Dios [oscar.gonzalezdedios@telefonica.com]

o Victor Lopez Alvarez [victor.lopezalvarez@telefonica.com]

o Telefonica I+D, Madrid, Spain

The PCE code is publicly available in a GitHub repository:

o https://github.com/telefonicaid/netphony-pce

The PCEP protocol encodings are located in the following repository:

o https://github.com/telefonicaid/netphony-network protocols

The traffic engineering database and a BGP-LS speaker to fill the database is located in:

o https://github.com/telefonicaid/netphony-topology

The parent and child PCE are multi-threaded java applications. The path computation uses the jgrapht free Java class library (0.9.1) that provides mathematical graph-theory objects and algorithms. Current version of netphony PCE runs on java 1.7 and 1.8, and has been tested in GNU/Linux, Mac OS-X and Windows environments. The management of the parent and domain PCEs is supported though CLI via Telnet, and configured via XML files.

Further details of the netphony H-PCE prototyping and experimentation can be found in the following research papers:

o O. Gonzalez de Dios, R. Casellas, F. Paolucci, A. Napoli, L. Gifre, A. Dupas, E, Hugues-Salas, R. Morro, S. Belotti, G. Meloni, T. Rahman, V.P Lopez, R. Martinez, F. Fresi, M. Bohn, S. Yan, L. Velasco, . Layec and J. P. Fernandez-Palacios: Experimental Demonstration of Multivendor and Multidomain EON With Data and Control Interoperability Over a Pan-European Test Bed, in Journal of Lightwave Technology, Dec. 2016, Vol. 34, Issue 7, pp. 1610-1617.

o O. Gonzalez de Dios, R. Casellas, R. Morro, F. Paolucci, V. Lopez, R. Martinez, R. Munoz, R. Villalta, P. Castoldi: "Multi-partner Demonstration of BGP-LS enabled multi-domain EON, in Journal of Optical Communications and Networking, Dec. 2015, Vol. 7, Issue 12, pp. B153-B162.

o F. Paolucci, O. Gonzalez de Dios, R. Casellas, S. Duhovnikov, P. Castoldi, R. Munoz, R. Martinez, "Experimenting Hierarchical PCE Architecture in a Distributed Multi-Platform Control Plane Testbed" , in Proceedings of Optical Fiber Communication Conference and Exposition (OFC) and The National Fiber Optic Engineers Conference (NFOEC), 4-8 March, 2012, Los Angeles, California (USA).

A1.3. H-PCE Proof of Concept developed by Huawei

Huawei developed this H-PCE on the Huawei Versatile Routing Platform (VRP) to experiment with the hierarchy of PCE. Both end to end path computation as well as computation for domain-sequence are supported.

This work was led by:

o Udayasree Pallee [udayasreereddy@gmail.com]

o Dhruv Dhody [dhruv.ietf@gmail.com]

o Huawei Technologies, Bangalore, India

Further work on stateful H-PCE [I-D.ietf-pce-stateful-hpce] is being carried out on ONOS.

PCE Working Group D. King
Internet Draft Old Dog Consulting
Intended status: Informational H. Zheng
Expires: January 9, 2020 Huawei Technologies
July 8, 2019

Applicability of the Path Computation Element to Inter-Area and Inter-AS MPLS and GMPLS Traffic Engineering

draft-ietf-pce-inter-area-as-applicability-08

Abstract

The Path Computation Element (PCE) may be used for computing services that traverse multi-area and multi-AS Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineered (TE) networks.

This document examines the applicability of the PCE architecture, protocols, and protocol extensions for computing multi-area and multi-AS paths in MPLS and GMPLS networks.

Status of This Memo

Copyright Notice

1. Introduction

Computing paths across large multi-domain environments may require special computational components and cooperation between entities in different domains capable of complex path computation.

Issues that may exist when routing in multi-domain networks include:

- o Often there is a lack of full topology and TE information across domains;
- o No single node has the full visibility to determine an optimal or even feasible end-to-end path across domains;
- o How to evaluate and select the exit point and next domain boundary from a domain?
- o How might the ingress node determine which domains should be used for the end-to-end path?

Often information exchange across multiple domains is limited due to the lack of trust relationship, security issues, or scalability issues even if there is a trust relationship between domains.

The Path Computation Element (PCE) [RFC4655] provides an architecture and a set of functional components to address the problem space, and issues highlighted above.

A PCE may be used to compute end-to-end paths across multi-domain environments using a per-domain path computation technique [RFC5152]. The so called backward recursive path computation (BRPC) mechanism [RFC5441] defines a PCE-based path computation procedure to compute inter-domain constrained Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineered (TE) networks. However, both per-domain and BRPC techniques assume that the sequence of domains to be crossed from source to destination is known, either fixed by the network operator or obtained by other means.

In more advanced deployments (including multi-area and multi-Autonomous System (multi-AS) environments) the sequence of domains may not be known in advance and the choice of domains in the end-to-end domain sequence might be critical to the determination of an optimal end-to-end path. In this case the use of the Hierarchical PCE [RFC6805] architecture and mechanisms may be used to discover the intra-area path and select the optimal end-to-end domain sequence.

This document describes the processes and procedures available when using the PCE architecture and protocols, for computing inter-area and inter-AS MPLS and GMPLS Traffic Engineered paths.

This document scope does not include discussion on stateful PCE, active PCE, remotely initiated PCE, or PCE as a central controller (PCECC) deployment scenarios.

1.1 Domains

Generally, a domain can be defined as a separate administrative, geographic, or switching environment within the network. A domain may be further defined as a zone of routing or computational ability. Under these definitions a domain might be categorized as an Autonomous System (AS) or an Interior Gateway Protocol (IGP) area (as per [RFC4726] and [RFC4655]).

For the purposes of this document, a domain is considered to be a collection of network elements within an area or AS that has a common sphere of address management or path computational responsibility. Wholly or partially overlapping domains are not within the scope of this document.

In the context of GMPLS, a particularly important example of a domain is the Automatically Switched Optical Network (ASON) subnetwork [G-8080]. In this case, computation of an end-to-end path requires the selection of nodes and links within a parent domain where some nodes may, in fact, be subnetworks. Furthermore, a domain might be an ASON routing area [G-7715]. A PCE may perform the path computation function of an ASON routing controller as described in [G-7715-2].

It is assumed that the PCE architecture is not applied to a large group of domains, such as the Internet.

1.2 Path Computation

For the purpose of this document, it is assumed that the path computation is the sole responsibility of the PCE as per the architecture defined in [RFC4655]. When a path is required the Path Computation Client (PCC) will send a request to the PCE. The PCE will apply the required constraints and compute a path and return a response to the PCC. In the context of this document it may be necessary for the PCE to co-operate with other PCEs in adjacent domains (as per BRPC [RFC5441]) or cooperate with a Parent PCE (as per [RFC6805]).

It is entirely feasible that an operator could compute a path across multiple domains without the use of a PCE if the relevant domain information is available to the network planner or network management platform. The definition of what relevant information is required to perform this network planning operation and how that information is discovered and applied is outside the scope of this document.

1.2.1 PCE-based Path Computation Procedure

As highlighted, the PCE is an entity capable of computing an inter-domain TE path upon receiving a request from a PCC. There could be a single PCE per domain, or single PCE responsible for all domains. A PCE may or may not reside on the same node as the requesting PCC. A path may be computed by either a single PCE node or a set of distributed PCE nodes that collaborate during path computation.

[RFC4655] defines that a PCC should send a path computation request to a particular PCE, using [RFC5440] (PCC-to-PCE communication). This negates the need to broadcast a request to all the PCEs. Each PCC can maintain information about the computation capabilities of the PCEs, it is aware of. The PCC-PCE capability awareness can be configured using static configurations or by automatic and dynamic PCE discovery procedures.

If a network path is required, the PCC will send a path computation request to the PCE. A PCE may then compute the end-to-end path if it is aware of the topology and TE information required to compute the entire path. If the PCE is unable to compute the entire path, the PCE architecture provides co-operative PCE mechanisms for the resolution of path computation requests when an individual PCE does not have sufficient TE visibility.

End-to-end path segments may be kept confidential through the application of path keys, to protect partial or full path information. A path key that is a token that replaces a path segment in an explicit route. The path key mechanism is described in [RFC5520]

1.3 Traffic Engineering Aggregation and Abstraction

Networks are often constructed from multiple areas or ASes that are interconnected via multiple interconnect points. To maintain network confidentiality and scalability TE properties of each area and AS are not generally advertized outside each specific area or AS.

TE aggregation or abstraction provide mechanism to hide information but may cause failed path setups or the selection of suboptimal end-to-end paths [RFC4726]. The aggregation process may also have significant scaling issues for networks with many possible routes and multiple TE metrics. Flooding TE information breaks confidentiality and does not scale in the routing protocol.

The PCE architecture and associated mechanisms provide a solution to avoid the use of TE aggregation and abstraction.

1.4 Traffic Engineered Label Switched Paths

This document highlights the PCE techniques and mechanisms that exist for establishing TE packet and optical LSPs across multiple areas (inter-area TE LSP) and ASes (inter-AS TE LSP). In this context and within the remainder of this document, we consider all LSPs to be constraint-based and traffic engineered.

Three signaling options are defined for setting up an inter-area or inter-AS LSP [RFC4726]:

- Contiguous LSP
- Stitched LSP
- Nested LSP

All three signaling methods are applicable to the architectures and procedures discussed in this document.

1.5 Inter-area and Inter-AS Capable PCE Discovery

When using a PCE-based approach for inter-area and inter-AS path computation, a PCE in one area or AS may need to learn information related to inter-AS capable PCEs located in other ASes. The PCE discovery mechanism defined in [RFC5088] and [RFC5089] facilitates the discovery of PCEs, and disclosure of information related to inter-area and inter-AS capable PCEs.

1.6 Objective Functions

An Objective Function (OF) [RFC5541], or set of OFs, specifies the intentions of the path computation and so defines the "optimality", in the context of the computation request.

An OF specifies the desired outcome of a computation. An OF does not describe or specify the algorithm to use. Also, an implementation may apply any algorithm, or set of algorithms, to achieve the result indicated by the OF. A number of general OFs are specified in [RFC5541].

Various OFs may be included in the PCE computation request to satisfy the policies encoded or configured at the PCC, and a PCE may be subject to policy in determining whether it meets the OFs included in the computation request or applies its own OFs.

During inter-domain path computation, the selection of a domain sequence, the computation of each (per-domain) path fragment, and the determination of the end-to-end path may each be subject to different OFs and policy.

2. Terminology

This document also uses the terminology defined in [RFC4655] and [RFC5440]. Additional terminology is defined below:

ABR: IGP Area Border Router, a router that is attached to more than one IGP area.

ASBR: Autonomous System Border Router, a router used to connect together ASes of a different or the same Service Provider via one or more inter-AS links.

Inter-area TE LSP: A TE LSP whose path transits through two or more IGP areas.

Inter-AS MPLS TE LSP: A TE LSP whose path transits through two or more ASes or sub-ASes (BGP confederations

SRLG: Shared Risk Link Group.

TED: Traffic Engineering Database, which contains the topology and resource information of the domain. The TED may be fed by Interior Gateway Protocol (IGP) extensions or potentially by other means.

3. Issues and Considerations

3.1 Multi-homing

Networks constructed from multi-areas or multi-AS environments may have multiple interconnect points (multi-homing). End-to-end path computations may need to use different interconnect points to avoid a single point failure disrupting both primary and backup services.

3.2 Destination Location

The PCC asking for an inter-domain path computation is typically aware of the identity of the destination node. If the PCC is aware of the destination domain, it may supply the destination domain information as part of the path computation request. However, if the PCC does not know the destination domain this information must be determined by another method.

3.3 Domain Confidentiality

Where the end-to-end path crosses multiple domains, it may be possible that each domain (AS or area) are administered by separate Service Providers, it would break confidentiality rules for a PCE to supply a path segment to a PCE in another domain, thus disclosing AS-internal topology information.

If confidentiality is required between domains (ASes and areas) belonging to different Service Providers, then cooperating PCEs cannot exchange path segments or else the receiving PCE or PCC will be able to see the individual hops through another domain.

This topic is discussed further in Section 8 of this document.

4. Domain Topologies

Constraint-based inter-domain path computation is a fundamental requirement for operating traffic engineered MPLS [RFC3209] and GMPLS [RFC3473] networks, in inter-area and inter-AS (multi-domain) environments. Path computation across multi-domain networks is complex and requires computational co-operational entities like the PCE.

4.1 Selecting Domain Paths

Where the sequence of domains is known a priori, various techniques can be employed to derive an optimal multi-domain path. If the domains are connected to a simple path with no branches and single links between all domains, or if the preferred points of interconnection is also known, the Per-Domain Path Computation [RFC5152] technique may be used. Where there are multiple connections between domains and there is no preference for the choice of points of interconnection, BRPC [RFC5441] can be used to derive an optimal path.

When the sequence of domains is not known in advance, or the end-to-end path will have to navigate a mesh of small domains (especially typical in optical networks), the optimum path may be derived through the application of a Hierarchical PCE [RFC6805].

4.2 Domain Sizes

Very frequently network domains are composed of dozens or hundreds of network elements. These network elements are usually interconnected in a partial-mesh fashion, to provide survivability against dual failures, and to benefit from the traffic engineering capabilities from MPLS and GMPLS protocols. Network operator feedback in the development of the document highlighted that node degree (the number of neighbors per node) typically ranges from 3 to 10 (4-5 is quite common).

4.3 Domain Diversity

Domain and path diversity may also be required when computing end-to-end paths. Domain diversity should facilitate the selection of paths that share ingress and egress domains, but do not share transit domains. Therefore, there must be a method allowing the inclusion or exclusion of specific domains when computing end-to-end paths.

4.4 Synchronized Path Computations

In some scenarios, it would be beneficial for the operator to rely on the capability of the PCE to perform synchronized path computation.

Synchronized path computations, known as Synchronization VECtors (SVECs) are used for dependent path computations. SVECs are defined in [RFC5440] and [RFC6007] provides an overview for the use of the PCE SVEC list for synchronized path computations when computing dependent requests.

In H-PCE deployments, a child PCE will be able to request both dependent and synchronized domain diverse end to end paths from its parent PCE.

4.5 Domain Inclusion or Exclusion

A domain sequence is an ordered sequence of domains traversed to reach the destination domain. A domain sequence may be supplied during path computation to guide the PCEs or derived via the use of

Hierarchical PCE (H-PCE).

During multi-domain path computation, a PCC may request specific domains to be included or excluded in the domain sequence using the Include Route Object (IRO) [RFC5440] and Exclude Route Object (XRO) [RFC5521]. The use of Autonomous Number (AS) as an abstract node representing a domain is defined in [RFC3209]. [RFC7897] specifies new sub-objects to include or exclude domains such as an IGP area or a 4-Byte AS number.

An operator may also need to avoid a path that uses specified nodes for administrative reasons, or if a specific connectivity service required to have a 1+1 protection capability, two completely disjoint paths must be established. A mechanism known as Shared Risk Link Group (SRLG) information may be used to ensure path diversity.

5. Applicability of the PCE to Inter-area Traffic Engineering

As networks increase in size and complexity, it may be required to introduce scaling methods to reduce the amount of information flooded within the network and make the network more manageable. An IGP hierarchy is designed to improve IGP scalability by dividing the IGP domain into areas and limiting the flooding scope of topology information to within area boundaries. This restricts visibility of the area to routers in a single area. If a router needs to compute the route to a destination located in another area, a method would be required to compute a path across area boundaries.

In order to support multiple vendors in a network, in cases where data or control plane technologies cannot interoperate, it is useful to divide the network into vendor domains. Each vendor domain is an IGP area, and the flooding scope of the topology (as well as any other relevant information) is limited to the area boundaries.

Per-domain path computation [RFC5152] exists to provide a method of inter-area path computation. The per-domain solution is based on loose hop routing with an Explicit Route Object (ERO) expansion on each Area Border Router (ABR). This allows an LSP to be established using a constrained path, however at least two issues exist:

- o This method does not guarantee an optimal constrained path.
- o The method may require several crankback signaling messages, as per [RFC4920], increasing signaling traffic and delaying the LSP setup.

The PCE-based architecture [RFC4655] is designed to solve inter-area path computation problems. The issue of limited topology visibility is resolved by introducing path computation entities that are able to cooperate in order to establish LSPs with source and destinations located in different areas.

5.1. Inter-area Routing

An inter-area TE-LSP is an LSP that transits through at least two IGP areas. In a multi-area network, topology visibility remains local to a given area for scaling and privacy purposes, a node in one area will not be able to compute an end-to-end path across multiple areas without the use of a PCE.

5.1.1. Area Inclusion and Exclusion

The BRPC method [RFC5441] of path computation provides a more optimal method to specify inclusion or exclusion of an ABR. Using the BRPC procedure an end-to-end path is recursively computed in reverse from the destination domain, towards the source domain. Using this method, an operator might decide if an area must be included or excluded from the inter-area path computation.

5.1.2. Strict Explicit Path and Loose Path

A strict explicit Path is defined as a set of strict hops, while a loose path is defined as a set of at least one loose hop and zero or more strict hops. It may be useful to indicate, during the path computation request, if a strict explicit path is required or not. An inter-area path may be strictly explicit or loose (e.g., a list of ABRs as loose hops).

A PCC request to a PCE does allow the indication of whether a strict explicit path across specific areas ([RFC7897]) is required or desired, or if the path request is loose.

5.1.3. Inter-Area Diverse Path Computation

It may be necessary to compute a path that is partially or entirely diverse, from a previously computed path, to avoid fate sharing of a primary service with a corresponding backup service. There are various levels of diversity in the context of an inter-area network:

o Per-area diversity (intra-area path segments are link, node or SRLG disjoint.

o Inter-area diversity (end-to-end inter-area paths are link, node or SRLG disjoint).

Note that two paths may be disjoint in the backbone area but non-disjoint in peripheral areas. Also, two paths may be node disjoint within areas but may share ABRs, in which case path segments within an area is node disjoint, but end-to-end paths are not node-disjoint. Per-Domain [RFC5152], BRPC [RFC5441] and H-PCE [RFC6805] mechanisms all support the capability to compute diverse paths across multi-area topologies.

6. Applicability of the PCE to Inter-AS Traffic Engineering

As discussed in section 4 (Applicability of the PCE to Inter-area Traffic Engineering) it is necessary to divide the network into smaller administrative domains, or ASes. If an LSR within an AS needs to compute a path across an AS boundary, it must also use an inter-AS computation technique. [RFC5152] defines mechanisms for the computation of inter-domain TE LSPs using network elements along the signaling paths to compute per-domain constrained path segments.

The PCE was designed to be capable of computing MPLS and GMPLS paths across AS boundaries. This section outlines the features of a PCE-enabled solution for computing inter-AS paths.

6.1 Inter-AS Routing

6.1.1. AS Inclusion and Exclusion

[RFC5441] allows the specifying of inclusion or exclusion of an AS or an ASBR. Using this method, an operator might decide if an AS must be include or exclude from the inter-AS path computation. Exclusion and/or inclusion could also be specified at any step in the LSP path computation process by a PCE (within the BRPC algorithm) but the best practice would be to specify them at the edge. In opposition to the strict and loose path, AS inclusion or exclusion doesn’t impose topology disclosure as ASes are public entity as well as their interconnection.

6.2 Inter-AS Bandwidth Guarantees

Many operators with multi-AS domains will have deployed MPLS-TE DiffServ either across their entire network or at the domain edges on CE-PE links. In situations where strict QOS bounds are required, admission control inside the network may also be required.

When the propagation delay can be bounded, the performance targets, such as maximum one-way transit delay may be guaranteed by providing bandwidth guarantees along the DiffServ-enabled path, these requirements are described in [RFC4216].

One typical example of the requirements in [RFC4216] is to provide bandwidth guarantees over an end-to-end path for VoIP traffic classified as EF (Expedited Forwarding) class in a DiffServ-enabled network. In the case where the EF path is extended across multiple ASes, inter-AS bandwidth guarantee would be required.

Another case for inter-AS bandwidth guarantee is the requirement for guaranteeing a certain amount of transit bandwidth across one or multiple ASes.

6.3 Inter-AS Recovery

During a path computation process, a PCC request may contain the requirement to compute a backup LSP for protecting the primary LSP, 1+1 protection. A single LSP or multiple backup LSPs may also be used for a group of primary LSPs, this is typically known as m:n protection.

Other inter-AS recovery mechanisms include [RFC4090] which adds fast re-route (FRR) protection to an LSP. So, the PCE could be used to trigger computation of backup tunnels in order to protect Inter-AS connectivity.

Inter-AS recovery clearly requires backup LSPs for service protection but it would also be advisable to have multiple PCEs deployed for path computation redundancy, especially for service restoration in the event of catastrophic network failure.

6.4 Inter-AS PCE Peering Policies

Like BGP peering policies, inter-AS PCE peering policies is a requirement for operator. In inter-AS BRPC process, PCE must cooperate in order to compute the end-to-end LSP. So, the AS path must not only follow technical constraints, e.g. bandwidth availability, but also policies defined by the operator.

Typically PCE interconnections at an AS level must follow agreed contract obligations, also known as peering agreements. The PCE peering policies are the result of the contract negotiation and govern the relation between the different PCE.

7. Multi-domain PCE Deployment Options

7.1 Traffic Engineering Database and Synchronization

An optimal path computation requires knowledge of the available network resources, including nodes and links, constraints, link connectivity, available bandwidth, and link costs. The PCE operates on a view of the network topology as presented by a TED. As discussed in [RFC4655] the TED used by a PCE may be learnt by the relevant IGP extensions.

Thus, the PCE may operate its TED is by participating in the IGP running in the network. In an MPLS-TE network, this would require OSPF-TE [RFC3630] or ISIS-TE [RFC5305]. In a GMPLS network it would utilize the GMPLS extensions to OSPF and IS-IS defined in [RFC4203] and [RFC5307]. Inter-as connectivity information may be populated via [RFC5316] and [RFC5392].

An alternative method to provide network topology and resource information is offered by [RFC7752], which is described in the following section.

7.1.1 Applicability of BGP-LS to PCE

The concept of exchange of TE information between Autonomous Systems (ASes) is discussed in [RFC7752]. The information exchanged in this way could be the full TE information from the AS, an aggregation of that information, or a representation of the potential connectivity across the AS. Furthermore, that information could be updated frequently (for example, for every new LSP that is set up across the AS) or only at threshold-crossing events.

In an H-PCE deployment, the parent PCE will require the inter-domain topology and link status between child domains. This information may be learnt by a BGP-LS speaker and provided to the parent PCE, furthermore link-state performance including delay, available bandwidth and utilized bandwidth may also be provided to the parent PCE for optimal path link selection.

7.2 Pre-Planning and Management-Based Solutions

Offline path computation is performed ahead of time, before the LSP setup is requested. That means that it is requested by, or performed as part of, an Operation Support System (OSS) management application. This model can be seen in Section 5.5 of [RFC4655].

The offline model is particularly appropriate to long-lived LSPs (such as those present in a transport network) or for planned responses to network failures. In these scenarios, more planning is normally a feature of LSP provisioning.

The management system may also use a PCE and BRPC to pre-plan an AS sequence, and the source domain PCE and per-domain path computation to be used when the actual end-to-end path is required. This model may also be used where the operator wishes to retain full manual control of the placement of LSPs, using the PCE only as a computation tool to assist the operator, not as part of an automated network.

In environments where operators peer with each other to provide end-to-end paths, the operator responsible for each domain must agree to what extent paths must be pre-planned or manually controlled.

8. Domain Confidentiality

This section discusses the techniques that co-operating PCEs can use to compute inter-domain paths without each domain disclosing sensitive internal topology information (such as explicit nodes or links within the domain) to the other domains.

Confidentiality typically applies to inter-provider (inter-AS) PCE communication. Where the TE LSP crosses multiple domains (ASes or areas), the path may be computed by multiple PCEs that cooperate together. With each local PCE responsible for computing a segment of the path.

In situations where ASes are administered by separate Service Providers, it would break confidentiality rules for a PCE to supply a path segment details to a PCE responsible another domain, thus disclosing AS-internal or area topology information.

8.1 Loose Hops

A method for preserving the confidentiality of the path segment is for the PCE to return a path containing a loose hop in place of the segment that must be kept confidential. The concept of loose and strict hops for the route of a TE LSP is described in [RFC3209].

[RFC5440] supports the use of paths with loose hops, and it is a local policy decision at a PCE whether it returns a full explicit path with strict hops or uses loose hops. A path computation request may require an explicit path with strict hops, or may allow loose hops as detailed in [RFC5440].

8.2 Confidential Path Segments and Path Keys

[RFC5520] defines the concept and mechanism of Path-Key. A Path-Key is a token that replaces the path segment information in an explicit route. The Path-Key allows the explicit route information to be encoded and in the PCEP ([RFC5440]) messages exchanged between the

PCE and PCC.

This Path-Key technique allows explicit route information to be used for end-to-end path computation, without disclosing internal topology information between domains.

9. Point-to-Multipoint

For inter-domain point-to-multipoint application scenarios using MPLS-TE LSPs, the complexity of domain sequences, domain policies, choice and number of domain interconnects is magnified compared to point-to-point path computations. As the size of the network grows, the number of leaves and branches increase, further increasing the complexity of the overall path computation problem. A solution for managing point-to-multipoint path computations may be achieved using the PCE inter-domain point-to-multipoint path computation [RFC7334] procedure.

10. Optical Domains

The International Telecommunications Union (ITU) defines the ASON architecture in [G-8080]. [G-7715] defines the routing architecture for ASON and introduces a hierarchical architecture. In this architecture, the Routing Areas (RAs) have a hierarchical relationship between different routing levels, which means a parent (or higher level) RA can contain multiple child RAs. The interconnectivity of the lower RAs is visible to the higher-level RA.

In the ASON framework, a path computation request is termed a Route Query. This query is executed before signaling is used to establish an LSP termed a Switched Connection (SC) or a Soft Permanent Connection (SPC). [G-7715-2] defines the requirements and architecture for the functions performed by Routing Controllers (RC) during the operation of remote route queries - an RC is synonymous with a PCE.

In the ASON routing environment, an RC responsible for an RA may communicate with its neighbor RC to request the computation of an end-to-end path across several RAs. The path computation components and sequences are defined as follows:

- o Remote route query. An operation where a routing controller communicates with another routing controller, which does not have the same set of layer resources, in order to compute a routing path in a collaborative manner.

- o Route query requester. The connection controller or RC that sends a route query message to a routing controller requesting for one or more routing paths that satisfy a set of routing constraints.

- o Route query responder. An RC that performs path computation upon reception of a route query message from a routing controller or connection controller, sending a response back at the end of computation.

When computing an end-to-end connection, the route may be computed by a single RC or multiple RCs in a collaborative manner and the two scenarios can be considered a centralized remote route query model and distributed remote route query model. RCs in an ASON environment can also use the hierarchical PCE [RFC6805] model to match fully the ASON hierarchical routing model.

10.1 Abstraction and Control of TE Networks (ACTN)

Where a single operator operates multiple TE domains (including optical environments) then Abstraction and Control of TE Networks (ACTN) framework [RFC8453] may be used to create an abstracted (virtualized network) view of underlay interconnected domains. This underlay connectivity then be exposed to higher-layer control entities and applications.

ACTN describes the method and procedure for coordinating the underlay per-domain Physical Network Controllers (PNCs), which may be PCEs, via a hierarchical model to facilitate setup of end-to-end connections across inter-connected TE domains.

11. Policy

Policy is important in the deployment of new services and the operation of the network. [RFC5394] provides a framework for PCE-based policy-enabled path computation. This framework is based on the Policy Core Information Model (PCIM) as defined in [RFC3060] and further extended by [RFC3460].

When using a PCE to compute inter-domain paths, policy may be invoked by specifying:

- o Each PCC must select which computations will be requested to a PCE;

- o Each PCC must select which PCEs it will use;

- o Each PCE must determine which PCCs are allowed to use its services and for what computations;

- o The PCE must determine how to collect the information in its TED, whom to trust for that information, and how to refresh/update the information;

- o Each PCE must determine which objective functions and which algorithms to apply.

12. Manageability Considerations

General PCE management considerations are discussed in [RFC4655]. In the case of multi-domains within a single service provider network, the management responsibility for each PCE would most likely be handled by the same service provider. In the case of multiple ASes within different service provider networks, it will likely be necessary for each PCE to be configured and managed separately by each participating service provider, with policy being implemented based on a previously agreed set of principles.

12.1 Control of Function and Policy

As per PCEP [RFC5440] implementation allow the user to configure a number of PCEP session parameters. These are detailed in section 8.1 of [RFC5440].

In H-PCE deployments the administrative entity responsible for the management of the parent PCEs for multi-areas would typically be a single service provider. In the multiple ASes (managed by different service providers), it may be necessary for a third party to manage the parent PCE.

12.2 Information and Data Models

A PCEP MIB module is defined in [RFC7420] that describes managed objects for modeling of PCEP communication including:

- o PCEP client configuration and status,

- o PCEP peer configuration and information,

- o PCEP session configuration and information,

- o Notifications to indicate PCEP session changes.

A YANG module for PCEP has also been proposed [PCEP-YANG].

An H-PCE MIB module, or YANG data model, will be required to report parent PCE and child PCE information, including:

- parent PCE configuration and status,
- child PCE configuration and information,
- notifications to indicate session changes between parent PCEs and child PCEs, and
- notification of parent PCE TED updates and changes.

12.3 Liveness Detection and Monitoring

PCEP includes a keepalive mechanism to check the liveliness of a PCEP peer and a notification procedure allowing a PCE to advertise its overloaded state to a PCC. In a multi-domain environment [RFC5886] provides the procedures necessary to monitor the liveliness and performances of a given PCE chain.

12.4 Verifying Correct Operation

It is important to verify the correct operation of PCEP, [RFC5440] specifies the monitoring of key parameters. These parameters are detailed in [RFC5520].

12.5 Impact on Network Operation

[RFC5440] states that in order to avoid any unacceptable impact on network operations, a PCEP implementation should allow a limit to be placed on the number of sessions that can be set up on a PCEP speaker, it may also be practical to place a limit on the rate of messages sent by a PCC and received my the PCE.

13. Security Considerations

PCEP Security considerations are discussed in [RFC5440] and [RFC6952] Potential vulnerabilities include spoofing, snooping, falsification and using PCEP as a mechanism for denial of service attacks.

As PCEP operates over TCP, it may make use of TCP security encryption mechanisms, such as Transport Layer Security (TLS) and TCP Authentication Option (TCP-AO). Usage of these security mechanisms for PCEP is described in [RFC8253], and recommendations and best current practices in [RFC7525].

13.1 Multi-domain Security

Any multi-domain operation necessarily involves the exchange of information across domain boundaries. This does represent significant security and confidentiality risk.

It is expected that PCEP is used between PCCs and PCEs belonging to the same administrative authority, and using one of the aforementioned encryption mechanisms. Furthermore, PCEP allows individual PCEs to maintain confidentiality of their domain path information using path-keys.

14. IANA Considerations

This document makes no requests for IANA action.

15. Acknowledgements

The author would like to thank Adrian Farrel for his review, and Meral Shirazipour and Francisco Javier Jimenex Chico for their comments.

16. References

16.1. Normative References

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC4216] Zhang, R., Ed., and J.-P. Vasseur, Ed., "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, November 2005.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC4726] Farrel, A., Vasseur, J., and A. Ayyangar, "A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering", RFC 4726, November 2006.

[RFC5152] Vasseur, JP., Ayyangar, A., and R. Zhang, "A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)", RFC 5152, February 2008.

[RFC5440] Ayyangar, A., Farrel, A., Oki, E., Atlas, A., Dolganow, A., Ikejiri, Y., Kumaki, K., Vasseur, J., and J. Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5441] Vasseur, J.P., Ed., "A Backward Recursive PCE-based Computation (BRPC) procedure to compute shortest inter-domain Traffic Engineering Label Switched Paths", RFC5441, April 2009.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009.

[RFC5541] Le Roux, J., Vasseur, J., Lee, Y., "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC5541, December 2008.

[RFC6805] King, D. and A. Farrel, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS & GMPLS", RFC6805, July 2010.

16.2. Informative References

[RFC3060] Moore, B., Ellesson, E., Strassner, J., and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.

[RFC3460] Moore, B., Ed., "Policy Core Information Model (PCIM) Extensions", RFC 3460, January 2003.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC4203] Kompella, K., Ed., and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.

[RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, July 2007.

[RFC5088] Le Roux, JL., Vasseur, JP., Ikejiri, Y., and R. Zhang, "OSPF Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5088, January 2008.

[RFC5089] Le Roux, JL., Ed., Vasseur, JP., Ed., Ikejiri, Y., and R. Zhang, "IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery", RFC 5089, January 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.

[RFC5307] Kompella, K., Ed., and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008.

[RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", December 2008.

[RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", RFC 5392, January 2009.

[RFC5394] Bryskin, I., Papadimitriou, D., Berger, L., and J. Ash, "Policy-Enabled Path Computation Framework", RFC 5394, December 2008.

[RFC5521] Oki, E., Takeda, T., and A. Farrel, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Route Exclusions", RFC 5521, April 2009.

[RFC5886] Vasseur, JP., Le Roux, JL., and Y. Ikejiri, "A Set of Monitoring Tools for Path ComputationElement (PCE)-Based Architecture", RFC 5886, June 2010.

[RFC6007] Nishioka, I., King, D., "Use of the Synchronization VECtor (SVEC) List for Synchronized Dependent Path Computations", RFC6007, September 2010.

[G-8080] ITU-T Recommendation G.8080/Y.1304, Architecture for the automatically switched optical network (ASON).

[G-7715] ITU-T Recommendation G.7715 (2002), Architecture and Requirements for the Automatically Switched Optical Network (ASON).

[G-7715-2] ITU-T Recommendation G.7715.2 (2007), ASON routing architecture and requirements for remote route query.

[RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, May 2013.

[RFC7334] Zhao, Q., Dhody, D., Ali Z., King, D., Casellas, R., "PCE-based Computation Procedure To Compute Shortest Constrained P2MP Inter-domain Traffic Engineering Label Switched Paths", August 2014.

[RFC7420] Stephan, E., Koushik, K., Zhao, Q., King, D., "PCE Communication Protocol (PCEP) Management Information Base", December 2014.

[RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, May 2015.

[RFC7752] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", March 2016.

[RFC7897] Dhody, D., Palle, U., and R. Casellas, "Domain Subobjects for the Path Computation Element Communication Protocol (PCEP)", June 2016.

[RFC8253] Lopez, D., Gonzalez de Dios, O., Wu, Q., and D. Dhody, "PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)", RFC 8253, October 2017.

[RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.

[PCEP-YANG] Dhody, D., Hardwick, J., Beeram, V., and J. Tantsura, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", work in progress, October 2018.

17. Contributors

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Quintin Zhao
Huawei Technology
125 Nagog Technology Park
Acton, MA 01719
US

Email: qzhao@huawei.com

Julien Meuric
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex

Email: julien.meuric@orange-ftgroup.com

Olivier Dugeon
France Telecom
2, avenue Pierre-Marzin
22307 Lannion Cedex

Email: olivier.dugeon@orange-ftgroup.com

Jon Hardwick
Metaswitch Networks
100 Church Street
Enfield, Middlesex
United Kingdom

Email: jonathan.hardwick@metaswitch.com

Oscar Gonzalez de Dios
Telefonica I+D
Emilio Vargas 6, Madrid
Spain

Email: ogondio@tid.es

18. Author's Addresses

Daniel King
Old Dog Consulting
UK

Email: daniel@olddog.co.uk

Haomian Zheng
Huawei Technologies
F3 R&D Center, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: zhenghaomian@huawei.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

T. Saad
Juniper Networks
R. Gandhi
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

Copyright Notice

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:

- o Maximum reservable bandwidth, bandwidth constraints (BC)
- o Flooding parameters
  - * Flooding intervals and threshold values
- o interface attributes
  - * (Extended) administrative groups
  - * SRLG values
  - * TE metric value
- o 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:

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

TEAS Working Group   Xufeng Liu
Internet Draft   Volta Networks
Intended status: Standards Track   Igor Bryskin
                                                   Huawei Technologies
                                                   Vishnu Pavan Beeram
                                                            Tarek Saad
                                                      Juniper Networks
                                                         Himanshu Shah
                                                                 Ciena
                                               Oscar Gonzalez De Dios
                                                            Telefonica

Expires: December 19, 2019   June 19, 2019

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

Status of this Memo

Copyright Notice

Abstract

This document defines a YANG data model for representing, retrieving and manipulating Traffic Engineering (TE) Topologies. The model serves as a base model that other technology specific TE Topology models can augment.

Conventions used in this document

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

Table of Contents

1. Introduction

The Traffic Engineering Database (TED) is an essential component of Traffic Engineered (TE) systems that are based on MPLS-TE [RFC2702] and GMPLS [RFC3945]. The TED is a collection of all TE information about all TE nodes and TE links in the network. The TE Topology is a schematic arrangement of TE nodes and TE links present in a given TED. There could be one or more TE Topologies present in a given Traffic Engineered system. A TE Topology is the topology on which path computational algorithms are run to compute Traffic Engineered Paths (TE Paths).

This document defines a YANG [RFC7950] data model for representing and manipulating TE Topologies. This model contains technology agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

1.1. Terminology

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

The reader is assumed to be familiar with general body of work captured in currently available TE related RFCs. [RFC7926] serves as a good starting point for those who may be less familiar with Traffic Engineering related RFCs.

Some of the key terms used in this document are:

TED: The Traffic Engineering Database is a collection of all TE information about all TE nodes and TE links in a given network.

TE-Topology: The TE Topology is a schematic arrangement of TE nodes and TE links in a given TED. It forms the basis for a graph suitable for TE path computations.

Native TE Topology: Native TE Topology is a topology that is native to a given provider network. Native TE topology could be discovered via various routing protocols and/or subscribe/publish techniques. This is the topology on which path computational algorithms are run to compute TE Paths.

Customized TE Topology: Customized TE Topology is a custom topology that is produced by a provider for a given client. This topology typically makes abstractions on the provider’s Native TE Topology, and is provided to the client. The client receives the Customized TE Topology, and merges it into the client’s Native TE Topology. The client’s path computational algorithms aren’t typically run on the Customized TE Topology; they are run on the client’s Native TE Topology after the merge.

1.2. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A. of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[RFC6991]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te -types]

Table 1: Prefixes and corresponding YANG modules

2. Characterizing TE Topologies

The data model proposed by this document takes the following characteristics of TE Topologies into account:

- TE Topology is an abstract control-plane representation of the data-plane topology. Hence attributes specific to the data-plane must make their way into the corresponding TE Topology modeling. The TE Topology comprises of dynamic auto-discovered data as well as fairly static data associated with data-plane nodes and links. The dynamic data may change frequently, such as unreserved bandwidth available on data-plane links. The static data rarely changes, such as layer network identification, switching and adaptation capabilities and limitations, fate sharing, and administrative colors. It is possible for a single TE Topology to encompass TE information at multiple switching layers.

- TE Topologies are protocol independent. Information about topological elements may be learnt via link-state protocols, but the topology can exist without being dependent on any particular protocol.

- TE Topology may not be congruent to the routing topology in a given TE System. The routing topology is constructed based on routing adjacencies. There isn't always a one-to-one association between a TE-link and a routing adjacency. For example, the presence of a TE link between a pair of nodes doesn't necessarily imply the existence of a routing-adjacency between these nodes. To learn more, see [I-D.ietf-teas-te-topo-and-tunnel-modeling] and [I-D.ietf-teas-yang-l3-te-topo].

- Each TE Topological element has at least one information source associated with it. In some scenarios, there could be more than one information source associated with any given topological element.

- TE Topologies can be hierarchical. Each node and link of a given TE Topology can be associated with respective underlay topology. This means that each node and link of a given TE Topology can be associated with an independent stack of supporting TE Topologies.

- TE Topologies can be customized. TE topologies of a given network presented by the network provider to its client could be customized on per-client request basis. This customization could be performed by provider, by client or by provider/client negotiation. The relationship between a customized topology and provider's native topology could be captured as hierarchical (overlay-underlay), but otherwise the two topologies are decoupled from each other. A customized topology is presented to the client, while provider's native topology is known in its entirety to the provider itself.

3. Modeling Abstractions and Transformations

```
                                |  +---+              __
                                |  |   | TE Node     \/ TTP  o LTP
                                |  +---+
                                |
                                |  ----- TE Link
                                |  ***** Node Connectivity Matrix,
                                |        TTP Local Link Connectivity
                                |  @@@@@ TE Tunnel
                                o----------------------------------

      Node-1                                            Node-3
   +------------+                                    +------------+
   |    TTP-1   |                                    |    TTP-1   |
   |LTP  __     |           TE-Tunel-1               |     __     |
   |-6   \/@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\/     |
   o    *  *    oLTP-1           Node-2         LTP-6o    *  *    o
   |   *    *   |           +------------+           |   *    *   |
   |  * TTP-2*  |           |            |           |  * TTP-2*  |
   | *   __   * |LTP-2 LTP-6|            |LTP-1 LTP-5| *   __   * |
   o*    \/    *o-----------o************o-----------o*    \/    *o
   |LTP *  *    | Link-12   |          * | Link-23   |    *  *    |
   |-5 *    *   |      LTP-5|        *   |LTP-2      |   *    *   |
   +--o------o--+           o************o           +--o------o--+
    LTP-4  LTP-3            | *   *    * |            LTP-4  LTP-3
                            |  **     *  |
                            +--o------o--+
                             LTP-4  LTP-3
```

Figure 1: TE Topology Modeling Abstractions

3.1. TE Topology

TE topology is a traffic engineering representation of one or more layers of network topologies. TE topology is comprised of TE nodes (TE graph vertices) interconnected via TE links (TE graph edges). A TE topology is mapped to a TE graph.

3.2. TE Node

TE node is an element of a TE topology, presented as a vertex on TE graph. TE node represents one or several nodes, or a fraction of a node, which can be a switch or router that is physical or virtual. TE node belongs to and is fully defined in exactly one TE topology. TE node is assigned a unique ID within the TE topology scope. TE node attributes include information related to the data plane aspects of the associated node(s) (e.g. connectivity matrix), as well as configuration data (such as TE node name). A given TE node can be reached on the TE graph over one of TE links terminated by the TE node.

Multi-layer TE nodes providing switching functions at multiple network layers are an example where a physical node can be decomposed into multiple logical TE nodes, which are fractions of the physical node. Some of these (logical) TE nodes may reside in the client layer TE topology while the remaining TE nodes belong to the server layer TE topology.

In Figure 1, Node-1, Node-2, and Node-3 are TE nodes.

3.3. TE Link

TE link is an element of a TE topology, presented as an edge on TE graph. The arrows on an edge indicate one or both directions of the TE link. When there are a pair of parallel links of opposite directions, an edge without arrows is also used. TE link represents one or several (physical) links or a fraction of a link. TE link belongs to and is fully defined in exactly one TE topology. TE link is assigned a unique ID within the TE topology scope. TE link attributes include parameters related to the data plane aspects of the associated link(s) (e.g. unreserved bandwidth, resource maps/pools, etc.), as well as the configuration data (such as remote node/link IDs, SRLGs, administrative colors, etc.). TE link is connected to TE node, terminating the TE link via exactly one TE link termination point (LTP).

In Figure 1, Link-12 and Link-23 are TE links.

3.4. Transitional TE Link for Multi-Layer Topologies

Networks are typically composed of multiple network layers where one or multiple signals in the client layer network can be multiplexed and encapsulated into a server layer signal [RFC5212] [G.805]. The server layer signal can be carried in the server layer network across multiple nodes until the server layer signal is terminated and the client layer signals reappear in the node that terminates the server layer signal. Examples of multi-layer networks are: IP over MPLS over Ethernet, low order Optical Data Unit-k (ODUk) signals multiplexed into a high order ODUl (l>k) carried over an Optical Channel (OCh) signal in an optical transport network as defined in [G.872] and [G.709].

TE links as defined in Section 3.3. can be used to represent links within a network layer. In case of a multi-layer network, TE nodes and TE links only allow representation of each network layer as a separate TE topology. Each of these single layer TE topologies would be isolated from their client and their server layer TE topology, if present. The highest and the lowest network layer in the hierarchy only have a single adjacent layer below or above, respectively. Multiplexing of client layer signals and encapsulating them into a server layer signal requires a function that is provided inside a node (typically realized in hardware). This function is also called layer transition.

One of the key requirements for path computation is to be able to calculate a path between two endpoints across a multi-layer network based on the TE topology representing this multi-layer network. This means that an additional TE construct is needed that represents potential layer transitions in the multi-layer TE-topology that connects the TE-topologies representing each separate network layer. The so-called transitional TE link is such a construct and it represents the layer transition function residing inside a node that is decomposed into multiple logical nodes that are represented as TE nodes (see also the transitional link definition in [G.8080] for the optical transport network). Hence, a transitional TE link connects a client layer node with a server layer node. A TE link as defined in 3.3. has LTPs of exactly the same kind on each link end whereas the transitional TE link has client layer LTPs on the client side of the transitional link and in most cases a single server layer LTP on the server side. It should be noted that transitional links are a helper construct in the multi-layer TE topology and they only exist as long as they are not in use, as they represent potential connectivity. When the server layer trail has been established between the server layer LTP of two transitional links in the server layer network, the resulting client layer link in the data plane will be represented as a normal TE link in the client layer topology. The transitional TE links will re-appear when the server layer trail has been torn down.

```
                                |                 |
                                | +---+          ---
                                | |   | TE Node  \ / Transitional
                                | +---+           |  Link
                                |
                                | ----- Client Layer Link
                                | ===== Server Layer Link
                                | ***** Layer Boundary
                                o-----------------------------------

      +------------------+
      | +------+         |                          +------+
   -----|Client|------+  |              Client -----|Client|
      | |Layer |---+  |  |              Layer       |Layer |
   -----|Switch|-+ |  |  |              Links  -----|Node  |
      | +------+ | |  |  |                          +------+
      |          | |  |  | Client                    | |  |
      |          | | ---_| Layer                     --- ---
   ***|**********|*| \ /*|***************************\ /*\ /****
      |         ---   |  | Server       Transitional  |   |
      |    Layer \ /  |  | Layer               Links  |   |
      |    Term.  |   |  |                            |   |
      |           |   |  |                            |   |
      |         +------+ |                          +------+
   =============|Server|=====            Server ====|Server|====
      |         |Layer | |               Layer       |Layer |
   =============|Switch|=====            Links  ====|Node  |====
      |         +------+ |                          +------+
      +------------------+

       Physical Node View                       TE-Topology View
```

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

3.5. TE Link Termination Point (LTP)

TE link termination point (LTP) is a conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1.

In Figure 1, Node-2 has six LTPs: LTP-1 to LTP-6.

3.6. TE Tunnel Termination Point (TTP)

TE tunnel termination point (TTP) is an element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as

WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned a unique ID within the TE node scope. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node.

In Figure 1, Node-1 has two TTPs: TTP-1 and TTP-2.

3.7. TE Node Connectivity Matrix

TE node connectivity matrix is a TE node's attribute describing the TE node's switching limitations in a form of valid switching combinations of the TE node's LTPs (see below). From the point of view of a potential TE path arriving at the TE node at a given inbound LTP, the node's connectivity matrix describes valid (permissible) outbound LTPs for the TE path to leave the TE node from.

In Figure 1, the connectivity matrix on Node-2 is:
{<LTP-6, LTP-1>, <LTP-5, LTP-2>, <LTP-5, LTP-4>, <LTP-4, LTP-1>,
<LTP-3, LTP-2>}

3.8. TTP Local Link Connectivity List (LLCL)

TTP Local Link Connectivity List (LLCL) is a List of TE links terminated by the TTP hosting TE node (i.e. list of the TE link LTPs), which the TTP could be connected to. From the point of view of a potential TE path, LLCL provides a list of valid TE links the TE path needs to start/stop on for the connection, taking the TE path, to be successfully terminated on the TTP in question.

In Figure 1, the LLCL on Node-1 is:
{<TTP-1, LTP-5>, <TTP-1, LTP-2>, <TTP-2, LTP-3>, <TTP-2, LTP4>}

3.9. TE Path

TE path is an ordered list of TE links and/or TE nodes on the TE topology graph, inter-connecting a pair of TTPs to be taken by a potential connection. TE paths, for example, could be a product of successful path computation performed for a given transport service.

In Figure 1, the TE Path for TE-Tunnel-1 is:
{Node-1:TTP-1, Link-12, Node-2, Link-23, Node-3:TTP1}

3.10. TE Inter-Layer Lock

TE inter-layer lock is a modeling concept describing client-server layer adaptation relationships and hence important for the multi-layer traffic engineering. It is an association of M client layer LTPs and N server layer TTPs, within which data arriving at any of the client layer LTPs could be adopted onto any of the server layer TTPs. TE inter-layer lock is identified by inter-layer lock ID, which is unique across all TE topologies provided by the same provider. The client layer LTPs and the server layer TTPs associated within a given TE inter-layer lock are annotated with the same inter-layer lock ID attribute.

```
                                |
                                | +---+          __
                                | |   | TE Node  \/ TTP  o LTP
                                | +---+
                                |
                                | ----- TE Link
                                | ***** TTP Local Link Connectivity
                                o----------------------------------

             (IL-1) C-LTP-1 +------------+   C-LTP-2 (IL-1)
                    --------O   (IL-1)   O--------
             (IL-1) C-LTP-3 |   S-TTP-1  |   C-LTP-4 (IL-1)
                    --------O     __     0--------
             (IL-1) C-LTP-5 |    *\/*    |   C-LTP-5 (IL-1)
                    --------O   *    *   O--------
                            |  *(IL-1)*  |
                    S-LTP-3 | * S-TTP-2* |   S-LTP-4
                    --------o*    __    *o--------
                            |    *\/*    |
                            |   *    *   |
                            +--o------o--+
                       S-LTP-1 |      | S-LTP-2
```

Figure 3: TE Inter-Layer Lock ID Associations

On the picture above a TE inter-layer lock with IL_1 ID associates 6 client layer LTPs (C-LTP-1 - C-LTP-6) with two server layer TTPs (S-TTP-1 and S-TTP-2). They all have the same attribute - TE inter-layer lock ID: IL-1, which is the only thing that indicates the association. A given LTP may have 0, 1 or more inter-layer lock IDs. In the latter case this means that the data arriving at the LTP may be adopted onto any of TTPs associated with all specified inter-layer locks. For example, C-LTP-1 could have two inter-layer lock IDs - IL-1 and IL-2. This would mean that C-LTP-1 for adaptation purposes could use not just TTPs associated with inter-layer lock IL-1 (i.e.

S-TTP-1 and S-TTP-2 on the picture), but any of TTPs associated with inter-layer lock IL-2 as well. Likewise, a given TTP may have one or more inter-layer lock IDs, meaning that it can offer the adaptation service to any of client layer LTPs with inter-layer lock ID matching one of its own. Additionally, each TTP has an attribute - Unreserved Adaptation Bandwidth, which announces its remaining adaptation resources sharable between all potential client LTPs.

LTPs and TTPs associated within the same TE inter-layer lock may be hosted by the same (hybrid, multi-layer) TE node or multiple TE nodes located in the same or separate TE topologies. The latter is especially important since TE topologies of different layer networks could be modeled by separate augmentations of the basic (common to all layers) TE topology model.

3.11. Underlay TE topology

Underlay TE topology is a TE topology that serves as a base for constructing of overlay TE topologies

3.12. Overlay TE topology

Overlay TE topology is a TE topology constructed based on one or more underlay TE topologies. Each TE node of the overlay TE topology represents an arbitrary segment of an underlay TE topology; each TE link of the overlay TE topology represents an arbitrary TE path in one of the underlay TE topologies. The overlay TE topology and the supporting underlay TE topologies may represent distinct layer networks (e.g. OTN/ODUk and WDM/OCh respectively) or the same layer network.

3.13. Abstract TE topology

Abstract TE topology is a topology that contains abstract topological elements (nodes, links, tunnel termination points). Abstract TE topology is an overlay TE topology created by a topology provider and customized for a topology provider’s client based on one or more of the provider’s native TE topologies (underlay TE topologies), the provider’s policies and the client’s preferences. For example, a first level topology provider (such as Domain Controller) can create an abstract TE topology for its client (e.g. Multi-Domain Service Coordinator) based on the provider’s one or more native TE topologies, local policies/profiles and the client’s TE topology configuration requests

Figure 4 shows an example of abstract TE topology.

```
                                | +---+
                                | |   | TE Node
                                | +---+
                                | ----- TE Link
                                o-----------------------------------

     +---+              +---+
     |s31|--------------|S5 |
     +---+\          / +---+
          \        /
           \      /
            \+---+/               +---+
            /|AN1|\---------------|S8 |
           / +---+ \              +---+
     +---+ /         \ +---+
     |S9 |-------------|S11|
     +---+             +---+
           Abstract TE Topology

            +---+                    +---+
            |S1 |--------------------|S2 |
            +---+                    +---+
             /                          \
            /                            \
     +---+ /                  +---+      \ +---+
     |s3 |--------------------|S4 |---------|S5 |
     +---+\                   +---+         +---+
          \                   \             \
           \                   \             \
            \+---+               +---+         +---+
            /|S6 |\              |S7 |---------|S8 |
           / +---+ \             +---+\       /+---+
     +---+ /         \ +---+              +---+ /
     |S9 |-------------|S10|--------------|S11|/
     +---+             +---+              +---+
           Native TE Topology
```

Figure 4: Abstract TE Topology

4. Model Applicability

4.1. Native TE Topologies

The model discussed in this draft can be used to represent and retrieve native TE topologies on a given TE system.

```
                                |  +---+
                                |  |   | TE Node
                                |  +---+
                                |  ----- TE Link
                                o-----------------------------------

     +---+       +---+        +---+         +---+         +---+
     | R1|-------| R2|--------| R3|---------| R4|---------| R5|
     +---+       +---+        +---+         +---+         +---+
       |                       /   \         /   \         /
       |                      /     \       /     \       /
       |                     /       \     /       \     /
       |                    /         \   /         \   /
       |                   /           \ /           \ /
     +---+             +---+         +---+         +---+
     | R6|-------------| R7|         | R8|---------| R9|
     +---+             +---+         +---+         +---+
```

Figure 5a: Example Network Topology

Consider the network topology depicted in Figure 5a. R1 .. R9 are nodes representing routers. An implementation MAY choose to construct a native TE Topology using all nodes and links present in the given TED as depicted in Figure 5b. The data model proposed in this document can be used to retrieve/represent this TE topology.

```
          ---------------                    |  [ ] TE Node
          | Native      |                    |  +++ TE Link
          | TE-Topology |                    o--------------
          ---------------

      [R1] ++++ [R2] ++++ [R3] ++++ [R4] ++++ [R5]
        +                +    +    +    +    +
        +               +      +  +      +  +
        +              +        ++        ++
      [R6] +++++++++ [R7]      [R8] ++++ [R9]
```

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

Consider the case of the topology being split in a way that some nodes participate in OSPF-TE while others participate in ISIS-TE (Figure 6a). An implementation MAY choose to construct separate TE Topologies based on the information source. The native TE Topologies constructed using only nodes and links that were learnt via a specific information source are depicted in Figure 6b. The data model proposed in this document can be used to retrieve/represent these TE topologies.

Similarly, the data model can be used to represent/retrieve a TE Topology that is constructed using only nodes and links that belong to a particular technology layer. The data model is flexible enough to retrieve and represent many such native TE Topologies.

```
                                  :
TE info distributed via ISIS-TE   :  TE info distributed via OSPF-TE
                                  :
       +---+       +---+       +---+          +---+          +---+
       | R1|-------| R2|--------| R3|---------| R4|---------| R5|
       +---+       +---+       +---+          +---+          +---+
         |                     / : \           /   \           /
         |                    /  :  \         /     \         /
         |                   /   :   \       /       \       /
         |                  /    :    \     /         \     /
         |                 /     :     \ /           \ /
       +---+             +---+   :    +---+          +---+
       | R6|-------------| R7|   :    | R8|----------| R9|
       +---+             +---+   :    +---+          +---+
                                 :
```

Figure 6a: Example Network Topology

```
        -----------------------   :  -----------------------
        |Native TE Topology    |  :  |Native TE Topology    |
        |Info-Source: ISIS-TE  |  :  |Info-Source: OSPF-TE  |
        -----------------------   :  -----------------------
                                  :
         [R1] ++++ [R2] ++++ [R3] : [R3'] ++++ [R4] ++++ [R5]
          +                +      :      +    +    +    +
          +               +       :       +  +      +  +
          +              +        :        ++        ++
         [R6] +++++++++ [R7]      :       [R8] ++++ [R9]
```

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

4.2. Customized TE Topologies

Customized TE topology is a topology that was modified by the provider to honor a particular client's requirements or preferences. The model discussed in this draft can be used to represent, retrieve and manipulate customized TE Topologies. The model allows the provider to present the network in abstract TE Terms on a per client basis. These customized topologies contain sufficient information for the path computing client to select paths according to its policies.

```
                                  | +---+            /-\
                                  | |   | Router    (   ) WDM
                                  | +---+ Node       \-/  node
                                  |
                                  o----------------------------

     +---+        /-\          /-\           /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/          \-/           \-/          +---+
                              /   \         /   \
                             /     \       /     \
                            /       \     /       \
                           /         \   /         \
                          /           \ /           \
         +---+          /-\           /-\           /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

Figure 7: Example packet optical topology

Consider the network topology depicted in Figure 7. This is a typical packet optical transport deployment scenario where the WDM layer network domain serves as a Server Network Domain providing transport connectivity to the packet layer network Domain (Client Network Domain). Nodes R1, R2, R3 and R4 are IP routers that are connected to an Optical WDM transport network. A, B, C, D, E and F are WDM nodes that constitute the Server Network Domain.

```
                                            | *****  B-F WDM Path
                                            | @@@@@  B-E WDM Path
                                            | $$$$$  A-E WDM Path
                                            o--------------------

     +---+        /-\ $$$$$$$$ /-\ $$$$$$$$$ /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/         @\-/ @@@@@@@@@ \-/          +---+
                             @/   \         /   \
                            @/     \       /     \
                           @/       \     /       \
                          @/         \   /         \
                         @/           \ /           \
         +---+          /-\ ********* /-\ ********* /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

Figure 8a: Paths within the provider domain

```
              ++++++++ [A] ++++++++++++++++++++ [E] +++++++++
                                            +++++
                                        ++++
                                    ++++
                                ++++
                            ++++
              ++++++++ [B] ++++++++++++++++++++ [F] +++++++++
```

Figure 8b: Customized TE Topology provided to the Client

The goal here is to augment the Client TE Topology with a customized TE Topology provided by the WDM network. Given the availability of the paths A-E, B-F and B-E (Figure 8a), a customized TE Topology as depicted in Figure 8b is provided to the Client. This customized TE Topology is merged with the Client's Native TE Topology and the resulting topology is depicted in Figure 8c.

```
         [R1] ++++++++ [A] ++++++++++++++++++++ [E] +++++++++ [R3]
                                            +++++
                                        ++++
                                    ++++
                                ++++
                            ++++
         [R2] ++++++++ [B] ++++++++++++++++++++ [F] +++++++++ [R4]
```

Figure 8c: Customized TE Topology merged with the Client's Native TE Topology

The data model proposed in this document can be used to retrieve/represent/manipulate the customized TE Topology depicted in Figure 8b.

A customized TE topology is not necessarily an abstract TE topology. The provider may produce, for example, an abstract TE topology of certain type (e.g. single-abstract-node-with-connectivity-matrix topology, a border-nodes-connected-via-mesh-of-abstract-links topology, etc.) and expose it to all/some clients in expectation that the clients will use it without customization.
On the other hand, a client may request a customized version of the provider's native TE topology (e.g. by requesting removal of TE links

which belong to certain layers, are too slow, not protected and/or have a certain affinity). Note that the resulting TE topology will not be abstract (because it will not contain abstract elements), but customized (modified upon client's instructions).

The client ID field in the TE topology identifier (Section 5.4.) indicates which client the TE topology is customized for. Although an authorized client MAY receive a TE topology with the client ID field matching some other client, the client can customize only TE topologies with the client ID field either 0 or matching the ID of the client in question. If the client starts reconfiguration of a topology its client ID will be automatically set in the topology ID field for all future configurations and updates wrt. the topology in question.

The provider MAY tell the client that a given TE topology cannot be re-negotiated, by setting its own (provider's) ID in the client ID field of the topology ID.

Even though this data model allows to access TE topology information across clients, implementations MAY restrict access for particular clients to particular data fields. The Network Configuration Access Control Model (NACM) [RFC8341] provides such a mechanism.

4.3. Merging TE Topologies Provided by Multiple Providers

A client may receive TE topologies provided by multiple providers, each of which managing a separate domain of multi-domain network. In order to make use of said topologies, the client is expected to merge the provided TE topologies into one or more client's native TE topologies, each of which homogeneously representing the multi-domain network. This makes it possible for the client to select end-to-end TE paths for its services traversing multiple domains.

In particular, the process of merging TE topologies includes:

- Identifying neighboring domains and locking their topologies horizontally by connecting their inter-domain open-ended TE links;
- Renaming TE node, link, and SRLG IDs to ones allocated from a separate name space; this is necessary because all TE topologies are considered to be, generally speaking, independent with a possibility of clashes among TE node, link or SRLG IDs;
- Locking, vertically, TE topologies associated with different layer networks, according to provided topology inter-layer locks; this is to facilitate inter-layer path computations across multiple TE topologies provided by the same topology provider.

```
     /---\      +---+    +---+      +---+    +---+      /---\
     |s3 |------|S13|----|S15|------|S23|----|S25|------|C21|
     \---/      +---+\   +---+      +---+   /+---+      \---/
                      \                    /
                       \                  /
                        \+---+      +---+/   +---+      /---\
                         |S18|------|S24|    |S28|------|C22|
                         +---+      +---+\  /+---+      \---/
                                          \/
                                          /\
     /---\      +---+    +---+      +---+/  \+---+      /---\
     |C12|------|S19|----|S17|------|S29|----|S27|------|C23|
     \---/      +---+    +---+      +---+    +---+      \---/


        Domain 1 TE Topology              Domain 2 TE Topology

           +---+    +---+                +---+    +---+
      -----|S13|----|S15|----        ----|S23|----|S25|----
           +---+\   +---+                +---+   /+---+
                 \                              /
                  \                            /
                   \+---+                +---+/   +---+
                    |S18|----        ----|S24|    |S28|----
                    +---+                +---+\  /+---+
                                               \/
                                               /\
           +---+    +---+                +---+/  \+---+
      -----|S19|----|S17|----        ----|S29|----|S27|----
           +---+    +---+                +---+    +---+
```

Figure 9: Merging Domain TE Topologies

Figure 9 illustrates the process of merging, by the client, of TE topologies provided by the client's providers. In the Figure, each of the two providers caters to the client (abstract or native) TE topology, describing the network domain under the respective provider's control. The client, by consulting the attributes of the inter-domain TE links - such as inter-domain plug IDs or remote TE node/link IDs (as defined by the TE Topology model) - is able to determine that:

a) the two domains are adjacent and are inter-connected via three inter-domain TE links, and;

b) each domain is connected to a separate customer site, connecting the left domain in the Figure to customer devices C-11 and C-12, and the right domain to customer devices C-21, C-22 and C-23.

Therefore, the client inter-connects the open-ended TE links, as shown on the upper part of the Figure.

As mentioned, one way to inter-connect the open-ended inter-domain TE links of neighboring domains is to mandate the providers to specify remote nodeID/linkID attribute in the provided inter-domain TE links. This, however, may prove to be not flexible. For example, the providers may not know the respective remote nodeIDs/ linkIDs. More importantly, this option does not allow for the client to mix-n-match multiple (more than one) topologies catered by the same providers (see below). Another, more flexible, option to resolve the open-ended inter-domain TE links is by annotating them with the inter-domain plug ID attribute. Inter-domain plug ID is a network-wide unique number that identifies on the network a connectivity supporting a given inter-domain TE link. Instead of specifying remote node ID/link ID, an inter-domain TE link may provide a non-zero inter-domain plug ID. It is expected that two neighboring domain TE topologies (provided by separate providers) will have each at least one open-ended inter-domain TE link with an inter-domain plug ID matching to one provided by its neighbor. For example, the inter-domain TE link originating from node S15 of the Domain 1 TE topology (Figure 9) and the inter-domain TE link coming from node S23 of Domain 2 TE topology may specify matching inter-domain plug ID (e.g. 175344). This allows for the client to identify adjacent nodes in the separate neighboring TE topologies and resolve the inter-domain TE links connecting them regardless of their respective nodeIDs/linkIDs (which, as mentioned, could be allocated from independent name spaces). Inter-domain plug IDs may be assigned and managed by a central network authority. Alternatively, inter-domain plug IDs could be dynamically auto-discovered (e.g. via LMP protocol).

Furthermore, the client renames the TE nodes, links and SRLGs offered in the abstract TE topologies by assigning to them IDs allocated from a separate name space managed by the client. Such renaming is necessary, because the two abstract TE topologies may have their own name spaces, generally speaking, independent one from another; hence, ID overlaps/clashes are possible. For example, both TE topologies have TE nodes named S7, which, after renaming, appear in the merged TE topology as S17 and S27, respectively.

Once the merging process is complete, the client can use the merged TE topology for path computations across both domains, for example, to compute a TE path connecting C-11 to C-23.

4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same Provider

```
      Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1

             +---+    +---+                  +---+    +---+
        -----|S13|----|S15|----          ----|S23|----|S25|----
             +---+\   +---+                  +---+   /+---+
                   \                                /
                    \                              /
                     \+---+                  +---+/   +---+
                      |S18|----          ----|S24|    |S28|----
                      +---+                  +---+\  /+---+
                                                   \/
                                                   /\
             +---+    +---+                  +---+/  \+---+
        -----|S19|----|S17|----          ----|S29|----|S27|----
             +---+    +---+                  +---+    +---+


      Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1

             +------------+                  +------------+
        -----|            |----          ----|            |----
             |            |                  |            |
             |    AN1     |----          ----|    AN1     |----
             |            |                  |            |
        -----|            |----          ----|            |----
             +------------+                  +------------+
```

Figure 10: Merging Domain TE Topologies

Based on local configuration, templates and/or policies pushed by the client, a given provider may expose more than one abstract TE topology to the client. For example, one abstract TE topology could be optimized based on a lowest-cost criterion, while another one could be based on best possible delay metrics, while yet another one could be based on maximum bandwidth availability for the client services. Furthermore, the client may request all or some providers to expose additional abstract TE topologies, possibly of a different type and/or optimized differently, as compared to already-provided TE topologies. In any case, the client should be prepared for a provider to offer to the client more than one abstract TE topology.

It should be up to the client (based on the client's local configuration and/or policies conveyed to the client by the client's clients) to decide how to mix-and-match multiple abstract TE topologies provided by each or some of the providers, as well as how to merge them into the client's native TE topologies. The client also decides how many such merged TE topologies it needs to produce and maintain. For example, in addition to the merged TE topology depicted in the upper part of Figure 9, the client may merge the abstract TE topologies received from the two providers, as shown in Figure 10, into the client's additional native TE topologies, as shown in Figure 11.

Note that allowing for the client mix-n-matching of multiple TE topologies assumes that inter-domain plug IDs (rather than remote nodeID/linkID) option is used for identifying neighboring domains and inter-domain TE link resolution.

```
                Client's Merged TE Topology 2

     /---\      +------------+      +------------+      /---\
     |s3 |------|            |------|            |------|C21|
     \---/      |            |      |            |      \---/
                |            |      |            |
                |            |      |            |      /---\
                |    AN11    |------|    AN21    |------|C22|
                |            |      |            |      \---/
                |            |      |            |
     /---\      |            |      |            |      /---\
     |C12|------|            |------|            |------|C23|
     \---/      +------------+      +------------+      \---/


                Client's Merged TE Topology 3

     /---\      +------------+      +---+    +---+      /---\
     |s3 |------|            |------|S23|----|S25|------|C21|
     \---/      |            |      +---+   /+---+      \---/
                |            |             /
                |            |            /
                |            |      +---+/   +---+      /---\
                |    AN11    |------|S24|    |S28|------|C22|
                |            |      +---+\  /+---+      \---/
                |            |            \/
                |            |            /\
     /---\      |            |      +---+/  \+---+      /---\
     |C12|------|            |------|S29|----|S27|------|C23|
     \---/      +------------+      +---+    +---+      \---/
```

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

It is important to note that each of the three native (merged) TE topologies could be used by the client for computing TE paths for any of the multi-domain services. The choice as to which topology to use for a given service depends on the service parameters/requirements and the topology's style, optimization criteria and the level of details.

5. Modeling Considerations

5.1. Network topology building blocks

The network topology building blocks are discussed in [RFC8345]. The TE Topology model proposed in this document augments and uses the ietf-network-topology module defined in [RFC8345].

```
+------------------------+
|                        |
| Network Topology Model |
|  (ietf-network-topology)|
+------------------------+
             |
             |
             |
             V
+------------------------+
|       TE Topology      |
|         Model          |
|                        |
+------------------------+
```

Figure 12: Augmenting the Network Topology Model

5.2. Technology agnostic TE Topology model

The TE Topology model proposed in this document is meant to be network technology agnostic. Other technology specific TE Topology models can augment and use the building blocks provided by the proposed model.

```
                  +------------------------------+
                  |       TE Topology Model      |
                  |  (Defined in This Document)  |
                  +------------------------------+
                                 |
            +-------------+-------------+-------------+
            |             |             |             |
            V             V             V             V
      +------------+                              +------------+
      | Technology |                              | Technology |
      |  Specific  |     ......................   |  Specific  |
      | TE Topology|                              | TE Topology|
      |   Model 1  |                              |   Model n  |
      +------------+                              +------------+
```

Figure 13: Augmenting the Technology agnostic TE Topology model

5.3. Model Structure

The high-level model structure proposed by this document is as shown below:

```
module: ietf-te-topology
augment /nw:networks/nw:network/nw:network-types:
   +--rw te-topology!

augment /nw:networks:
   +--rw te!
      +--rw templates
         +--rw node-template* [name] {template}?
         |  ............
         +--rw link-template* [name] {template}?
            ............

augment /nw:networks/nw:network:
   +--rw te-topology-identifier
   |  +--rw provider-id?   te-global-id
   |  +--rw client-id?     te-global-id
   |  +--rw topology-id?   te-topology-id
   +--rw te!
      |  ............

augment /nw:networks/nw:network/nw:node:
   +--rw te-node-id?   te-types:te-node-id
   +--rw te!
      |  ............
      +--rw tunnel-termination-point* [tunnel-tp-id]
```

```
            +--rw tunnel-tp-id    binary
            |  ............
            +--rw supporting-tunnel-termination-point* [node-ref tunnel-
   tp-ref]
               |  ............

   augment /nw:networks/nw:network/nt:link:
      +--rw te!
         |  ..........

   augment /nw:networks/nw:network/nw:node/nt:termination-point:
      +--rw te-tp-id?   te-types:te-tp-id
      +--rw te!
         |  ............
```

5.4. Topology Identifiers

The TE-Topology is uniquely identified by a key that has 3 constituents - topology-id, provider-id and client-id. The combination of provider-id and topology-id uniquely identifies a native TE Topology on a given provider. The client-id is used only when Customized TE Topologies come into play; a value of "0" is used as the client-id for native TE Topologies.

```
   augment /nw:networks/nw:network:
      +--rw te-topology-identifier
      |  +--rw provider-id?   te-global-id
      |  +--rw client-id?     te-global-id
      |  +--rw topology-id?   te-topology-id
      +--rw te!
         |  ............
```

5.5. Generic TE Link Attributes

The model covers the definitions for generic TE Link attributes - bandwidth, admin groups, SRLGs, switching capabilities, TE metric extensions etc.

```
     +--rw te-link-attributes
        .....................
        +--rw admin-status?                      te-admin-status
        |  .....................
        +--rw link-index?                        uint64
        +--rw administrative-group?              te-types:admin-groups
        +--rw link-protection-type?              enumeration
        +--rw max-link-bandwidth?                te-bandwidth
```

```
    +--rw max-resv-link-bandwidth?          te-bandwidth
    +--rw unreserved-bandwidth* [priority]
    |  .....................
    +--rw te-default-metric?                uint32
    |  .....................
    +--rw te-srlgs
    +--rw te-nsrlgs {nsrlg}?       .....................
```

5.6. Generic TE Node Attributes

The model covers the definitions for generic TE Node attributes.

The definition of a generic connectivity matrix is shown below:

```
 +--rw te-node-attributes
    ...........
    +--rw connectivity-matrices
    ...........
    |  +--rw connectivity-matrix* [id]
    |  |  +--rw id             uint32
    |  |  +--rw from
    |  |  |  +--rw tp-ref?        leafref
    |  |  |  +--rw label-restrictions
    |  |  +--rw to
    |  |  |  +--rw tp-ref?        leafref
    |  |  |  +--rw label-restrictions
    |  |  +--rw is-allowed?   boolean
    ...........
    |  |  +--rw underlay! {te-topology-hierarchy}?
    ...........
    |  |  +--rw path-constraints
    ...........
    |  |  +--rw optimizations
    ...........
    |  |  +--ro path-properties
    ...........
```

The definition of a TTP Local Link Connectivity List is shown below:

```
 +--rw tunnel-termination-point* [tunnel-tp-id]
    +--rw tunnel-tp-id              binary
    +--rw admin-status?             te-types:te-admin-status
    +--rw name?                     string
    +--rw switching-capability?     identityref
    +--rw encoding?                 identityref
    +--rw inter-layer-lock-id*      uint32
```

```
        +--rw protection-type?           Identityref
        +--rw client-layer-adaptation
        ...........
        +--rw local-link-connectivities
        ...........
        |  +--rw local-link-connectivity* [link-tp-ref]
        |     +--rw link-tp-ref              leafref
        |     +--rw label-restrictions
        ...........
        |     +--rw is-allowed?               boolean
        |     +--rw underlay {te-topology-hierarchy}?
        ...........
        |     +--rw path-constraints
        ...........
        |     +--rw optimizations
        ...........
        |     +--ro path-properties
        ...........
        +--rw supporting-tunnel-termination-point* [node-ref tunnel-tp-
   ref]
           +--rw node-ref         inet:uri
           +--rw tunnel-tp-ref    binary
```

The attributes directly under container connectivity-matrices are the default attributes for all connectivity-matrix entries when the per entry corresponding attribute is not specified. When a per entry attribute is specified, it overrides the corresponding attribute directly under the container connectivity-matrices. The same rule applies to the attributes directly under container local-link-connectivities.

Each TTP (Tunnel Termination Point) MAY be supported by one or more supporting TTPs. If the TE node hosting the TTP in question refers to a supporting TE node, then the supporting TTPs are hosted by the supporting TE node. If the TE node refers to an underlay TE topology, the supporting TTPs are hosted by one or more specified TE nodes of the underlay TE topology.

5.7. TED Information Sources

The model allows each TE topological element to have multiple TE information sources (OSPF-TE, ISIS-TE, BGP-LS, User-Configured, System-Processed, Other). Each information source is associated with a credibility preference to indicate precedence. In scenarios where a customized TE Topology is merged into a Client's native TE Topology, the merged topological elements would point to the corresponding customized TE Topology as its information source.

```
   augment /nw:networks/nw:network/nw:node:
      +--rw te!
         ...........
         +--ro information-source?            te-info-source
         +--ro information-source-instance?   string
         +--ro information-source-state
         |  +--ro credibility-preference?    uint16
         |  +--ro logical-network-element?   string
         |  +--ro network-instance?          string
         |  +--ro topology
         |     +--ro node-ref?      leafref
         |     +--ro network-ref?   leafref
         +--ro information-source-entry*
         |       [information-source information-source-instance]
         |  +--ro information-source             te-info-source
         |  +--ro information-source-instance    string
            ............

   augment /nw:networks/nw:network/nt:link:
      +--rw te!
         ...........
         +--ro information-source?            te-info-source
         +--ro information-source-instance?   string
         +--ro information-source-state
         |  +--ro credibility-preference?    uint16
         |  +--ro logical-network-element?   string
         |  +--ro network-instance?          string
         |  +--ro topology
         |     +--ro link-ref?      leafref
         |     +--ro network-ref?   leafref
         +--ro information-source-entry*
         |       [information-source information-source-instance]
         |  +--ro information-source                te-info-source
         |  +--ro information-source-instance       string
            ............
```

5.8. Overlay/Underlay Relationship

The model captures overlay and underlay relationship for TE nodes/links. For example - in networks where multiple TE Topologies are built hierarchically, this model allows the user to start from a specific topological element in the top most topology and traverse all the way down to the supporting topological elements in the bottom most topology.

This relationship is captured via the "underlay-topology" field for the node and via the "underlay" field for the link. The use of these fields is optional and this functionality is tagged as a "feature" ("te-topology-hierarchy").

```
augment /nw:networks/nw:network/nw:node:
   +--rw te-node-id?   te-types:te-node-id
   +--rw te!
      +--rw te-node-template*           leafref {template}?
      +--rw te-node-attributes
      |  +--rw admin-status?            te-types:te-admin-status
      |  |  ....................
      |  +--rw underlay-topology {te-topology-hierarchy}?
      |     +--rw network-ref?   leafref

augment /nw:networks/nw:network/nt:link:
   +--rw te!
      +--rw te-link-attributes
      |  ....................
      |  +--rw underlay {te-topology-hierarchy}?
      |  |  +--rw enabled?                      boolean
      |  |  +--rw primary-path
      |  |  |  +--rw network-ref?    leafref
      |  |  |     ....................
      |  |  +--rw backup-path* [index]
      |  |  |  +--rw index           uint32
      |  |  |  +--rw network-ref?    leafref
      |  |  |     ....................
      |  |  +--rw protection-type?            identityref
      |  |  +--rw tunnel-termination-points
      |  |  |  +--rw source?        binary
      |  |  |  +--rw destination?   binary
      |  |  +--rw tunnels
      |  |  |  ....................
```

5.9. Templates

The data model provides the users with the ability to define templates and apply them to link and node configurations. The use of "template" configuration is optional and this functionality is tagged as a "feature" ("template").

```
augment /nw:networks/nw:network/nw:node:
   +--rw te-node-id?   te-types:te-node-id
   +--rw te!
      +--rw te-node-template*
      |       -> ../../../../te/templates/node-template/name
      |       {template}?
```

```
augment /nw:networks/nw:network/nt:link:
   +--rw te!
      +--rw te-link-template*
      |       -> ../../../../te/templates/link-template/name
      |       {template}?

augment /nw:networks:
   +--rw te!
      +--rw templates
         +--rw node-template* [name] {template}?
         |  +--rw name
         |  |       te-types:te-template-name
         |  +--rw priority?                  uint16
         |  +--rw reference-change-policy?   enumeration
         |  +--rw te-node-attributes
            ..........
         +--rw link-template* [name] {template}?
            +--rw name
            |       te-types:te-template-name
            +--rw priority?                  uint16
            +--rw reference-change-policy?   enumeration
            +--rw te-link-attributes
            ..........
```

Multiple templates can be specified to a configuration element. When two or more templates specify values for the same configuration field, the value from the template with the highest priority is used. The range of the priority is from 0 to 65535, with a lower number indicating a higher priority. The reference-change-policy specifies the action that needs to be taken when the template changes on a configuration element that has a reference to this template. The choices of action include taking no action, rejecting the change to the template and applying the change to the corresponding configuration.

5.10. Scheduling Parameters

The model allows time scheduling parameters to be specified for each topological element or for the topology as a whole. These parameters allow the provider to present different topological views to the client at different time slots. The use of "scheduling parameters" is optional.

The YANG data model for configuration scheduling is defined in [I-D.liu-netmod-yang-schedule], which allows specifying configuration schedules without altering this data model.

5.11. Notifications

Notifications are a key component of any topology data model.

[I-D.ietf-netconf-subscribed-notifications] and [I-D.ietf-netconf-yang-push] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

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

6. Guidance for Writing Technology Specific TE Topology Augmentations

The TE topology model defined in this document is technology agnostic as it defines concepts, abstractions and attributes that are common across multiple network technologies. It is envisioned that this base model will be widely used when defining technology specific TE topology models for various layer networks. [I-D.ietf-ccamp-wson-yang], [I-D.ietf-ccamp-otn-topo-yang], and [I-D.ietf-teas-yang-l3-te-topo] are some examples of technology specific TE Topology models. Writers of such models are encouraged to augment the basic TE topology model's containers, such as TE Topology, TE Node, TE Link, Link Termination Point (LTP), Tunnel Termination Point (TTP), Bandwidth and Label with the layer specific attributes instead of defining new containers.

Consider the following technology specific example-topology model:

```
module: example-topology
  augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
    +--rw example-topology!
  augment /nw:networks/nw:network/tet:te:
    +--rw attributes
       +--rw attribute-1?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes:
    +--rw attributes
       +--rw attribute-2?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices:
    +--rw attributes
       +--rw attribute-3?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
```

```
              /tet:te-node-attributes/tet:connectivity-matrices
              /tet:connectivity-matrix:
       +--rw attributes
          +--rw attribute-3?   uint8
     augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point:
       +--rw attributes
          +--rw attribute-4?   uint8
     augment /nw:networks/nw:network/nw:node/nt:termination-point
              /tet:te:
       +--rw attributes
          +--rw attribute-5?   uint8
     augment /nw:networks/nw:network/nt:link/tet:te
              /tet:te-link-attributes:
       +--rw attributes
          +--rw attribute-6?   uint8
```

The technology specific TE bandwidth for this example topology can be specified using the following augment statements:

```
     augment /nw:networks/tet:te/tet:templates/tet:link-template
              /tet:te-link-attributes
              /tet:interface-switching-capability/tet:max-lsp-bandwidth
              /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/tet:te/tet:templates/tet:link-template
              /tet:te-link-attributes/tet:max-link-bandwidth
              /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/tet:te/tet:templates/tet:link-template
              /tet:te-link-attributes/tet:max-resv-link-bandwidth
              /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/tet:te/tet:templates/tet:link-template
```

```
             /tet:te-link-attributes/tet:unreserved-bandwidth
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:path-constraints/tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:path-constraints
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:path-constraints/tet:te-bandwidth/tet:technology:
     +--:(example)
        +--ro example
           +--ro bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:path-constraints
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--ro example
           +--ro bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point/tet:client-layer-adaptation
             /tet:switching-capability/tet:te-bandwidth
             /tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:path-constraints
```

```
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:path-constraints
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes
               /tet:interface-switching-capability/tet:max-lsp-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:max-link-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:max-resv-link-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry
               /tet:interface-switching-capability/tet:max-lsp-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry/tet:max-link-bandwidth
               /tet:te-bandwidth/tet:technology:
```

```
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry/tet:max-resv-link-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry/tet:unreserved-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te
               /tet:interface-switching-capability/tet:max-lsp-bandwidth
               /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
```

The technology specific TE label for this example topology can be specified using the following augment statements:

```
     augment /nw:networks/tet:te/tet:templates/tet:link-template
               /tet:te-link-attributes/tet:underlay/tet:primary-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/tet:te/tet:templates/tet:link-template
               /tet:te-link-attributes/tet:underlay/tet:backup-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/tet:te/tet:templates/tet:link-template
```

```
              /tet:te-link-attributes/tet:label-restrictions
              /tet:label-restriction/tet:label-start/tet:te-label
              /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/tet:te/tet:templates/tet:link-template
              /tet:te-link-attributes/tet:label-restrictions
              /tet:label-restriction/tet:label-end/tet:te-label
              /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
              /tet:te-node-attributes/tet:connectivity-matrices
              /tet:label-restrictions/tet:label-restriction
              /tet:label-start/tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
              /tet:te-node-attributes/tet:connectivity-matrices
              /tet:label-restrictions/tet:label-restriction
              /tet:label-end/tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
              /tet:te-node-attributes/tet:connectivity-matrices
              /tet:underlay/tet:primary-path/tet:path-element/tet:type
              /tet:label/tet:label-hop/tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
              /tet:te-node-attributes/tet:connectivity-matrices
              /tet:underlay/tet:backup-path/tet:path-element/tet:type
              /tet:label/tet:label-hop/tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
```

```
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices
            /tet:path-properties/tet:path-route-objects
            /tet:path-route-object/tet:type/tet:label/tet:label-hop
            /tet:te-label/tet:technology:
    +--:(example)
       +--ro example
          +--ro label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices
            /tet:connectivity-matrix/tet:from/tet:label-restrictions
            /tet:label-restriction/tet:label-start/tet:te-label
            /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices
            /tet:connectivity-matrix/tet:from/tet:label-restrictions
            /tet:label-restriction/tet:label-end/tet:te-label
            /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices
            /tet:connectivity-matrix/tet:to/tet:label-restrictions
            /tet:label-restriction/tet:label-start/tet:te-label
            /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
            /tet:te-node-attributes/tet:connectivity-matrices
            /tet:connectivity-matrix/tet:to/tet:label-restrictions
            /tet:label-restriction/tet:label-end/tet:te-label
            /tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
```

```
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:underlay/tet:primary-path
             /tet:path-element/tet:type/tet:label/tet:label-hop
             /tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:underlay/tet:backup-path
             /tet:path-element/tet:type/tet:label/tet:label-hop
             /tet:te-label/tet:technology:
    +--:(example)
       +--rw example
          +--rw label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:path-properties
             /tet:path-route-objects/tet:path-route-object/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
    +--:(example)
       +--ro example
          +--ro label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:label-restrictions/tet:label-restriction
             /tet:label-start/tet:te-label/tet:technology:
    +--:(example)
       +--ro example
          +--ro label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:label-restrictions/tet:label-restriction
             /tet:label-end/tet:te-label/tet:technology:
    +--:(example)
       +--ro example
          +--ro label-1?   uint32
  augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:underlay/tet:primary-path/tet:path-element/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
```

```
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:underlay/tet:backup-path/tet:path-element/tet:type
               /tet:label/tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:path-properties/tet:path-route-objects
               /tet:path-route-object/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:to/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
```

```
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:to/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:underlay/tet:primary-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:underlay/tet:backup-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:path-properties
               /tet:path-route-objects/tet:path-route-object/tet:type
               /tet:label/tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
```

```
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities/tet:underlay
               /tet:primary-path/tet:path-element/tet:type/tet:label
               /tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities/tet:underlay
               /tet:backup-path/tet:path-element/tet:type/tet:label
               /tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities/tet:path-properties
               /tet:path-route-objects/tet:path-route-object/tet:type
               /tet:label/tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
```

```
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:underlay
               /tet:primary-path/tet:path-element/tet:type/tet:label
               /tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:underlay/tet:backup-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:tunnel-termination-point
               /tet:local-link-connectivities
               /tet:local-link-connectivity/tet:path-properties
               /tet:path-route-objects/tet:path-route-object/tet:type
               /tet:label/tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
```

```
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:underlay/tet:primary-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:te-link-attributes/tet:underlay/tet:backup-path
               /tet:path-element/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nt:link/tet:te
               /tet:information-source-entry/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
```

The YANG module to implement the above example topology can be seen in Appendix C.

7. TE Topology YANG Module

This module references [RFC1195], [RFC3209], [RFC3272], [RFC3471], [RFC3630], [RFC3785], [RFC4201], [RFC4202], [RFC4203], [RFC4206], [RFC4872], [RFC5152], [RFC5212], [RFC5305], [RFC5316], [RFC5329], [RFC5392], [RFC6001], [RFC6241], [RFC6991], [RFC7308], [RFC7471], [RFC7579], [RFC7752], [RFC8345], and [I-D.ietf-teas-yang-te-types].

```
<CODE BEGINS> file "ietf-te-topology@2019-02-07.yang"
module ietf-te-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology";

  prefix "tet";

  import ietf-yang-types {
    prefix "yang";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-inet-types {
    prefix "inet";
    reference "RFC 6991: Common YANG Data Types";
  }

  import ietf-te-types {
    prefix "te-types";
    reference
      "I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
       Types";
  }

  import ietf-network {
    prefix "nw";
    reference "RFC 8345: A YANG Data Model for Network Topologies";
  }

  import ietf-network-topology {
    prefix "nt";
    reference "RFC 8345: A YANG Data Model for Network Topologies";
  }
```

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

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

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

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

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

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

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

        Editor:   Oscar Gonzalez De Dios
                  <mailto:oscar.gonzalezdedios@telefonica.com>";

     description
       "TE topology model for representing and manipulating technology
        agnostic TE Topologies.

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

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

        This version of this YANG module is part of RFC XXXX; see the
```

```
        RFC itself for full legal notices.";

     revision "2019-02-07" {
       description "Initial revision";
       reference "RFC XXXX: YANG Data Model for TE Topologies";
     // RFC Ed.: replace XXXX with actual RFC number and remove
     // this note
     }

     /*
      * Features
      */
     feature nsrlg {
       description
         "This feature indicates that the system supports NSRLG
          (Not Sharing Risk Link Group).";
     }

     feature te-topology-hierarchy {
       description
         "This feature indicates that the system allows underlay
          and/or overlay TE topology hierarchy.";
     }

     feature template {
       description
         "This feature indicates that the system supports
          template configuration.";
     }

     /*
      * Typedefs
      */
     typedef geographic-coordinate-degree {
         type decimal64 {
           fraction-digits 8;
         }
         description
           "Decimal degree (DD) used to express latitude and longitude
            geographic coordinates.";
     } // geographic-coordinate-degree
```

```
     typedef te-info-source {
       type enumeration {
         enum "unknown" {
           description "The source is unknown.";
         }
         enum "locally-configured" {
           description "Configured entity.";
         }
         enum "ospfv2" {
           description "OSPFv2.";
         }
         enum "ospfv3" {
           description "OSPFv3.";
         }
         enum "isis" {
           description "ISIS.";
         }
         enum "bgp-ls" {
           description "BGP-LS.";
           reference
             "RFC 7752: North-Bound Distribution of Link-State and
              Traffic Engineering (TE) Information Using BGP";
         }
         enum "system-processed" {
           description "System processed entity.";
         }
         enum "other" {
           description "Other source.";
         }
       }
       description
         "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

PCE Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2017

D. Dhody, Ed.
Huawei Technologies
J. Hardwick
Metaswitch
V. Beeram
Juniper Networks
J. Tantsura

July 7, 2016

A YANG Data Model for Path Computation Element Communications Protocol (PCEP)

draft-pkd-pce-pcep-yang-06

Abstract

This document defines a YANG data model for the management of Path Computation Element communications Protocol (PCEP) for communications between a Path Computation Client (PCC) and a Path Computation Element (PCE), or between two PCEs. The data model includes configuration data and state data (status information and counters for the collection of statistics).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Path Computation Element (PCE) defined in [RFC4655] is an entity that is capable of computing a network path or route based on a network graph, and applying computational constraints. A Path

Computation Client (PCC) may make requests to a PCE for paths to be computed.

PCEP is the communication protocol between a PCC and PCE and is defined in [RFC5440]. PCEP interactions include path computation requests and path computation replies as well as notifications of specific states related to the use of a PCE in the context of Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) Traffic Engineering (TE). [I-D.ietf-pce-stateful-pce] specifies extensions to PCEP to enable stateful control of MPLS TE LSPs.

This document defines a YANG [RFC6020] data model for the management of PCEP speakers. It is important to establish a common data model for how PCEP speakers are identified, configured, and monitored. The data model includes configuration data and state data (status information and counters for the collection of statistics).

This document contains a specification of the PCEP YANG module, "ietf-pcep" which provides the PCEP [RFC5440] data model.

2. Requirements Language

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

3. Terminology and Notation

This document uses the terminology defined in [RFC4655] and [RFC5440]. In particular, it uses the following acronyms.

- o Path Computation Request message (PCReq).
- o Path Computation Reply message (PCRep).
- o Notification message (PCNtf).
- o Error message (PCErr).
- o Request Parameters object (RP).
- o Synchronization Vector object (SVEC).
- o Explicit Route object (ERO).

This document also uses the following terms defined in [RFC7420]:

- o PCEP entity: a local PCEP speaker.

- o PCEP peer: to refer to a remote PCEP speaker.

- o PCEP speaker: where it is not necessary to distinguish between local and remote.

Further, this document also uses the following terms defined in [I-D.ietf-pce-stateful-pce] :

- o Stateful PCE, Passive Stateful PCE, Active Stateful PCE

- o Delegation, Revocation, Redelegation

- o LSP State Report, Path Computation Report message (PCRpt).

- o LSP State Update, Path Computation Update message (PCUpd).

[I-D.ietf-pce-pce-initiated-lsp] :

- o PCE-initiated LSP, Path Computation LSP Initiate Message (PCInitiate).

[I-D.ietf-pce-lsp-setup-type] :

- o Path Setup Type (PST).

[I-D.ietf-pce-segment-routing] :

- o Segment Routing (SR).

- o Segment Identifier (SID).

- o Maximum SID Depth (MSD).

3.1. Tree Diagrams

A graphical representation of the complete data tree is presented in Section 5. The meaning of the symbols in these diagrams is as follows and as per [I-D.ietf-netmod-rfc6087bis]:

- o Brackets "[" and "]" enclose list keys.

- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

- o Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).

- o Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

3.2. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and corresponding YANG modules

4. Objectives

This section describes some of the design objectives for the model:

- o In case of existing implementations, it needs to map the data model defined in this document to their proprietary native data model. To facilitate such mappings, the data model should be simple.
- o The data model should be suitable for new implementations to use as is.
- o Mapping to the PCEP MIB Module should be clear.
- o The data model should allow for static configurations of peers.
- o The data model should include read-only counters in order to gather statistics for sent and received PCEP messages, received messages with errors, and messages that could not be sent due to errors.

o It should be fairly straightforward to augment the base data model for advanced PCE features.

5. The Design of PCEP Data Model

The module, "ietf-pcep", defines the basic components of a PCE speaker.

```
module: ietf-pcep
   +--rw pcep!
   |  +--rw entity
   |     +--rw addr                            inet:ip-address
   |     +--rw enabled?                        boolean
   |     +--rw role                            pcep-role
   |     +--rw description?                    string
   |     +--rw domain
   |     |  +--rw domain* [domain-type domain]
   |     |     +--rw domain-type    domain-type
   |     |     +--rw domain         domain
   |     +--rw capability
   |     |  +--rw gmpls?                    boolean {gmpls}?
   |     |  +--rw bi-dir?                   boolean
   |     |  +--rw diverse?                  boolean
   |     |  +--rw load-balance?             boolean
   |     |  +--rw synchronize?              boolean {svec}?
   |     |  +--rw objective-function?       boolean {obj-fn}?
   |     |  +--rw add-path-constraint?      boolean
   |     |  +--rw prioritization?           boolean
   |     |  +--rw multi-request?            boolean
   |     |  +--rw gco?                      boolean {gco}?
   |     |  +--rw p2mp?                     boolean {p2mp}?
   |     |  +--rw stateful {stateful}?
   |     |  |  +--rw enabled?          boolean
   |     |  |  +--rw active?           boolean
   |     |  |  +--rw pce-initiated?    boolean {pce-initiated}?
   |     |  +--rw sr {sr}?
   |     |     +--rw enabled?    boolean
   |     |     +--rw msd?        uint8
   |     +--rw pce-info
   |     |  +--rw scope
   |     |  |  +--rw intra-area-scope?            boolean
   |     |  |  +--rw intra-area-pref?             uint8
   |     |  |  +--rw inter-area-scope?            boolean
   |     |  |  +--rw inter-area-scope-default?    boolean
   |     |  |  +--rw inter-area-pref?             uint8
   |     |  |  +--rw inter-as-scope?              boolean
   |     |  |  +--rw inter-as-scope-default?      boolean
   |     |  |  +--rw inter-as-pref?               uint8
```

```
   |      |  |  +--rw inter-layer-scope?          boolean
   |      |  |  +--rw inter-layer-pref?           uint8
   |      |  +--rw neigh-domains
   |      |  |  +--rw domain* [domain-type domain]
   |      |  |     +--rw domain-type    domain-type
   |      |  |     +--rw domain         domain
   |      |  +--rw (auth-type-selection)?
   |      |     +--:(auth-key-chain)
   |      |     |  +--rw key-chain?          key-chain:key-chain-ref
   |      |     +--:(auth-key)
   |      |     |  +--rw key?                string
   |      |     |  +--rw crypto-algorithm
   |      |     |     +--rw (algorithm)?
   |      |     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
   |      |     |        |  +--rw hmac-sha1-12?             empty
   |      |     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
   |      |     |        |  +--rw aes-cmac-prf-128?         empty
   |      |     |        +--:(md5)
   |      |     |        |  +--rw md5?                      empty
   |      |     |        +--:(sha-1)
   |      |     |        |  +--rw sha-1?                    empty
   |      |     |        +--:(hmac-sha-1)
   |      |     |        |  +--rw hmac-sha-1?               empty
   |      |     |        +--:(hmac-sha-256)
   |      |     |        |  +--rw hmac-sha-256?             empty
   |      |     |        +--:(hmac-sha-384)
   |      |     |        |  +--rw hmac-sha-384?             empty
   |      |     |        +--:(hmac-sha-512)
   |      |     |        |  +--rw hmac-sha-512?             empty
   |      |     |        +--:(clear-text) {clear-text}?
   |      |     |        |  +--rw clear-text?               empty
   |      |     |        +--:(replay-protection-only) {replay-protection-only}?
   |      |     |           +--rw replay-protection-only?   empty
   |      |     +--:(auth-tls) {tls}?
   |      |        +--rw tls
   |     +--rw connect-timer?               uint32
   |     +--rw connect-max-retry?           uint32
   |     +--rw init-backoff-timer?          uint32
   |     +--rw max-backoff-timer?           uint32
   |     +--rw open-wait-timer?             uint32
   |     +--rw keep-wait-timer?             uint32
   |     +--rw keep-alive-timer?            uint32
   |     +--rw dead-timer?                  uint32
   |     +--rw allow-negotiation?           boolean
   |     +--rw max-keep-alive-timer?        uint32
   |     +--rw max-dead-timer?              uint32
   |     +--rw min-keep-alive-timer?        uint32
   |     +--rw min-dead-timer?              uint32
```

```
  |     +--rw sync-timer?                   uint32 {svec}?
  |     +--rw request-timer?                uint32
  |     +--rw max-sessions?                 uint32
  |     +--rw max-unknown-reqs?             uint32
  |     +--rw max-unknown-msgs?             uint32
  |     +--rw pcep-notification-max-rate    uint32
  |     +--rw stateful-parameter {stateful}?
  |     |  +--rw state-timeout?          uint32
  |     |  +--rw redelegation-timeout?   uint32
  |     |  +--rw rpt-non-pcep-lsp?       boolean
  |     +--rw peers
  |        +--rw peer* [addr]
  |           +--rw addr                  inet:ip-address
  |           +--rw description?          string
  |           +--rw domain
  |           |  +--rw domain* [domain-type domain]
  |           |     +--rw domain-type    domain-type
  |           |     +--rw domain         domain
  |           +--rw capability
  |           |  +--rw gmpls?                   boolean {gmpls}?
  |           |  +--rw bi-dir?                  boolean
  |           |  +--rw diverse?                 boolean
  |           |  +--rw load-balance?            boolean
  |           |  +--rw synchronize?             boolean {svec}?
  |           |  +--rw objective-function?      boolean {obj-fn}?
  |           |  +--rw add-path-constraint?     boolean
  |           |  +--rw prioritization?          boolean
  |           |  +--rw multi-request?           boolean
  |           |  +--rw gco?                     boolean {gco}?
  |           |  +--rw p2mp?                    boolean {p2mp}?
  |           |  +--rw stateful {stateful}?
  |           |  |  +--rw enabled?         boolean
  |           |  |  +--rw active?          boolean
  |           |  |  +--rw pce-initiated?   boolean {pce-initiated}?
  |           |  +--rw sr {sr}?
  |           |     +--rw enabled?   boolean
  |           |     +--rw msd?       uint8
  |           +--rw scope
  |           |  +--rw intra-area-scope?           boolean
  |           |  +--rw intra-area-pref?            uint8
  |           |  +--rw inter-area-scope?           boolean
  |           |  +--rw inter-area-scope-default?   boolean
  |           |  +--rw inter-area-pref?            uint8
  |           |  +--rw inter-as-scope?             boolean
  |           |  +--rw inter-as-scope-default?     boolean
  |           |  +--rw inter-as-pref?              uint8
  |           |  +--rw inter-layer-scope?          boolean
  |           |  +--rw inter-layer-pref?           uint8
```

```
   |            +--rw neigh-domains
   |            |  +--rw domain* [domain-type domain]
   |            |     +--rw domain-type    domain-type
   |            |     +--rw domain         domain
   |            +--rw delegation-pref?    uint8 {stateful}?
   |            +--rw (auth-type-selection)?
   |               +--:(auth-key-chain)
   |               |  +--rw key-chain?          key-chain:key-chain-ref
   |               +--:(auth-key)
   |               |  +--rw key?                string
   |               |  +--rw crypto-algorithm
   |               |     +--rw (algorithm)?
   |               |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
   |               |        |  +--rw hmac-sha1-12?             empty
   |               |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
   |               |        |  +--rw aes-cmac-prf-128?         empty
   |               |        +--:(md5)
   |               |        |  +--rw md5?                      empty
   |               |        +--:(sha-1)
   |               |        |  +--rw sha-1?                    empty
   |               |        +--:(hmac-sha-1)
   |               |        |  +--rw hmac-sha-1?               empty
   |               |        +--:(hmac-sha-256)
   |               |        |  +--rw hmac-sha-256?             empty
   |               |        +--:(hmac-sha-384)
   |               |        |  +--rw hmac-sha-384?             empty
   |               |        +--:(hmac-sha-512)
   |               |        |  +--rw hmac-sha-512?             empty
   |               |        +--:(clear-text) {clear-text}?
   |               |        |  +--rw clear-text?               empty
   |               |        +--:(replay-protection-only) {replay-protection-only}
?
   |               |           +--rw replay-protection-only?   empty
   |               +--:(auth-tls) {tls}?
   |                  +--rw tls
   +--ro pcep-state
      +--ro entity
         +--ro addr?                    inet:ip-address
         +--ro index?                   uint32
         +--ro admin-status?            pcep-admin-status
         +--ro oper-status?             pcep-admin-status
         +--ro role?                    pcep-role
         +--ro domain
         |  +--ro domain* [domain-type domain]
         |     +--ro domain-type    domain-type
         |     +--ro domain         domain
         +--ro capability
         |  +--ro gmpls?                  boolean {gmpls}?
         |  +--ro bi-dir?                 boolean
```

```
         |  +--ro diverse?                boolean
         |  +--ro load-balance?           boolean
         |  +--ro synchronize?            boolean {svec}?
         |  +--ro objective-function?     boolean {obj-fn}?
         |  +--ro add-path-constraint?    boolean
         |  +--ro prioritization?         boolean
         |  +--ro multi-request?          boolean
         |  +--ro gco?                    boolean {gco}?
         |  +--ro p2mp?                   boolean {p2mp}?
         |  +--ro stateful {stateful}?
         |  |  +--ro enabled?          boolean
         |  |  +--ro active?           boolean
         |  |  +--ro pce-initiated?    boolean {pce-initiated}?
         |  +--ro sr {sr}?
         |     +--ro enabled?    boolean
         |     +--ro msd?        uint8
         +--ro pce-info
         |  +--ro scope
         |  |  +--ro intra-area-scope?           boolean
         |  |  +--ro intra-area-pref?            uint8
         |  |  +--ro inter-area-scope?           boolean
         |  |  +--ro inter-area-scope-default?   boolean
         |  |  +--ro inter-area-pref?            uint8
         |  |  +--ro inter-as-scope?             boolean
         |  |  +--ro inter-as-scope-default?     boolean
         |  |  +--ro inter-as-pref?              uint8
         |  |  +--ro inter-layer-scope?          boolean
         |  |  +--ro inter-layer-pref?           uint8
         |  +--ro neigh-domains
         |  |  +--ro domain* [domain-type domain]
         |  |     +--ro domain-type    domain-type
         |  |     +--ro domain         domain
         |  +--ro (auth-type-selection)?
         |     +--:(auth-key-chain)
         |     |  +--ro key-chain?          key-chain:key-chain-ref
         |     +--:(auth-key)
         |     |  +--ro key?                string
         |     |  +--ro crypto-algorithm
         |     |     +--ro (algorithm)?
         |     |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
         |     |        |  +--ro hmac-sha1-12?             empty
         |     |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
         |     |        |  +--ro aes-cmac-prf-128?         empty
         |     |        +--:(md5)
         |     |        |  +--ro md5?                      empty
         |     |        +--:(sha-1)
         |     |        |  +--ro sha-1?                    empty
         |     |        +--:(hmac-sha-1)
```

```
         |     |        |  +--ro hmac-sha-1?               empty
         |     |        +--:(hmac-sha-256)
         |     |        |  +--ro hmac-sha-256?             empty
         |     |        +--:(hmac-sha-384)
         |     |        |  +--ro hmac-sha-384?             empty
         |     |        +--:(hmac-sha-512)
         |     |        |  +--ro hmac-sha-512?             empty
         |     |        +--:(clear-text) {clear-text}?
         |     |        |  +--ro clear-text?               empty
         |     |        +--:(replay-protection-only) {replay-protection-only}?
         |     |           +--ro replay-protection-only?   empty
         |     +--:(auth-tls) {tls}?
         |        +--ro tls
         +--ro connect-timer?          uint32
         +--ro connect-max-retry?      uint32
         +--ro init-backoff-timer?     uint32
         +--ro max-backoff-timer?      uint32
         +--ro open-wait-timer?        uint32
         +--ro keep-wait-timer?        uint32
         +--ro keep-alive-timer?       uint32
         +--ro dead-timer?             uint32
         +--ro allow-negotiation?      boolean
         +--ro max-keep-alive-timer?   uint32
         +--ro max-dead-timer?         uint32
         +--ro min-keep-alive-timer?   uint32
         +--ro min-dead-timer?         uint32
         +--ro sync-timer?             uint32 {svec}?
         +--ro request-timer?          uint32
         +--ro max-sessions?           uint32
         +--ro max-unknown-reqs?       uint32
         +--ro max-unknown-msgs?       uint32
         +--ro stateful-parameter {stateful}?
         |  +--ro state-timeout?          uint32
         |  +--ro redelegation-timeout?   uint32
         |  +--ro rpt-non-pcep-lsp?       boolean
         +--ro lsp-db {stateful}?
         |  +--ro association-list* [id source global-source extended-id]
         |  |  +--ro type?            assoc-type
         |  |  +--ro id               uint16
         |  |  +--ro source           inet:ip-address
         |  |  +--ro global-source    uint32
         |  |  +--ro extended-id      string
         |  |  +--ro lsp* [plsp-id pcc-id]
         |  |     +--ro plsp-id    -> /pcep-state/entity/lsp-db/lsp/plsp-id
         |  |     +--ro pcc-id     -> /pcep-state/entity/lsp-db/lsp/pcc-id
         |  +--ro lsp* [plsp-id pcc-id]
         |     +--ro plsp-id               uint32
         |     +--ro pcc-id                inet:ip-address
```

```
         |     +--ro lsp-ref
         |     |  +--ro source?               -> /te:te/lsps-state/lsp/source
         |     |  +--ro destination?          -> /te:te/lsps-state/lsp/destinati
on
         |     |  +--ro tunnel-id?            -> /te:te/lsps-state/lsp/tunnel-id
         |     |  +--ro lsp-id?               -> /te:te/lsps-state/lsp/lsp-id
         |     |  +--ro extended-tunnel-id?   -> /te:te/lsps-state/lsp/extended-
tunnel-id
         |     |  +--ro type?                 -> /te:te/lsps-state/lsp/type
         |     +--ro admin-state?          boolean
         |     +--ro operational-state?    operational-state
         |     +--ro delegated
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     |  +--ro srp-id?    uint32
         |     +--ro initiation {pce-initiated}?
         |     |  +--ro enabled?   boolean
         |     |  +--ro pce?       -> /pcep-state/entity/peers/peer/addr
         |     +--ro symbolic-path-name?   string
         |     +--ro last-error?           lsp-error
         |     +--ro pst?                  pst
         |     +--ro association-list* [id source global-source extended-id]
         |        +--ro id              -> /pcep-state/entity/lsp-db/associatio
n-list/id
         |        +--ro source          -> /pcep-state/entity/lsp-db/associatio
n-list/source
         |        +--ro global-source    -> /pcep-state/entity/lsp-db/associatio
n-list/global-source
         |        +--ro extended-id      -> /pcep-state/entity/lsp-db/associatio
n-list/extended-id
         +--ro peers
            +--ro peer* [addr]
               +--ro addr                    inet:ip-address
               +--ro role?                   pcep-role
               +--ro domain
               |  +--ro domain* [domain-type domain]
               |     +--ro domain-type    domain-type
               |     +--ro domain         domain
               +--ro capability
               |  +--ro gmpls?                 boolean {gmpls}?
               |  +--ro bi-dir?                boolean
               |  +--ro diverse?               boolean
               |  +--ro load-balance?          boolean
               |  +--ro synchronize?           boolean {svec}?
               |  +--ro objective-function?    boolean {obj-fn}?
               |  +--ro add-path-constraint?   boolean
               |  +--ro prioritization?        boolean
               |  +--ro multi-request?         boolean
               |  +--ro gco?                   boolean {gco}?
               |  +--ro p2mp?                  boolean {p2mp}?
               |  +--ro stateful {stateful}?
               |  |  +--ro enabled?         boolean
               |  |  +--ro active?          boolean
               |  |  +--ro pce-initiated?   boolean {pce-initiated}?
```

```
               |  +--ro sr {sr}?
               |     +--ro enabled?   boolean
               |     +--ro msd?       uint8
               +--ro pce-info
               |  +--ro scope
               |  |  +--ro intra-area-scope?           boolean
               |  |  +--ro intra-area-pref?            uint8
               |  |  +--ro inter-area-scope?           boolean
               |  |  +--ro inter-area-scope-default?   boolean
               |  |  +--ro inter-area-pref?            uint8
               |  |  +--ro inter-as-scope?             boolean
               |  |  +--ro inter-as-scope-default?     boolean
               |  |  +--ro inter-as-pref?              uint8
               |  |  +--ro inter-layer-scope?          boolean
               |  |  +--ro inter-layer-pref?           uint8
               |  +--ro neigh-domains
               |     +--ro domain* [domain-type domain]
               |        +--ro domain-type    domain-type
               |        +--ro domain         domain
               +--ro delegation-pref?        uint8 {stateful}?
               +--ro (auth-type-selection)?
               |  +--:(auth-key-chain)
               |  |  +--ro key-chain?             key-chain:key-chain-ref
               |  +--:(auth-key)
               |  |  +--ro key?                   string
               |  |  +--ro crypto-algorithm
               |  |     +--ro (algorithm)?
               |  |        +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
               |  |        |  +--ro hmac-sha1-12?            empty
               |  |        +--:(aes-cmac-prf-128) {aes-cmac-prf-128}?
               |  |        |  +--ro aes-cmac-prf-128?        empty
               |  |        +--:(md5)
               |  |        |  +--ro md5?                     empty
               |  |        +--:(sha-1)
               |  |        |  +--ro sha-1?                   empty
               |  |        +--:(hmac-sha-1)
               |  |        |  +--ro hmac-sha-1?              empty
               |  |        +--:(hmac-sha-256)
               |  |        |  +--ro hmac-sha-256?            empty
               |  |        +--:(hmac-sha-384)
               |  |        |  +--ro hmac-sha-384?            empty
               |  |        +--:(hmac-sha-512)
               |  |        |  +--ro hmac-sha-512?            empty
               |  |        +--:(clear-text) {clear-text}?
               |  |        |  +--ro clear-text?              empty
               |  |        +--:(replay-protection-only) {replay-protection-only}
?
               |  |           +--ro replay-protection-only?   empty
               |  +--:(auth-tls) {tls}?
```

```
               |     +--ro tls
               +--ro discontinuity-time?     yang:timestamp
               +--ro initiate-session?       boolean
               +--ro session-exists?         boolean
               +--ro num-sess-setup-ok?      yang:counter32
               +--ro num-sess-setup-fail?    yang:counter32
               +--ro session-up-time?        yang:timestamp
               +--ro session-fail-time?      yang:timestamp
               +--ro session-fail-up-time?   yang:timestamp
               +--ro pcep-stats
               |  +--ro avg-rsp-time?               uint32
               |  +--ro lwm-rsp-time?               uint32
               |  +--ro hwm-rsp-time?               uint32
               |  +--ro num-pcreq-sent?             yang:counter32
               |  +--ro num-pcreq-rcvd?             yang:counter32
               |  +--ro num-pcrep-sent?             yang:counter32
               |  +--ro num-pcrep-rcvd?             yang:counter32
               |  +--ro num-pcerr-sent?             yang:counter32
               |  +--ro num-pcerr-rcvd?             yang:counter32
               |  +--ro num-pcntf-sent?             yang:counter32
               |  +--ro num-pcntf-rcvd?             yang:counter32
               |  +--ro num-keepalive-sent?         yang:counter32
               |  +--ro num-keepalive-rcvd?         yang:counter32
               |  +--ro num-unknown-rcvd?           yang:counter32
               |  +--ro num-corrupt-rcvd?           yang:counter32
               |  +--ro num-req-sent?               yang:counter32
               |  +--ro num-req-sent-pend-rep?      yang:counter32
               |  +--ro num-req-sent-ero-rcvd?      yang:counter32
               |  +--ro num-req-sent-nopath-rcvd?   yang:counter32
               |  +--ro num-req-sent-cancel-rcvd?   yang:counter32
               |  +--ro num-req-sent-error-rcvd?    yang:counter32
               |  +--ro num-req-sent-timeout?       yang:counter32
               |  +--ro num-req-sent-cancel-sent?   yang:counter32
               |  +--ro num-req-rcvd?               yang:counter32
               |  +--ro num-req-rcvd-pend-rep?      yang:counter32
               |  +--ro num-req-rcvd-ero-sent?      yang:counter32
               |  +--ro num-req-rcvd-nopath-sent?   yang:counter32
               |  +--ro num-req-rcvd-cancel-sent?   yang:counter32
               |  +--ro num-req-rcvd-error-sent?    yang:counter32
               |  +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
               |  +--ro num-rep-rcvd-unknown?       yang:counter32
               |  +--ro num-req-rcvd-unknown?       yang:counter32
               |  +--ro svec {svec}?
               |  |  +--ro num-svec-sent?       yang:counter32
               |  |  +--ro num-svec-req-sent?   yang:counter32
               |  |  +--ro num-svec-rcvd?       yang:counter32
               |  |  +--ro num-svec-req-rcvd?   yang:counter32
               |  +--ro stateful {stateful}?
```

```
               |  |  +--ro num-pcrpt-sent?             yang:counter32
               |  |  +--ro num-pcrpt-rcvd?             yang:counter32
               |  |  +--ro num-pcupd-sent?             yang:counter32
               |  |  +--ro num-pcupd-rcvd?             yang:counter32
               |  |  +--ro num-rpt-sent?               yang:counter32
               |  |  +--ro num-rpt-rcvd?               yang:counter32
               |  |  +--ro num-rpt-rcvd-error-sent?    yang:counter32
               |  |  +--ro num-upd-sent?               yang:counter32
               |  |  +--ro num-upd-rcvd?               yang:counter32
               |  |  +--ro num-upd-rcvd-unknown?       yang:counter32
               |  |  +--ro num-upd-rcvd-undelegated?   yang:counter32
               |  |  +--ro num-upd-rcvd-error-sent?    yang:counter32
               |  |  +--ro initiation {pce-initiated}?
               |  |     +--ro num-pcinitiate-sent?            yang:counter32
               |  |     +--ro num-pcinitiate-rcvd?            yang:counter32
               |  |     +--ro num-initiate-sent?              yang:counter32
               |  |     +--ro num-initiate-rcvd?              yang:counter32
               |  |     +--ro num-initiate-rcvd-error-sent?   yang:counter32
               |  +--ro num-req-sent-closed?        yang:counter32
               |  +--ro num-req-rcvd-closed?        yang:counter32
               +--ro sessions
                  +--ro session* [initiator]
                     +--ro initiator               pcep-initiator
                     +--ro state-last-change?      yang:timestamp
                     +--ro state?                  pcep-sess-state
                     +--ro session-creation?       yang:timestamp
                     +--ro connect-retry?          yang:counter32
                     +--ro local-id?               uint32
                     +--ro remote-id?              uint32
                     +--ro keepalive-timer?        uint32
                     +--ro peer-keepalive-timer?   uint32
                     +--ro dead-timer?             uint32
                     +--ro peer-dead-timer?        uint32
                     +--ro ka-hold-time-rem?       uint32
                     +--ro overloaded?             boolean
                     +--ro overload-time?          uint32
                     +--ro peer-overloaded?        boolean
                     +--ro peer-overload-time?     uint32
                     +--ro lspdb-sync?             sync-state {stateful}?
                     +--ro discontinuity-time?     yang:timestamp
                     +--ro pcep-stats
                        +--ro avg-rsp-time?              uint32
                        +--ro lwm-rsp-time?              uint32
                        +--ro hwm-rsp-time?              uint32
                        +--ro num-pcreq-sent?            yang:counter32
                        +--ro num-pcreq-rcvd?            yang:counter32
                        +--ro num-pcrep-sent?            yang:counter32
                        +--ro num-pcrep-rcvd?            yang:counter32
```

```
                        +--ro num-pcerr-sent?             yang:counter32
                        +--ro num-pcerr-rcvd?             yang:counter32
                        +--ro num-pcntf-sent?             yang:counter32
                        +--ro num-pcntf-rcvd?             yang:counter32
                        +--ro num-keepalive-sent?         yang:counter32
                        +--ro num-keepalive-rcvd?         yang:counter32
                        +--ro num-unknown-rcvd?           yang:counter32
                        +--ro num-corrupt-rcvd?           yang:counter32
                        +--ro num-req-sent?               yang:counter32
                        +--ro num-req-sent-pend-rep?      yang:counter32
                        +--ro num-req-sent-ero-rcvd?      yang:counter32
                        +--ro num-req-sent-nopath-rcvd?   yang:counter32
                        +--ro num-req-sent-cancel-rcvd?   yang:counter32
                        +--ro num-req-sent-error-rcvd?    yang:counter32
                        +--ro num-req-sent-timeout?       yang:counter32
                        +--ro num-req-sent-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd?               yang:counter32
                        +--ro num-req-rcvd-pend-rep?      yang:counter32
                        +--ro num-req-rcvd-ero-sent?      yang:counter32
                        +--ro num-req-rcvd-nopath-sent?   yang:counter32
                        +--ro num-req-rcvd-cancel-sent?   yang:counter32
                        +--ro num-req-rcvd-error-sent?    yang:counter32
                        +--ro num-req-rcvd-cancel-rcvd?   yang:counter32
                        +--ro num-rep-rcvd-unknown?       yang:counter32
                        +--ro num-req-rcvd-unknown?       yang:counter32
                        +--ro svec {svec}?
                        |  +--ro num-svec-sent?       yang:counter32
                        |  +--ro num-svec-req-sent?   yang:counter32
                        |  +--ro num-svec-rcvd?       yang:counter32
                        |  +--ro num-svec-req-rcvd?   yang:counter32
                        +--ro stateful {stateful}?
                           +--ro num-pcrpt-sent?             yang:counter32
                           +--ro num-pcrpt-rcvd?             yang:counter32
                           +--ro num-pcupd-sent?             yang:counter32
                           +--ro num-pcupd-rcvd?             yang:counter32
                           +--ro num-rpt-sent?               yang:counter32
                           +--ro num-rpt-rcvd?               yang:counter32
                           +--ro num-rpt-rcvd-error-sent?    yang:counter32
                           +--ro num-upd-sent?               yang:counter32
                           +--ro num-upd-rcvd?               yang:counter32
                           +--ro num-upd-rcvd-unknown?       yang:counter32
                           +--ro num-upd-rcvd-undelegated?   yang:counter32
                           +--ro num-upd-rcvd-error-sent?    yang:counter32
                           +--ro initiation {pce-initiated}?
                              +--ro num-pcinitiate-sent?           yang:counter
32
                              +--ro num-pcinitiate-rcvd?           yang:counter
32
                              +--ro num-initiate-sent?             yang:counter
32
                              +--ro num-initiate-rcvd?             yang:counter
32
```

```
                              +--ro num-initiate-rcvd-error-sent?   yang:counter
32
notifications:
   +---n pcep-session-up
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-down
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   pcep-initiator
   |  +--ro state-last-change?   yang:timestamp
   |  +--ro state?               pcep-sess-state
   +---n pcep-session-local-overload
   |  +--ro peer-addr?           -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?   -> /pcep-state/entity/peers/peer/sessions/sessi
on/initiator
   |  +--ro overloaded?          boolean
   |  +--ro overload-time?       uint32
   +---n pcep-session-local-overload-clear
   |  +--ro peer-addr?    -> /pcep-state/entity/peers/peer/addr
   |  +--ro overloaded?   boolean
   +---n pcep-session-peer-overload
   |  +--ro peer-addr?            -> /pcep-state/entity/peers/peer/addr
   |  +--ro session-initiator?    -> /pcep-state/entity/peers/peer/sessions/sess
ion/initiator
   |  +--ro peer-overloaded?      boolean
   |  +--ro peer-overload-time?   uint32
   +---n pcep-session-peer-overload-clear
      +--ro peer-addr?         -> /pcep-state/entity/peers/peer/addr
      +--ro peer-overloaded?   boolean
```

5.1. The Entity

The PCEP yang module may contain status information for the local PCEP entity.

The entity has an IP address (using ietf-inet-types [RFC6991]) and a "role" leaf (the local entity PCEP role) as mandatory.

Note that, the PCEP MIB module [RFC7420] uses an entity list and a system generated entity index as a primary index to the read only entity table. If the device implements the PCEP MIB, the "index" leaf MUST contain the value of the corresponding pcePcepEntityIndex and only one entity is assumed.

5.2. The Peer Lists

The peer list contains peer(s) that the local PCEP entity knows about. A PCEP speaker is identified by its IP address. If there is a PCEP speaker in the network that uses multiple IP addresses then it looks like multiple distinct peers to the other PCEP speakers in the network.

Since PCEP sessions can be ephemeral, the peer list tracks a peer even when no PCEP session currently exists to that peer. The statistics contained are an aggregate of the statistics for all successive sessions to that peer.

To limit the quantity of information that is stored, an implementation MAY choose to discard this information if and only if no PCEP session exists to the corresponding peer.

The data model for PCEP peer presented in this document uses a flat list of peers. Each peer in the list is identified by its IP address (addr-type, addr).

There is one list for static peer configuration ("/pcep/entity/ peers"), and a separate list for the operational state of all peers (i.e. static as well as discovered)("/pcep-state/entity/peers"). The former is used to enable remote PCE configuration at PCC (or PCE) while the latter has the operational state of these peers as well as the remote PCE peer which were discovered and PCC peers that have initiated session.

5.3. The Session Lists

The session list contains PCEP session that the PCEP entity (PCE or PCC) is currently participating in. The statistics in session are semantically different from those in peer since the former applies to the current session only, whereas the latter is the aggregate for all sessions that have existed to that peer.

Although [RFC5440] forbids more than one active PCEP session between a given pair of PCEP entities at any given time, there is a window during session establishment where two sessions may exist for a given pair, one representing a session initiated by the local PCEP entity and the other representing a session initiated by the peer. If either of these sessions reaches active state first, then the other is discarded.

The data model for PCEP session presented in this document uses a flat list of sessions. Each session in the list is identified by its initiator. This index allows two sessions to exist transiently for a given peer, as discussed above.

There is only one list for the operational state of all sessions ("/pcep-state/entity/peers/peer/sessions/session").

5.4. Notifications

This YANG model defines a list of notifications to inform client of important events detected during the protocol operation. The notifications defined cover the PCEP MIB notifications.

6. Advanced PCE Features

This document contains a specification of the base PCEP YANG module, "ietf-pcep" which provides the basic PCEP [RFC5440] data model.

This document further handles advanced PCE features like -

- Capability and Scope
- Domain information (local/neighbour)
- Path-Key
- OF
- GCO
- P2MP
- GMPLS
- Inter-Layer
- Stateful PCE
- Segement Routing
- Authentication including PCEPS (TLS)

[Editor's Note - Some of them would be added in a future revision.]

6.1. Stateful PCE's LSP-DB

In the operational state of PCEP which supports stateful PCE mode, the list of LSP state are maintained in LSP-DB. The key is the PLSP-ID and the PCC IP address.

The PCEP data model contains the operational state of LSPs (/pcep-state/entity/lsp-db/lsp/) with PCEP specific attributes. The generic TE attributes of the LSP are defined in [I-D.ietf-teas-yang-te]. A reference to LSP state in TE model is maintained.

7. Open Issues and Next Step

This section is added so that open issues can be tracked. This section would be removed when the document is ready for publication.

7.1. The PCE-Initiated LSP

The TE Model at [I-D.ietf-teas-yang-te] should support creationg of tunnels at the controller (PCE) and marking them as PCE-Initiated. The LSP-DB in the PCEP Yang (/pcep-state/entity/lsp-db/lsp/initiation) also marks the LSPs which are PCE-initiated.

7.2. PCEP over TLS (PCEPS)

A future version of this document would add TLS related configurations.

8. PCEP YANG Module

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-pcep@2016-07-07.yang"
module ietf-pcep {
    namespace "urn:ietf:params:xml:ns:yang:ietf-pcep";
    prefix pcep;

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

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

    import ietf-te {
        prefix "te";
    }

    import ietf-key-chain {
       prefix "key-chain";
```

```
    }


    organization
        "IETF PCE (Path Computation Element) Working Group";

    contact
        "WG Web:   <http://tools.ietf.org/wg/pce/>
         WG List:  <mailto:pce@ietf.org>
         WG Chair: JP Vasseur
                   <mailto:jpv@cisco.com>
         WG Chair: Julien Meuric
                   <mailto:julien.meuric@orange.com>
         WG Chair: Jonathan Hardwick
                   <mailto:Jonathan.Hardwick@metaswitch.com>
         Editor:   Dhruv Dhody
                   <mailto:dhruv.ietf@gmail.com>";

    description
        "The YANG module defines a generic configuration and
         operational model for PCEP common across all of the
         vendor implementations.";


    revision 2016-07-07 {
        description "Initial revision.";
        reference
            "RFC XXXX:  A YANG Data Model for Path Computation
                        Element Communications Protocol
                        (PCEP)";
    }

    /*
     * Identities
     */

    identity pcep {
        description "Identity for the PCEP protocol.";
    }

    /*
     * Typedefs
     */
    typedef pcep-role {
        type enumeration {
            enum unknown {
                value "0";
                description
```

```
                "An unknown role";
            }
            enum pcc {
                value "1";
                description
                "The role of a Path Computation Client";
            }
            enum pce {
                value "2";
                description
                "The role of Path Computation Element";
            }
            enum pcc-and-pce {
                value "3";
                description
                "The role of both Path Computation Client and
                 Path Computation Element";
            }
        }

        description
            "The role of a PCEP speaker.
             Takes one of the following values
             - unknown(0): the role is not known.
             - pcc(1): the role is of a Path Computation
               Client (PCC).
             - pce(2): the role is of a Path Computation
               Server (PCE).
             - pccAndPce(3): the role is of both a PCC and
               a PCE.";

    }

    typedef pcep-admin-status {
        type enumeration {
                enum admin-status-up {
                value "1";
                description
                "Admin Status is Up";
            }
            enum admin-status-down {
                value "2";
                description
                "Admin Status is Down";
            }
        }

        description
```

```
        "The Admin Status of the PCEP entity.
         Takes one of the following values
             - admin-status-up(1): Admin Status is Up.
             - admin-status-down(2): Admin Status is Down";
    }

    typedef pcep-oper-status {
        type enumeration {
            enum oper-status-up {
                value "1";
                description
                "The PCEP entity is active";
            }
            enum oper-status-down {
                value "2";
                description
                "The PCEP entity is inactive";
            }
            enum oper-status-going-up {
                value "3";
                description
                "The PCEP entity is activating";
            }
            enum oper-status-going-down {
                value "4";
                description
                "The PCEP entity is deactivating";
            }
            enum oper-status-failed {
                value "5";
                description
                "The PCEP entity has failed and will recover
                 when possible.";
            }
            enum oper-status-failed-perm {
                value "6";
                description
                "The PCEP entity has failed and will not recover
                 without operator intervention";
            }
        }
        description
        "The operational status of the PCEP entity.
         Takes one of the following values
             - oper-status-up(1): Active
             - oper-status-down(2): Inactive
             - oper-status-going-up(3): Activating
             - oper-status-going-down(4): Deactivating
```

```
             - oper-status-failed(5): Failed
             - oper-status-failed-perm(6): Failed Permanantly";
    }

    typedef pcep-initiator {
        type enumeration {
            enum local {
                value "1";
                description
                "The local PCEP entity initiated the session";
            }

            enum remote {
                value "2";
                description
                "The remote PCEP peer initiated the session";
            }
        }
        description
        "The initiator of the session, that is, whether the TCP
         connection was initiated by the local PCEP entity or
         the remote peer.
         Takes one of the following values
             - local(1): Initiated locally
             - remote(2): Initiated remotely";
    }

    typedef pcep-sess-state {
        type enumeration {
            enum tcp-pending {
                value "1";
                description
                    "The tcp-pending state of PCEP session.";
            }

            enum open-wait {
                value "2";
                description
                    "The open-wait state of PCEP session.";
            }

            enum keep-wait {
                value "3";
                description
                    "The keep-wait state of PCEP session.";
            }

            enum session-up {
```

```
                value "4";
                description
                    "The session-up state of PCEP session.";
            }
        }
        description
            "The current state of the session.
             The set of possible states excludes the idle state
             since entries do not exist in the idle state.
             Takes one of the following values
                - tcp-pending(1): PCEP TCP Pending state
                - open-wait(2): PCEP Open Wait state
                - keep-wait(3): PCEP Keep Wait state
                - session-up(4): PCEP Session Up state";
    }

    typedef domain-type {
        type enumeration {
            enum ospf-area {
                value "1";
                description
                    "The OSPF area.";
            }
            enum isis-area {
                value "2";
                description
                    "The IS-IS area.";
            }
            enum as {
                value "3";
                description
                    "The Autonomous System (AS).";
            }
        }
        description
            "The PCE Domain Type";
    }

    typedef domain-ospf-area {
        type union {
            type uint32;
            type yang:dotted-quad;
       }
       description
            "OSPF Area ID.";
    }

    typedef domain-isis-area {
```

```
        type string {
            pattern '[0-9A-Fa-f]{2}\.([0-9A-Fa-f]{4}\.){0,3}';
        }
        description
            "IS-IS Area ID.";
    }

    typedef domain-as {
        type uint32;
        description
            "Autonomous System number.";

    }

    typedef domain {
        type union {
            type domain-ospf-area;
            type domain-isis-area;
            type domain-as;
        }
        description
            "The Domain Information";
    }

    typedef operational-state {
        type enumeration {
            enum down {
                value "0";
                description
                    "not active.";
            }
            enum up {
                value "1";
                description
                    "signalled.";
            }
            enum active {
                value "2";
                description
                    "up and carrying traffic.";
            }
            enum going-down {
                value "3";
                description
                    "LSP is being torn down, resources are
                     being released.";
            }
            enum going-up {
```

```
                value "4";
                description
                    "LSP is being signalled.";
            }
        }
        description
            "The operational status of the LSP";
    }

    typedef lsp-error {
        type enumeration {
            enum no-error {
                value "0";
                description
                    "No error, LSP is fine.";
            }
            enum unknown {
                value "1";
                description
                    "Unknown reason.";
            }
            enum limit {
                value "2";
                description
                    "Limit reached for PCE-controlled LSPs.";
            }
            enum pending {
                value "3";
                description
                    "Too many pending LSP update requests.";
            }
            enum unacceptable {
                value "4";
                description
                    "Unacceptable parameters.";
            }
            enum internal {
                value "5";
                description
                    "Internal error.";
            }
            enum admin {
                value "6";
                description
                    "LSP administratively brought down.";
            }
            enum preempted {
                value "7";
```

```
                description
                    "LSP preempted.";
            }
            enum rsvp {
                value "8";
                description
                    "RSVP signaling error.";
            }
        }
        description
            "The LSP Error Codes.";
    }

    typedef sync-state {
        type enumeration {
            enum pending {
                value "0";
                description
                    "The state synchronization
                     has not started.";
            }
            enum ongoing {
                value "1";
                description
                    "The state synchronization
                     is ongoing.";
            }
            enum finished {
                value "2";
                description
                    "The state synchronization
                     is finished.";
            }
        }
        description
            "The LSP-DB state synchronization operational status.";
    }

        typedef pst{
                type enumeration{
                        enum rsvp-te{
                                value "0";
                                description
                                        "RSVP-TE signaling protocol";
                        }
                        enum sr{
                                value "1";
                                description
```

```
                                        "Segment Routing Traffic Engineering";
                        }
                }
                description
                        "The Path Setup Type";
        }

    typedef assoc-type{
        type enumeration{
            enum protection{
                value "1";
                description
                    "Path Protection Association Type";
            }
        }
        description
            "The PCEP Association Type";
    }

    /*
     * Features
     */

    feature svec {
        description
            "Support synchronized path computation.";
    }

    feature gmpls {
        description
            "Support GMPLS.";
    }

    feature obj-fn {
        description
            "Support OF as per RFC 5541.";
    }

    feature gco {
        description
            "Support GCO as per RFC 5557.";
    }

    feature pathkey {
        description
            "Support pathkey as per RFC 5520.";
    }
```

```
    feature p2mp {
        description
            "Support P2MP as per RFC 6006.";
    }

    feature stateful {
        description
            "Support stateful PCE.";
    }

    feature pce-initiated {
        description
            "Support PCE-Initiated LSP.";
    }

    feature tls {
        description
            "Support PCEP over TLS.";
    }

    feature sr {
        description
            "Support Segement Routing for PCE.";
    }

    /*
     * Groupings
     */



    grouping pcep-entity-info{
        description
            "This grouping defines the attributes for PCEP entity.";
        leaf connect-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to establish a TCP connection with a peer.  If a
                 TCP connection is not established within this time
                 then PCEP aborts the session setup attempt.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
```

```
        }

        leaf connect-max-retry {
            type uint32;
            default 5;
            description
                "The maximum number of times the system tries to
                 establish a TCP connection to a peer before the
                 session with the peer transitions to the idle
                 state.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf init-backoff-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
               "The initial back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed
                session setup attempt, until a maximum back-off
                time is reached.  The maximum back-off time is
                max-backoff-timer.";
        }

        leaf max-backoff-timer {
            type uint32;
            units "seconds";
            description
               "The maximum back-off time in seconds for retrying
                a failed session setup attempt to a peer.
                The back-off time increases for each failed session
                setup attempt, until this maximum value is reached.
                Session setup attempts then repeat periodically
                without any further increase in back-off time.";
        }

        leaf open-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
```

```
               "The time in seconds that the PCEP entity will wait
                to receive an Open message from a peer after the
                TCP connection has come up.
                If no Open message is received within this time then
                PCEP terminates the TCP connection and deletes the
                associated sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-wait-timer {
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            default 60;
            description
                "The time in seconds that the PCEP entity will wait
                 to receive a Keepalive or PCErr message from a peer
                 during session initialization after receiving an
                 Open message.  If no Keepalive or PCErr message is
                 received within this time then PCEP terminates the
                 TCP connection and deletes the associated
                 sessions.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf keep-alive-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            default 30;
            description
                "The keep alive transmission timer that this PCEP
                 entity will propose in the initial OPEN message of
                 each session it is involved in.  This is the
                 maximum time between two consecutive messages sent
                 to a peer. Zero means that the PCEP entity prefers
                 not to send Keepalives at all.
                 Note that the actual Keepalive transmission
                 intervals, in either direction of an active PCEP
                 session, are determined by negotiation between the
                 peers as specified by RFC 5440, and so may differ
                 from this configured value.";
```

```
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf dead-timer {
            type uint32 {
                range "0..255";
            }
            units "seconds";
            must ". >= ../keep-alive-timer" {
                error-message "The dead timer must be "
                       + "larger than the keep alive timer";
                description
                    "This value MUST be greater than
                     keep-alive-timer.";

            }
            default 120;
            description
                "The dead timer that this PCEP entity will propose
                 in the initial OPEN message of each session it is
                 involved in. This is the time after which a peer
                 should declare a session down if it does not
                 receive any PCEP messages. Zero suggests that the
                 peer does not run a dead timer at all." ;
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf allow-negotiation{
            type boolean;
            description
                "Whether the PCEP entity will permit negotiation of
                 session parameters.";
        }

        leaf max-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will
                 accept from a peer for the interval between
                 Keepalive transmissions. Zero means that the PCEP
```

```
                 entity will allow no Keepalive transmission at
                 all." ;
        }

        leaf max-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the maximum value that this PCEP entity will accept
                 from a peer for the Dead timer.  Zero means that
                 the PCEP entity will allow not running a Dead
                 timer.";
        }

        leaf min-keep-alive-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                "In PCEP session parameter negotiation in seconds,
                 the minimum value that this PCEP entity will
                 accept for the interval between Keepalive
                 transmissions. Zero means that the PCEP entity
                 insists on no Keepalive transmission at all.";
        }

        leaf min-dead-timer{
            type uint32 {
                range "0..255";
            }
            units "seconds";
            description
                 "In PCEP session parameter negotiation in
                  seconds, the minimum value that this PCEP entity
                  will accept for the Dead timer.  Zero means that
                  the PCEP entity insists on not running a Dead
                  timer.";
        }

        leaf sync-timer{
            if-feature svec;
            type uint32 {
                range "0..65535";
            }
```

```
            units "seconds";
            default 60;
            description
                "The value of SyncTimer in seconds is used in the
                 case of synchronized path computation request
                 using the SVEC object. Consider the case where a
                 PCReq message is received by a PCE that contains
                 the SVEC object referring to M synchronized path
                 computation requests.  If after the expiration of
                 the SyncTimer all the M path computation requests
                 have not been, received a protocol error is
                 triggered and the PCE MUST cancel the whole set
                 of path computation  requests.
                 The aim of the SyncTimer is to avoid the storage
                 of unused synchronized requests should one of
                 them get lost for some reasons (for example, a
                 misbehaving PCC).
                 Zero means that the PCEP entity does not use the
                 SyncTimer.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }


        leaf request-timer{
            type uint32 {
                range "1..65535";
            }
            units "seconds";
            description
                "The maximum time that the PCEP entity will wait
                 for a response to a PCReq message.";
        }

        leaf max-sessions{
            type uint32;
            description
               "Maximum number of sessions involving this PCEP
                entity that can exist at any time.";
        }

        leaf max-unknown-reqs{
            type uint32;
            default 5;
            description
              "The maximum number of unrecognized requests and
               replies that any session on this PCEP entity is
```

```
               willing to accept per minute before terminating
               the session.
               A PCRep message contains an unrecognized reply
               if it contains an RP object whose request ID
               does not correspond to any in-progress request
               sent by this PCEP entity.
               A PCReq message contains an unrecognized request
               if it contains an RP object whose request ID is
               zero.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

        leaf max-unknown-msgs{
            type uint32;
            default 5;
            description
             "The maximum number of unknown messages that any
              session on this PCEP entity is willing to accept
              per minute before terminating the session.";
            reference
                "RFC 5440: Path Computation Element (PCE)
                           Communication Protocol (PCEP)";
        }

    }//pcep-entity-info

    grouping  pce-scope{
        description
            "This grouping defines PCE path computation scope
             information which maybe relevant to PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf intra-area-scope{
            type boolean;
            default true;
            description
                "PCE can compute intra-area paths.";
        }
        leaf intra-area-pref{
```

```
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for intra-area TE LSP
              computation.";
        }
        leaf inter-area-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-area paths.";
        }
        leaf inter-area-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-area
                 path computation.";
        }
        leaf inter-area-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-area TE LSP
              computation.";
        }
        leaf inter-as-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-AS paths.";
        }
        leaf inter-as-scope-default{
            type boolean;
            default false;
            description
                "PCE can act as a default PCE for inter-AS
                 path computation.";
        }
        leaf inter-as-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-AS TE LSP
              computation.";
```

```
        }
        leaf inter-layer-scope{
            type boolean;
            default false;
            description
                "PCE can compute inter-layer paths.";
        }
        leaf inter-layer-pref{
            type uint8{
                range "0..7";
            }
            description
              "The PCE’s preference for inter-layer TE LSP
              computation.";
        }
    }//pce-scope

    grouping  domain{
        description
            "This grouping specifies a Domain where the
             PCEP speaker has topology visibility.";
        leaf domain-type{
            type domain-type;
            description
              "The domain type.";
        }
        leaf domain{
            type domain;
            description
              "The domain Information.";
        }
    }//domain

    grouping capability{
        description
            "This grouping specifies a capability
             information of local PCEP entity. This maybe
             relevant to PCE selection as well. This
             information corresponds to PCE auto-discovery
             information.";
        reference
            "RFC 5088: OSPF Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery
             RFC 5089: IS-IS Protocol Extensions for Path
                       Computation Element (PCE)
                       Discovery";
        leaf gmpls{
```

```
            if-feature gmpls;
            type boolean;
            description
              "Path computation with GMPLS link
               constraints.";
        }
        leaf bi-dir{
            type boolean;
            description
              "Bidirectional path computation.";
        }
        leaf diverse{
            type boolean;
            description
              "Diverse path computation.";
        }
        leaf load-balance{
            type boolean;
            description
              "Load-balanced path computation.";
        }
        leaf synchronize{
            if-feature svec;
            type boolean;
            description
              "Synchronized paths computation.";
        }
        leaf objective-function{
            if-feature obj-fn;
            type boolean;
            description
              "Support for multiple objective functions.";
        }
        leaf add-path-constraint{
            type boolean;
            description
              "Support for additive path constraints (max
               hop count, etc.).";
        }
        leaf prioritization{
            type boolean;
            description
              "Support for request prioritization.";
        }
        leaf multi-request{
            type boolean;
            description
              "Support for multiple requests per message.";
```

```
        }
        leaf gco{
            if-feature gco;
            type boolean;
            description
              "Support for Global Concurrent Optimization
               (GCO).";
        }
        leaf p2mp{
            if-feature p2mp;
            type boolean;
            description
              "Support for P2MP path computation.";
        }

        container stateful{
            if-feature stateful;
            description
                "If stateful PCE feature is present";
            leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
            }
            leaf active{
                type boolean;
                description
                  "Support for active stateful PCE.";
            }
            leaf pce-initiated{
                if-feature pce-initiated;
                type boolean;
                description
                  "Support for PCE-initiated LSP.";
            }
        }
                container sr{
                        if-feature sr;
                        description
                                "If segment routing is supported";
                        leaf enabled{
                type boolean;
                description
                    "Enabled or Disabled";
                        }
                        leaf msd{ /*should be in MPLS yang model (?)*/
                                type uint8;
                must "((../../role == 'pcc')" +
```

```
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC";
                    description
                        "When PCEP entity is PCC for
                        MSD to be applicable";
                }
                                description
                                        "Maximum SID Depth";
                        }
                }
    }//capability

    grouping  info{
        description
            "This grouping specifies all information which
             maybe relevant to both PCC and PCE.
             This information corresponds to PCE auto-discovery
             information.";
        container domain{
            description
                "The local domain for the PCEP entity";
            list domain{
                key "domain-type domain";
                description
                    "The local domain.";
                uses domain{
                    description
                        "The local domain for the PCEP entity.";
                }
            }
        }
        container capability{
            description
                "The PCEP entity capability";
            uses capability{
                description
                    "The PCEP entity supported
                    capabilities.";
            }
        }
    }//info

    grouping  pce-info{
        description
            "This grouping specifies all PCE information
```

```
             which maybe relevant to the PCE selection.
             This information corresponds to PCE auto-discovery
             information.";
        container scope{
            description
                "The path computation scope";
            uses pce-scope;
        }

        container neigh-domains{
            description
                "The list of neighbour PCE-Domain
                 toward which a PCE can compute
                 paths";
            list domain{
                key "domain-type domain";

                description
                    "The neighbour domain.";
                uses domain{
                    description
                        "The PCE neighbour domain.";
                }
            }
        }
    }//pce-info

    grouping  pcep-stats{
        description
            "This grouping defines statistics for PCEP. It is used
             for both peer and current session.";
        leaf avg-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and avg-rsp-time = 0))" {
                error-message
                    "Invalid average response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
              "The average response time.
               If an average response time has not been
               calculated then this leaf has the value zero.";
```

```
        }

        leaf lwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and lwm-rsp-time = 0))" {
                error-message
                    "Invalid smallest (low-water mark)
                     response time";
                description
                    "If role is pcc then this leaf is meaningless
                     and is set to zero.";
            }
            description
             "The smallest (low-water mark) response time seen.
              If no responses have been received then this
              leaf has the value zero.";
        }

        leaf hwm-rsp-time{
            type uint32;
            units "milliseconds";
            must "(/pcep-state/entity/peers/peer/role != 'pcc'" +
                 " or " +
                 "(/pcep-state/entity/peers/peer/role = 'pcc'" +
                 " and hwm-rsp-time = 0))" {
                error-message
                    "Invalid greatest (high-water mark)
                     response time seen";
                description
                    "If role is pcc then this field is
                     meaningless and is set to zero.";
            }
            description
             "The greatest (high-water mark) response time seen.
              If no responses have been received then this object
              has the value zero.";
        }

        leaf num-pcreq-sent{
            type yang:counter32;
            description
             "The number of PCReq messages sent.";
        }
```

```
        leaf num-pcreq-rcvd{
            type yang:counter32;
            description
              "The number of PCReq messages received.";
        }

        leaf num-pcrep-sent{
            type yang:counter32;
            description
              "The number of PCRep messages sent.";
        }

        leaf num-pcrep-rcvd{
            type yang:counter32;
            description
              "The number of PCRep messages received.";
        }

        leaf num-pcerr-sent{
            type yang:counter32;
            description
              "The number of PCErr messages sent.";
        }

        leaf num-pcerr-rcvd{
            type yang:counter32;
            description
              "The number of PCErr messages received.";
        }

        leaf num-pcntf-sent{
            type yang:counter32;
            description
              "The number of PCNtf messages sent.";
        }

        leaf num-pcntf-rcvd{
            type yang:counter32;
            description
              "The number of PCNtf messages received.";
        }

        leaf num-keepalive-sent{
            type yang:counter32;
            description
              "The number of Keepalive messages sent.";
        }
```

```
        leaf num-keepalive-rcvd{
            type yang:counter32;
            description
              "The number of Keepalive messages received.";
        }

        leaf num-unknown-rcvd{
            type yang:counter32;
            description
              "The number of unknown messages received.";
        }

        leaf num-corrupt-rcvd{
            type yang:counter32;
            description
              "The number of corrupted PCEP message received.";
        }

        leaf num-req-sent{
            type yang:counter32;
            description
              "The number of requests sent.  A request corresponds
               1:1 with an RP object in a PCReq message. This might
               be greater than num-pcreq-sent because multiple
               requests can be batched into a single PCReq
               message.";
        }

        leaf num-req-sent-pend-rep{
            type yang:counter32;
            description
              "The number of requests that have been sent for
               which a response is still pending.";
        }

        leaf num-req-sent-ero-rcvd{
            type yang:counter32;
            description
               "The number of requests that have been sent for
                which a response with an ERO object was received.
                Such responses indicate that a path was
                successfully computed by the peer.";
        }

        leaf num-req-sent-nopath-rcvd{
            type yang:counter32;
            description
              "The number of requests that have been sent for
```

```
                 which a response with a NO-PATH object was
                 received. Such responses indicate that the peer
                 could not  find a path to satisfy the
                 request.";
          }

          leaf num-req-sent-cancel-rcvd{
              type yang:counter32;
              description
                "The number of requests that were cancelled with
                 a PCNtf message.
                 This might be different than num-pcntf-rcvd because
                 not all PCNtf messages are used to cancel requests,
                 and a single PCNtf message can cancel multiple
                 requests.";
          }

          leaf num-req-sent-error-rcvd{
              type yang:counter32;
              description
               "The number of requests that were rejected with a
                PCErr message.
                This might be different than num-pcerr-rcvd because
                not all PCErr messages are used to reject requests,
                and a single PCErr message can reject multiple
                requests.";
          }

          leaf num-req-sent-timeout{
              type yang:counter32;
              description
               "The number of requests that have been sent to a peer
               and have been abandoned because the peer has taken too
               long to respond to them.";
          }

          leaf num-req-sent-cancel-sent{
              type yang:counter32;
              description
               "The number of requests that were sent to the peer and
                explicitly cancelled by the local PCEP entity sending
                a PCNtf.";
          }

          leaf num-req-rcvd{
              type yang:counter32;
              description
               "The number of requests received.  A request
```

```
                corresponds 1:1 with an RP object in a PCReq
                message.
                This might be greater than num-pcreq-rcvd because
                multiple requests can be batched into a single
                PCReq message.";
        }

        leaf num-req-rcvd-pend-rep{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response is still pending.";
        }

        leaf num-req-rcvd-ero-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with an ERO object was sent.  Such
              responses indicate that a path was successfully
              computed by the local PCEP entity.";
        }

        leaf num-req-rcvd-nopath-sent{
            type yang:counter32;
            description
             "The number of requests that have been received for
              which a response with a NO-PATH object was sent. Such
              responses indicate that the local PCEP entity could
              not find a path to satisfy the request.";
        }

        leaf num-req-rcvd-cancel-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCNtf message.
              This might be different than num-pcntf-sent because
              not all PCNtf messages are used to cancel requests,
              and a single PCNtf message can cancel multiple
              requests.";
        }

        leaf num-req-rcvd-error-sent{
            type yang:counter32;
            description
             "The number of requests received that were cancelled
              by the local PCEP entity sending a PCErr message.
```

```
              This might be different than num-pcerr-sent because
              not all PCErr messages are used to cancel requests,
              and a single PCErr message can cancel multiple
              requests.";
        }

        leaf  num-req-rcvd-cancel-rcvd{
            type yang:counter32;
            description
             "The number of requests that were received from the
              peer and explicitly cancelled by the peer sending
              a PCNtf.";
        }

        leaf  num-rep-rcvd-unknown{
            type yang:counter32;
            description
              "The number of responses to unknown requests
               received. A response to an unknown request is a
               response whose RP object does not contain the
               request ID of any request that is currently
               outstanding on the session.";
        }

        leaf  num-req-rcvd-unknown{
            type yang:counter32;
            description
              "The number of unknown requests that have been
               received. An unknown request is a request
               whose RP object contains a request ID of
               zero.";
        }

        container svec{
            if-feature svec;
            description
                "If synchronized path computation is supported";
            leaf num-svec-sent{
                type yang:counter32;
                description
                  "The number of SVEC objects sent in PCReq messages.
                   An SVEC object represents a set of synchronized
                   requests.";
            }

            leaf num-svec-req-sent{
                type yang:counter32;
                description
```

```
                  "The number of requests sent that appeared in one
                   or more SVEC objects.";
            }

            leaf num-svec-rcvd{
                type yang:counter32;
                description
                 "The number of SVEC objects received in PCReq
                  messages. An SVEC object represents a set of
                  synchronized requests.";
            }

            leaf num-svec-req-rcvd{
                type yang:counter32;
                description
                  "The number of requests received that appeared
                   in one or more SVEC objects.";
            }
        }
        container stateful{
            if-feature stateful;
            description
                "Stateful PCE related statistics";
            leaf num-pcrpt-sent{
                type yang:counter32;
                description
                 "The number of PCRpt messages sent.";
            }

            leaf num-pcrpt-rcvd{
                type yang:counter32;
                description
                  "The number of PCRpt messages received.";
            }

            leaf num-pcupd-sent{
                type yang:counter32;
                description
                 "The number of PCUpd messages sent.";
            }

            leaf num-pcupd-rcvd{
                type yang:counter32;
                description
                  "The number of PCUpd messages received.";
            }

            leaf num-rpt-sent{
```

```
                type yang:counter32;
                description
                  "The number of LSP Reports sent.  A LSP report
                   corresponds 1:1 with an LSP object in a PCRpt
                   message. This might be greater than
                   num-pcrpt-sent because multiple reports can
                   be batched into a single PCRpt message.";
            }

            leaf num-rpt-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Reports received.  A LSP report
                  corresponds 1:1 with an LSP object in a PCRpt
                  message.
                  This might be greater than num-pcrpt-rcvd because
                  multiple reports can be batched into a single
                  PCRpt message.";
            }

            leaf  num-rpt-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of reports of LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }

            leaf num-upd-sent{
                type yang:counter32;
                description
                  "The number of LSP updates sent.  A LSP update
                   corresponds 1:1 with an LSP object in a PCUpd
                   message. This might be greater than
                   num-pcupd-sent because multiple updates can
                   be batched into a single PCUpd message.";
            }

            leaf num-upd-rcvd{
                type yang:counter32;
                description
                 "The number of LSP Updates received.  A LSP update
                  corresponds 1:1 with an LSP object in a PCUpd
                  message.
                  This might be greater than num-pcupd-rcvd because
                  multiple updates can be batched into a single
                  PCUpd message.";
            }
```

```
            leaf  num-upd-rcvd-unknown{
                type yang:counter32;
                description
                  "The number of updates to unknown LSPs
                   received. An update to an unknown LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   present.";
            }

            leaf  num-upd-rcvd-undelegated{
                type yang:counter32;
                description
                  "The number of updates to not delegated LSPs
                   received. An update to an undelegated LSP is a
                   update whose LSP object does not contain the
                   PLSP-ID of any LSP that is currently
                   delegated to current PCEP session.";
            }

            leaf  num-upd-rcvd-error-sent{
                type yang:counter32;
                description
                  "The number of updates to LSPs received that were
                  responded by the local PCEP entity by sending a
                  PCErr message.";
            }
            container initiation {
                if-feature pce-initiated;
                description
                    "PCE-Initiated related statistics";
                leaf num-pcinitiate-sent{
                    type yang:counter32;
                    description
                     "The number of PCInitiate messages sent.";
                }

                leaf num-pcinitiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of PCInitiate messages received.";
                }

                leaf num-initiate-sent{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation sent via PCE.
                       A LSP initiation corresponds 1:1 with an LSP
```

```
                        object in a PCInitiate message. This might be
                        greater than num-pcinitiate-sent because
                        multiple initiations can be batched into a
                        single PCInitiate message.";
                }

                leaf num-initiate-rcvd{
                    type yang:counter32;
                    description
                      "The number of LSP Initiation received from
                       PCE.  A LSP initiation corresponds 1:1 with
                       an LSP object in a PCInitiate message. This
                       might be greater than num-pcinitiate-rcvd
                       because multiple initiations can be batched
                       into a single PCInitiate message.";
                }

                leaf  num-initiate-rcvd-error-sent{
                    type yang:counter32;
                    description
                      "The number of initiations of LSPs received
                       that were responded by the local PCEP entity
                       by sending a PCErr message.";
                }
            }
        }
    }//pcep-stats

    grouping lsp-state{
        description
            "This grouping defines the attributes for LSP in LSP-DB.
             These are the attributes specifically from the PCEP
             perspective";
        leaf plsp-id{
            type uint32{
                range "1..1048575";
            }
            description
                "A PCEP-specific identifier for the LSP.  A PCC
                 creates a unique PLSP-ID for each LSP that is
                 constant for the lifetime of a PCEP session.
                 PLSP-ID is 20 bits with 0 and 0xFFFFF are
                 reserved";
        }
        leaf pcc-id{
            type inet:ip-address;
            description
                "The local internet address of the PCC, that
```

```
                 generated the PLSP-ID.";
        }

        container lsp-ref{
            description
                "reference to ietf-te lsp state";

            leaf source {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:source";
                }
                description
                  "Tunnel sender address extracted from
                  SENDER_TEMPLATE  object";
                reference "RFC3209";
            }
            leaf destination {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:"
                         + "destination";
                }
                description
                    "Tunnel endpoint address extracted from
                    SESSION object";
                reference "RFC3209";
            }
            leaf tunnel-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:tunnel-id";
                }
                description
                    "Tunnel identifier used in the SESSION
                    that remains constant over the life
                    of the tunnel.";
                reference "RFC3209";
            }
            leaf lsp-id {
                type leafref {
                    path "/te:te/te:lsps-state/te:lsp/te:lsp-id";
                }
                description
                    "Identifier used in the SENDER_TEMPLATE
                    and the FILTER_SPEC that can be changed
                    to allow a sender to share resources with
                    itself.";
                reference "RFC3209";
            }
            leaf extended-tunnel-id {
```

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

        leaf admin-state{
            type boolean;
            description
                "The desired operational state";
        }
        leaf operational-state{
            type operational-state;
            description
                "The operational status of the LSP";
        }
        container delegated{
            description
                "The delegation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is delegated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "((../enabled == true)" +
                     " and " +
                     "((../../role == 'pcc')" +
                     " or " +
                    "(../../role == 'pcc-and-pce')))"
                {
                    error-message
                        "The PCEP entity must be PCC
                         and the LSP be delegated";
                    description
```

```
                        "When PCEP entity is PCC for
                        delegated LSP";
                }
                description
                    "The reference to the PCE peer to
                    which LSP is delegated";
            }
            leaf srp-id{
                type uint32;
                description
                    "The last SRP-ID-number associated with this
                    LSP.";
            }
        }
        container initiation {
            if-feature pce-initiated;
            description
                "The PCE initiation related parameters";
            leaf enabled{
                type boolean;
                description
                    "LSP is PCE-initiated or not";
            }
            leaf pce{
                type leafref {
                    path "/pcep-state/entity/peers/peer/addr";
                }
                must "(../enabled == true)"
                {
                    error-message
                        "The LSP must be PCE-Initiated";
                    description
                        "When the LSP must be PCE-Initiated";
                }
                description
                    "The reference to the PCE
                    that initiated this LSP";
            }
        }
        leaf symbolic-path-name{
            type string;
            description
                "The symbolic path name associated with the LSP.";
        }
        leaf last-error{
            type lsp-error;
            description
                "The last error for the LSP.";
```

```
        }
                leaf pst{
                        type pst;
                        default "rsvp-te";
                        description
                                "The Path Setup Type";

                }

    }//lsp-state

    grouping notification-instance-hdr {
        description
            "This group describes common instance specific data
             for notifications.";

        leaf peer-addr {
            type leafref {
                path "/pcep-state/entity/peers/peer/addr";
            }
            description
                "Reference to peer address";
        }

    }// notification-instance-hdr

    grouping notification-session-hdr {
        description
        "This group describes common session instance specific
         data for notifications.";

        leaf session-initiator {
            type leafref {
            path "/pcep-state/entity/peers/peer/sessions/" +
                 "session/initiator";
            }
            description
                "Reference to pcep session initiator leaf";
        }
    }// notification-session-hdr

    grouping stateful-pce-parameter {
        description
        "This group describes stateful PCE specific
         parameters.";
        leaf state-timeout{
            type uint32;
            units "seconds";
```

```
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before flushing
                 LSP state associated with that PCEP session
                 and reverting to operator-defined default
                 parameters or behaviours.";
        }
        leaf redelegation-timeout{
            type uint32;
            units "seconds";
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "When a PCEP session is terminated, a PCC
                 waits for this time period before revoking
                 LSP delegation to a PCE and attempting to
                 redelegate LSPs associated with the
                 terminated PCEP session to an alternate
                 PCE.";
        }
        leaf rpt-non-pcep-lsp{
            type boolean;
            must "((../role == 'pcc')" +
                 " or " +
                 "(../role == 'pcc-and-pce'))"
            {
                error-message "The PCEP entity must be PCC";
                description
                    "When PCEP entity is PCC";
            }
            description
                "If set, a PCC reports LSPs that are not
                controlled by any PCE (for example, LSPs
                that are statically configured at the
                PCC). ";
        }

    }

    grouping authentication {
        description "Authentication Information";
        choice auth-type-selection {
```

```
            description
                "Options for expressing authentication setting.";
            case auth-key-chain {
                leaf key-chain {
                    type key-chain:key-chain-ref;
                    description
                        "key-chain name.";
                }
            }
            case auth-key {
                leaf key {
                    type string;
                description
                    "Key string in ASCII format.";
                }
                container crypto-algorithm {
                    uses key-chain:crypto-algorithm-types;
                        description
                            "Cryptographic algorithm associated
                             with key.";
                }
            }
            case auth-tls {
                if-feature tls;
                container tls {
                    description
                        "TLS related information - TBD";
                }
            }
        }
    }

    grouping association {
        description
            "Generic Association parameters";
        leaf type {
            type "assoc-type";
            description
                "The PCEP association type";
        }
        leaf id {
            type uint16;
            description
                "PCEP Association ID";
        }
        leaf source {
          type inet:ip-address;
          description
```

```
                    "PCEP Association Source.";
        }
        leaf global-source {
          type uint32;
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type string;
            description
                "Additional information to
                support unique identification.";
        }
    }
    grouping association-ref {
        description
            "Generic Association parameters";
        leaf id {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/id";
            }
            description
                "PCEP Association ID";
        }
        leaf source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/source";
            }
          description
                "PCEP Association Source.";
        }
        leaf global-source {
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/global-source";
            }
          description
                "PCEP Association Global
                Source.";
        }
        leaf extended-id{
            type leafref {
                path "/pcep-state/entity/lsp-db/"
                + "association-list/extended-id";
            }
```

```
            description
                "Additional information to
                support unique identification.";
        }
    }
    /*
     * Configuration data nodes
     */
    container pcep{

        presence
            "The PCEP is enabled";

            description
            "Parameters for list of configured PCEP entities
             on the device.";

        container entity {

            description
                "The configured PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                mandatory true;
                description
                    "The local Internet address of this PCEP
                    entity.
                    If operating as a PCE server, the PCEP
                    entity listens on this address.
                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.
                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf enabled {
                type boolean;
                default true;
                description
                    "The administrative status of this PCEP
                     Entity.";
            }
```

```
            leaf role {
                type pcep-role;
                mandatory true;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP Entity role is not
                       known.
                     - pcc(1): this PCEP Entity is a PCC.
                     - pce(2): this PCEP Entity is a PCE.
                     - pcc-and-pce(3): this PCEP Entity is both
                       a PCC and a PCE.";
            }

            leaf description {
                type string;
                description
                    "Description of the PCEP entity configured
                     by the user";
            }

            uses info {
                description
                    "Local PCEP entity information";
            }


            container pce-info {
                must "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    error-message "The PCEP entity must be PCE";
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                        }


                description
                    "The Local PCE Entity PCE information";
```

```
            }

            uses pcep-entity-info {
                description
                    "The configuration related to the PCEP
                     entity.";
            }



            leaf pcep-notification-max-rate {
                type uint32;
                mandatory true;
                description
                    "This variable indicates the maximum number of
                     notifications issued per second. If events occur
                     more rapidly, the implementation may simply fail
                     to emit these notifications during that period,
                     or may queue them until an appropriate time. A
                     value of 0 means no notifications are emitted
                     and all should be discarded (that is, not
                     queued).";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The configured stateful parameters";
            }


            container peers{
                must "((../role == 'pcc')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    error-message
                        "The PCEP entity must be PCC";
```

```
                    description
                        "When PCEP entity is PCC, as remote
                         PCE peers are configured.";
                }
                description
                    "The list of configured peers for the
                     entity (remote PCE)";
                list peer{
                    key "addr";

                    description
                        "The peer configured for the entity.
                         (remote PCE)";

                    leaf addr {
                        type inet:ip-address;
                        description
                            "The local Internet address of this
                             PCEP peer.";
                    }

                    leaf description {
                        type string;
                        description
                            "Description of the PCEP peer
                             configured by the user";
                    }
                    uses info {
                        description
                            "PCE Peer information";
                    }
                    uses pce-info {
                        description
                            "PCE Peer information";
                    }

                    leaf delegation-pref{
                        if-feature stateful;
                        type uint8{
                            range "0..7";
                        }
                        must "(../../info/capability/stateful/active"
                             + "== true)"
                        {
                            error-message
                                "The Active Stateful PCE must be
                                 enabled";
                            description
```

```
                                "When PCEP entity is active stateful
                                 enabled";
                        }
                        description
                            "The PCE peer delegation preference.";
                    }
                    uses authentication {
                        description
                            "PCE Peer authentication";
                    }
                }//peer
            }//peers
        }//entity
    }//pcep

    /*
     * Operational data nodes
     */

    container pcep-state{
        config false;
        description
            "The list of operational PCEP entities on the
             device.";

        container entity{
            description
                "The operational PCEP entity on the device.";

            leaf addr {
                type inet:ip-address;
                description
                    "The local Internet address of this PCEP
                    entity.
                    If operating as a PCE server, the PCEP
                    entity listens on this address.
                    If operating as a PCC, the PCEP entity
                    binds outgoing TCP connections to this
                    address.
                    It is possible for the PCEP entity to
                    operate both as a PCC and a PCE Server, in
                    which case it uses this address both to
                    listen for incoming TCP connections and to
                    bind outgoing TCP connections.";
            }

            leaf index{
                type uint32;
```

```
                description
                    "The index of the operational PECP
                     entity";
            }


            leaf admin-status {
                type pcep-admin-status;
                description
                    "The administrative status of this PCEP Entity.
                     This is the desired operational status as
                     currently set by an operator or by default in
                     the implementation.  The value of enabled
                     represents the current status of an attempt
                     to reach this desired status.";
            }

            leaf oper-status {
                type pcep-admin-status;
                description
                   "The operational status of the PCEP entity.
                    Takes one of the following values.
                    - oper-status-up(1): the PCEP entity is
                      active.
                    - oper-status-down(2): the PCEP entity is
                      inactive.
                    - oper-status-going-up(3): the PCEP entity is
                      activating.
                    - oper-status-going-down(4): the PCEP entity is
                      deactivating.
                    - oper-status-failed(5): the PCEP entity has
                      failed and will recover when possible.
                    - oper-status-failed-perm(6): the PCEP entity
                      has failed and will not recover without
                      operator intervention.";
            }

            leaf role {
                type pcep-role;
                description
                    "The role that this entity can play.
                     Takes one of the following values.
                     - unknown(0): this PCEP entity role is
                       not known.
                     - pcc(1): this PCEP entity is a PCC.
                     - pce(2): this PCEP entity is a PCE.
                     - pcc-and-pce(3): this PCEP entity is
                       both a PCC and a PCE.";
```

```
            }

            uses info {
                description
                    "Local PCEP entity information";
            }

            container pce-info {
                when "((../role == 'pce')" +
                     " or " +
                     "(../role == 'pcc-and-pce'))"
                {
                    description
                        "When PCEP entity is PCE";
                }
                uses pce-info {
                    description
                        "Local PCE information";
                }
                                uses authentication {
                                        description
                                                "Local PCE authentication inform
ation";
                        }
                description
                    "The Local PCE Entity PCE information";
            }

            uses pcep-entity-info{
                description
                    "The operational information related to the
                     PCEP entity.";
            }

            container stateful-parameter{
                if-feature stateful;
                must "(../info/capability/stateful/active == true)"
                {
                    error-message
                        "The Active Stateful PCE must be enabled";
                    description
                        "When PCEP entity is active stateful
                         enabled";
                }
                uses stateful-pce-parameter;

                description
                    "The operational stateful parameters";
            }
```

```
            container lsp-db{
                if-feature stateful;
                description
                    "The LSP-DB";
                list association-list {
                    key "id source global-source extended-id";
                    description
                        "List of all PCEP associations";
                    uses association {
                        description
                            "The Association attributes";
                    }
                    list lsp {
                        key "plsp-id pcc-id";
                        description
                            "List of all LSP in this association";
                        leaf plsp-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/plsp-id";
                            }
                            description
                                "Reference to PLSP-ID in LSP-DB";
                        }
                        leaf pcc-id {
                            type leafref {
                                path "/pcep-state/entity/lsp-db/"
                                + "lsp/pcc-id";
                            }
                            description
                                "Reference to PCC-ID in LSP-DB";
                        }
                    }
                }
                list lsp{
                    key "plsp-id pcc-id";
                    description
                        "List of all LSPs in LSP-DB";
                    uses lsp-state{
                        description
                            "The PCEP specific attributes for
                             LSP-DB.";
                    }
                    list association-list {
                        key "id source global-source extended-id";
                        description
                            "List of all PCEP associations";
                        uses association-ref {
```

```
                            description
                                "Reference to the Association
                                 attributes";
                        }
                    }

                }
            }
            container peers{
                description
                        "The list of peers for the entity";

                list peer{
                    key "addr";

                    description
                        "The peer for the entity.";

                    leaf addr {
                        type inet:ip-address;
                        description
                            "The local Internet address of this PCEP
                             peer.";
                    }

                    leaf role {
                        type pcep-role;
                        description
                            "The role of the PCEP Peer.
                             Takes one of the following values.
                             - unknown(0): this PCEP peer role
                               is not known.
                             - pcc(1): this PCEP peer is a PCC.
                             - pce(2): this PCEP peer is a PCE.
                             - pcc-and-pce(3): this PCEP peer
                               is both a PCC and a PCE.";
                    }


                    uses info {
                        description
                            "PCEP peer information";
                    }


                    container pce-info {
                        when "((../role == 'pce')" +
                        " or " +
```

```
                        "(../role == 'pcc-and-pce'))"
                    {
                        description
                            "When PCEP entity is PCE";
                    }
                    uses pce-info {
                        description
                            "PCE Peer information";
                    }
                description
                    "The PCE Peer information";
                }

                leaf delegation-pref{
                    if-feature stateful;
                    type uint8{
                        range "0..7";
                    }
                    must "((../../role == 'pcc')" +
                         " or " +
                         "(../../role == 'pcc-and-pce'))"
                    {
                        error-message
                            "The PCEP entity must be PCC";
                        description
                            "When PCEP entity is PCC";
                    }
                    must "(../../info/capability/stateful/active"
                         + " == true)"
                    {
                        error-message
                            "The Active Stateful PCE must be
                             enabled";
                        description
                            "When PCEP entity is active stateful
                             enabled";
                    }
                    description
                        "The PCE peer delegation preference.";
                }

                uses authentication {
                    description
                        "PCE Peer authentication";
                }

                leaf discontinuity-time {
                    type yang:timestamp;
```

```
                    description
                        "The timestamp of the time when the
                         information and statistics were
                         last reset.";
                }

                leaf initiate-session {
                    type boolean;
                    description
                        "Indicates whether the local PCEP
                         entity initiates sessions to this peer,
                         or waits for the peer to initiate a
                         session.";
                }

                leaf session-exists{
                    type boolean;
                    description
                        "Indicates whether a session with
                         this peer currently exists.";
                }

                leaf num-sess-setup-ok{
                    type yang:counter32;
                    description
                        "The number of PCEP sessions successfully
                         successfully established with the peer,
                         including any current session.  This
                         counter is incremented each time a
                         session with this peer is successfully
                         established.";
                }

                leaf num-sess-setup-fail{
                    type yang:counter32;
                    description
                       "The number of PCEP sessions with the peer
                        that have been attempted but failed
                        before being fully established. This
                        counter is incremented each time a
                        session retry to this peer fails.";
                }

                leaf session-up-time{
                    type yang:timestamp;
                    must "(../num-sess-setup-ok != 0  or " +
                        "(../num-sess-setup-ok = 0  and "  +
                        "session-up-time = 0))" {
```

```
                            error-message
                                "Invalid Session Up timestamp";
                            description
                                "If num-sess-setup-ok is zero,
                                 then this leaf contains zero.";
                        }
                    description
                       "The timestamp value of the last time a
                        session with this peer was successfully
                        established.";
                }

                leaf session-fail-time{
                    type yang:timestamp;
                    must "(../num-sess-setup-fail != 0  or " +
                        "(../num-sess-setup-fail = 0  and "  +
                        "session-fail-time = 0))" {
                            error-message
                                "Invalid Session Fail timestamp";
                            description
                                "If num-sess-setup-fail is zero,
                                 then this leaf contains zero.";
                        }
                    description
                      "The timestamp value of the last time a
                       session with this peer failed to be
                       established.";
                }

                leaf session-fail-up-time{
                    type yang:timestamp;
                    must "(../num-sess-setup-ok != 0  or "     +
                        "(../num-sess-setup-ok = 0  and "     +
                        "session-fail-up-time = 0))" {
                            error-message
                                "Invalid Session Fail from
                                 Up timestamp";
                            description
                                "If num-sess-setup-ok is zero,
                                 then this leaf contains zero.";
                        }
                    description
                      "The timestamp value of the last time a
                       session with this peer failed from
                       active.";
                }

                container pcep-stats {
```

```
                        description
                            "The container for all statistics at peer
                             level.";
                        uses pcep-stats{
                            description
                                "Since PCEP sessions can be
                                ephemeral, the peer statistics tracks
                                a peer even when no PCEP session
                                currently exists to that peer. The
                                statistics contained are an aggregate
                                of the statistics for all successive
                                sessions to that peer.";
                        }

                        leaf num-req-sent-closed{
                            type yang:counter32;
                            description
                                "The number of requests that were
                                 sent to the peer and implicitly
                                 cancelled when the session they were
                                 sent over was closed.";
                        }

                        leaf  num-req-rcvd-closed{
                            type yang:counter32;
                            description
                                "The number of requests that were
                                 received from the peer and
                                 implicitly cancelled when the
                                 session they were received over
                                 was closed.";
                        }
                    }//pcep-stats


                    container sessions {
                        description
                            "This entry represents a single PCEP
                             session in which the local PCEP entity
                             participates.
                             This entry exists only if the
                             corresponding PCEP session has been
                             initialized by some event, such as
                             manual user configuration, auto-
                             discovery of a peer, or an incoming
                             TCP connection.";
```

```
                        list session {
                            key "initiator";

                            description
                                "The list of sessions, note that
                                 for a time being two sessions
                                 may exist for a peer";

                            leaf initiator {
                                type pcep-initiator;
                                description
                                    "The initiator of the session,
                                     that is, whether the TCP
                                     connection was initiated by
                                     the local PCEP entity or the
                                     peer.
                                     There is a window during
                                     session initialization where
                                     two sessions can exist between
                                     a pair of PCEP speakers, each
                                     initiated by one of the
                                     speakers. One of these
                                     sessions is always discarded
                                     before it leaves OpenWait state.
                                     However, before it is discarded,
                                     two sessions to the given peer
                                     appear transiently in this MIB
                                     module. The sessions are
                                     distinguished by who initiated
                                     them, and so this field is the
                                     key.";
                            }

                            leaf state-last-change {
                                type yang:timestamp;
                                description
                                    "The timestamp value at the
                                     time this session entered its
                                     current state as denoted by
                                     the state leaf.";
                            }

                            leaf state {
                                type pcep-sess-state;
                                description
                                    "The current state of the
                                     session.
                                     The set of possible states
```

```
                         excludes the idle state since
                         entries do not exist in the
                         idle state.";
                }

                leaf session-creation {
                    type yang:timestamp;
                    description
                        "The timestamp value at the
                         time this session was
                         created.";
                }

                leaf connect-retry {
                    type yang:counter32;
                    description
                         "The number of times that the
                          local PCEP entity has
                          attempted to establish a TCP
                          connection for this session
                          without success. The PCEP
                          entity gives up when this
                          reaches connect-max-retry.";
                }

                leaf local-id {
                    type uint32 {
                      range "0..255";
                    }
                    description
                         "The value of the PCEP session
                          ID used by the local PCEP
                          entity in the Open message
                          for this session.
                          If state is tcp-pending then
                          this is the session ID that
                          will be used in the Open
                          message. Otherwise, this is
                          the session ID that was sent
                          in the Open message.";
                }

                leaf remote-id {
                    type uint32 {
                        range "0..255";
                    }
                    must "((../state != 'tcp-pending'" +
                         "and " +
```

```
                                    "../state != 'open-wait' )" +
                                    "or " +
                                    "((../state = 'tcp-pending'" +
                                    " or " +
                                    "../state = 'open-wait' )" +
                                    "and remote-id = 0))" {
                                       error-message
                                           "Invalid remote-id";
                                       description
                                            "If state is tcp-pending
                                             or open-wait then this
                                             leaf is not used and
                                             MUST be set to zero.";
                                   }
                               description
                                    "The value of the PCEP session
                                     ID used by the peer in its
                                     Open message for this
                                     session.";
                           }

                           leaf keepalive-timer {
                               type uint32 {
                                 range "0..255";
                               }
                               units "seconds";
                               must "(../state = 'session-up'" +
                                    "or " +
                                    "(../state != 'session-up'" +
                                    "and keepalive-timer = 0))" {
                                       error-message
                                           "Invalid keepalive
                                            timer";
                                       description
                                           "This field is used if
                                            and only if state is
                                            session-up. Otherwise,
                                            it is not used and
                                            MUST be set to
                                            zero.";
                                   }
                               description
                                    "The agreed maximum interval at
                                     which the local PCEP entity
                                     transmits PCEP messages on this
                                     PCEP session.  Zero means that
                                     the local PCEP entity never
                                     sends Keepalives on this
```

```
                          session.";
                }

                leaf peer-keepalive-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "(../state = 'session-up'" +
                         "or "   +
                         "(../state != 'session-up'" +
                         "and " +
                         "peer-keepalive-timer = 0))" {
                            error-message
                                "Invalid Peer keepalive
                                 timer";
                            description
                                "This field is used if
                                 and only if state is
                                 session-up. Otherwise,
                                 it is not used and MUST
                                 be set to zero.";
                        }
                    description
                         "The agreed maximum interval at
                          which the peer transmits PCEP
                          messages on this PCEP session.
                          Zero means that the peer never
                          sends Keepalives on this
                          session.";
                }

                leaf dead-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    description
                         "The dead timer interval for
                          this PCEP session.";
                }

                leaf peer-dead-timer {
                    type uint32 {
                      range "0..255";
                    }
                    units "seconds";
                    must "((../state != 'tcp-pending'" +
```

```
                            "and " +
                            "../state != 'open-wait' )" +
                            "or " +
                            "((../state = 'tcp-pending'" +
                            " or " +
                            "../state = 'open-wait' )" +
                            "and " +
                            "peer-dead-timer = 0))" {
                               error-message
                                   "Invalid Peer Dead
                                    timer";
                               description
                                   "If state is tcp-
                                    pending or open-wait
                                    then this leaf is not
                                    used and MUST be set to
                                    zero.";
                           }
                       description
                            "The peer's dead-timer interval
                             for this PCEP session.";
                   }

                   leaf ka-hold-time-rem {
                       type uint32 {
                         range "0..255";
                       }
                       units "seconds";
                       must "((../state != 'tcp-pending'" +
                            "and " +
                            "../state != 'open-wait' ) " +
                            "or " +
                            "((../state = 'tcp-pending'" +
                            "or " +
                            "../state = 'open-wait' )" +
                            "and " +
                            "ka-hold-time-rem = 0))" {
                               error-message
                                   "Invalid Keepalive hold
                                    time remaining";
                               description
                                   "If state is tcp-pending
                                    or open-wait then this
                                    field is not used and
                                    MUST be set to zero.";
                           }
                       description
                            "The keep alive hold time
```

```
                                                  remaining for this session.";
                                        }

                                        leaf overloaded {
                                            type boolean;
                                            description
                                                 "If the local PCEP entity has
                                                  informed the peer that it is
                                                  currently overloaded, then this
                                                  is set to true.  Otherwise, it
                                                  is set to false.";
                                        }

                                        leaf overload-time {
                                            type uint32;
                                            units "seconds";
                                            must "(../overloaded = true or" +
                                                "(../overloaded != true and" +
                                                " overload-time = 0))" {
                                                    error-message
                                                        "Invalid overload-time";
                                                    description
                                                        "This field is only used
                                                         if overloaded is set to
                                                         true. Otherwise, it is
                                                         not used and MUST be set
                                                         to zero.";
                                                }
                                            description
                                                 "The interval of time that is
                                                  remaining until the local PCEP
                                                  entity will cease to be
                                                  overloaded on this session.";
                                        }

                                        leaf peer-overloaded {
                                            type boolean;
                                            description
                                                 "If the peer has informed the
                                                  local PCEP entity that it is
                                                  currently overloaded, then this
                                                  is set to true. Otherwise, it
                                                  is set to false.";
                                        }

                                        leaf peer-overload-time {
                                            type uint32;
                                            units "seconds";
```

```
                                must "(../peer-overloaded = true" +
                                     " or " +
                                     "(../peer-overloaded != true" +
                                     " and " +
                                     "peer-overload-time = 0))" {
                                        error-message
                                            "Invalid peer overload
                                             time";
                                        description
                                            "This field is only used
                                             if peer-overloaded is
                                             set to true. Otherwise,
                                             it is not used and MUST
                                             be set to zero.";
                                    }
                                description
                                     "The interval of time that is
                                      remaining until the peer will
                                      cease to be overloaded.  If it
                                      is not known how long the peer
                                      will stay in overloaded state,
                                      this leaf is set to zero.";
                            }
                            leaf lspdb-sync {
                                if-feature stateful;
                                type sync-state;
                                description
                                    "The LSP-DB state synchronization
                                    status.";
                            }
                            leaf discontinuity-time {
                                type yang:timestamp;
                                description
                                     "The timestamp value of the time
                                      when the statistics were last
                                      reset.";
                            }

                            container pcep-stats {
                                description
                                    "The container for all statistics
                                     at session level.";
                                uses pcep-stats{
                                    description
                                        "The statistics contained are
                                         for the current sessions to
                                         that peer. These are lost
                                         when the session goes down.
```

```
                                         ";
                                }
                            }//pcep-stats

                        } // session
                    } // sessions
                }//peer
            }//peers
        }//entity
    }//pcep-state

    /*
     * Notifications
     */
    notification pcep-session-up {
        description
            "This notification is sent when the value of
             '/pcep/pcep-state/peers/peer/sessions/session/state'
             enters the 'session-up' state.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.
                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification

    notification pcep-session-down {
        description
            "This notification is sent when the value of
             '/pcep/pcep-state/peers/peer/sessions/session/state'
             leaves the 'session-up' state.";

        uses notification-instance-hdr;
```

```
        leaf session-initiator {
            type pcep-initiator;
            description
                "The initiator of the session.";
        }

        leaf state-last-change {
            type yang:timestamp;
            description
                "The timestamp value at the time this session entered
                its current state as denoted by the state leaf.";
        }

        leaf state {
            type pcep-sess-state;
            description
                "The current state of the session.
                 The set of possible states excludes the idle state
                 since entries do not exist in the idle state.";
        }
    } //notification

    notification pcep-session-local-overload {
        description
            "This notification is sent when the local PCEP entity
            enters overload state for a peer.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer that
                 it is currently overloaded, then this is set to
                 true. Otherwise, it is set to false.";
        }

        leaf overload-time {
            type uint32;
            units "seconds";
            must "(../overloaded = true or "    +
                "(../overloaded != true  and "  +
                "overload-time = 0))" {
                    error-message
                        "Invalid overload-time";
                    description
```

```
                        "This field is only used if overloaded is
                         set to true. Otherwise, it is not used
                         and MUST be set to zero.";
                }
            description
                 "The interval of time that is remaining until the
                 local PCEP entity will cease to be overloaded on
                 this session.";
        }
    } //notification

    notification pcep-session-local-overload-clear {
        description
            "This notification is sent when the local PCEP entity
            leaves overload state for a peer.";

        uses notification-instance-hdr;

        leaf overloaded {
            type boolean;
            description
                 "If the local PCEP entity has informed the peer
                 that it is currently overloaded, then this is set
                 to true.  Otherwise, it is set to false.";
        }
    } //notification

    notification pcep-session-peer-overload {
        description
            "This notification is sent when a peer enters overload
            state.";

        uses notification-instance-hdr;

        uses notification-session-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }

        leaf peer-overload-time {
            type uint32;
            units "seconds";
            must "(../peer-overloaded = true or "    +
```

```
                "(../peer-overloaded != true  and "  +
                "peer-overload-time = 0))" {
                    error-message
                        "Invalid peer-overload-time";
                    description
                        "This field is only used if
                         peer-overloaded is set to true.
                         Otherwise, it is not used and MUST
                         be set to zero.";
                }
            description
                "The interval of time that is remaining until the
                peer will cease to be overloaded.  If it is not known
                how long the peer will stay in overloaded state, this
                leaf is set to zero.";
        }
    } //notification

    notification pcep-session-peer-overload-clear {
        description
            "This notification is sent when a peer leaves overload
            state.";

        uses notification-instance-hdr;

        leaf peer-overloaded {
            type boolean;
            description
                "If the peer has informed the local PCEP entity that
                it is currently overloaded, then this is set to true.
                Otherwise, it is set to false.";
        }
    } //notification
}//module

<CODE ENDS>
```

9. Security Considerations

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

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

TBD: List specific Subtrees and data nodes and their sensitivity/ vulnerability.

10. Manageability Considerations

10.1. Control of Function and Policy

10.2. Information and Data Models

10.3. Liveness Detection and Monitoring

10.4. Verify Correct Operations

10.5. Requirements On Other Protocols

10.6. Impact On Network Operations

11. IANA Considerations

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-pcep

Registrant Contact: The PCE WG of the IETF.

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

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

```
Name:         ietf-pcep
Namespace:    urn:ietf:params:xml:ns:yang:ietf-pcep
Prefix:       pcep
Reference:    This I-D
```

12. Acknowledgements

The initial document is based on the PCEP MIB [RFC7420]. Further this document structure is based on Routing Yang Module [I-D.ietf-netmod-routing-cfg]. We would like to thank the authors of aforementioned documents.

13. References

13.1. Normative References

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

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

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

[I-D.ietf-pce-stateful-pce] Crabbe, E., Minei, I., Medved, J., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-14 (work in progress), March 2016.

[I-D.ietf-pce-pce-initiated-lsp] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-05 (work in progress), October 2015.

[I-D.ietf-pce-lsp-setup-type]
Sivabalan, S., Medved, J., Minei, I., Crabbe, E., Varga, R., Tantsura, J., and J. Hardwick, "Conveying path setup type in PCEP messages", draft-ietf-pce-lsp-setup-type-03 (work in progress), June 2015.

[I-D.ietf-pce-segment-routing]
Sivabalan, S., Medved, J., Filsfils, C., Crabbe, E., Lopez, V., Tantsura, J., Henderickx, W., and J. Hardwick, "PCEP Extensions for Segment Routing", draft-ietf-pce-segment-routing-07 (work in progress), March 2016.

13.2. Informative References

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

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC7420] Koushik, A., Stephan, E., Zhao, Q., King, D., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module", RFC 7420, DOI 10.17487/RFC7420, December 2014, <http://www.rfc-editor.org/info/rfc7420>.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.

[I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-06 (work in progress), March 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-03 (work in progress), March 2016.

Appendix A. Contributor Addresses

Rohit Pobbathi
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: rohit.pobbathi@huawei.com

Vinod KumarS
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: vinods.kumar@huawei.com

Zafar Ali
Cisco Systems
Canada

EMail: zali@cisco.com

Xufeng Liu
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

EMail: xufeng.liu@ericsson.com

Young Lee
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023, USA

Phone: (469) 277-5838
EMail: leeyoung@huawei.com

Udayasree Palle
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: udayasree.palle@huawei.com

Xian Zhang
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R.China

EMail: zhang.xian@huawei.com

Avantika
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: avantika.sushilkumar@huawei.com

Authors' Addresses

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore, Karnataka 560066
India

EMail: dhruv.ietf@gmail.com

Jonathan Hardwick
Metaswitch
100 Church Street
Enfield EN2 6BQ
UK

EMail: jonathan.hardwick@metaswitch.com

Vishnu Pavan Beeram
Juniper Networks
USA

EMail: vbeeram@juniper.net

Jeff Tantsura
USA

EMail: jefftant@gmail.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

K. Raza
R. Asati
Cisco Systems, Inc.

X. Liu
Ericsson

S. Esale
Juniper Networks

X. Chen
Huawei Technologies

H. Shah
Ciena Corporation

July 8, 2016

YANG Data Model for MPLS LDP and mLDP
draft-raza-mpls-ldp-mldp-yang-04

Abstract

This document describes a YANG data model for Multi-Protocol Label Switching (MPLS) Label Distribution Protocol (LDP) and Multipoint LDP (mLDP).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is one of the network management protocols that defines mechanisms to manage network devices. YANG [RFC6020] is a modular language that represents data structures in an XML tree format, and is used as a data modelling language for the NETCONF.

This document introduces a YANG data model for MPLS Label Distribution Protocol (LDP) [RFC5036] and Multipoint LDP (mLDP) [RFC6388]. For LDP, it also covers LDP IPv6 [RFC7552] and LDP capabilities [RFC5561].

The data model is defined for following constructs that are used for managing the protocol:

o Configuration

o Operational State

o Executables (Actions)

o Notifications

This document is organized to define the data model for each of the above constructs (configuration, state, action, and notifications) in the sequence as listed earlier. Given that mLDP is tightly coupled with LDP, mLDP data model is defined under LDP tree and in the same sequence as listed above.

2. Specification of Requirements

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

In this document, the word "IP" is used to refer to both IPv4 and IPv6, unless otherwise explicitly stated. For example, "IP address family" means and be read as "IPv4 and/or IPv6 address family"

3. LDP YANG Model

3.1. Overview

This document defines a new module named "ietf-mpls-ldp" for LDP/mLDP data model where this module augments /rt:routing/rt:control-plane-protocols that is defined in [I-D.ietf-netmod-routing-cfg].

There are four main containers in "ietf-mpls-ldp" module as follows:

- o Read-Write parameters for configuration (Discussed in Section 3.2)
- o Read-only parameters for operational state (Discussed in Section 3.3)
- o Notifications for events (Discussed in Section 3.4)
- o RPCs for executing commands to perform some action (Discussed in Section 3.5)

For the configuration and state data, this model follows the similar approach described in [I-D.openconfig-netmod-opstate] to represent the configuration (intended state) and operational (applied and derived) state. This means that for every configuration (rw) item, there is an associated (ro) item under "state" container to represent the applied state. Furthermore, protocol derived state is also kept under "state" tree corresponding to the protocol area (discovery, peer etc.). [Ed note: This document will be (re-)aligned with [I-D.openconfig-netmod-opstate] once that specification is adopted as a WG document]

Following diagram depicts high level LDP yang tree organization and hierarchy:

```
module: ietf-mpls-ldp
     +-- rw routing
       +-- rw control-plane-protocols
          +-- rw mpls-ldp
              +-- rw global
              |   +-- rw config
              |       +-- rw ...
              |   +-- ro state
              |       +-- ro ...
              |       .
              +-- rw ...
              |
              |
              +-- rw ...
              ...

rpcs:
   +-- x mpls-ldp-rpc
   +-- x . . . . .

notifications:
   +--- n mpls-ldp-notif
   +--- n ...
```

Figure 1

Before going into data model details, it is important to take note of the following points:

- o This module aims to address only the core LDP/mLDP parameters as per RFC specification, as well as some widely used and deployed non-RFC features (such as label policies, session authentication etc). Any vendor specific feature should be defined in a vendor-specific augmentation of this model.

- o Multi-topology LDP [RFC7307] and Multi-topology mLDP [I-D.iwijnand-mpls-mldp-multi-topology] are beyond the scope of this document.

- o This module does not cover any applications running on top of LDP and mLDP, nor does it cover any OAM procedures for LDP and mLDP.

- o This model is a VPN Forwarding and Routing (VRF)-centric model. It is important to note that [RFC4364] defines VRF tables and default forwarding tables as different, however from a yang modelling perspective this introduces unnecessary complications, hence we are treating the default forwarding table as just another VRF.

- o A "network-instance" as defined in [I-D.rtgyangdt-rtgwg-ni-model] refers to a VRF instance (both default and non-default) within the scope of this model.

- o This model supports two address-families, namely "ipv4" and "ipv6".

- o This model assumes platform-wide label space (i.e. label space Id of zero). However, when Upstream Label assignment [RFC6389] is in use, an upstream assigned label is looked up in a Context-Specific label space as defined in [RFC5331].

- o The label and peer policies (including filters) are defined using a prefix-list. When used for a peer policy, the prefix refers to the LSR Id of the peer. The prefix-list is referenced from routing-policy model as defined in [I-D.ietf-rtgwg-policy-model].

- o The use of grouping (templates) for bundling and grouping the configuration items is not employed in current revision, and is a subject for consideration in future.

- o This model uses the terms LDP "neighbor"/"adjacency", "session", and "peer" with the following semantics:

  * Neighbor/Adjacency: An LDP enabled LSR that is discovered through LDP discovery mechanisms.

  * Session: An LDP neighbor with whom a TCP connection has been established.

  * Peer: An LDP session which has successfully progressed beyond its initialization phase and is either already exchanging the bindings or is ready to do so.

  It is to be noted that LDP Graceful Restart mechanisms defined in [RFC3478] allow keeping the exchanged bindings for some time after a session goes down with a peer. We call such a state -- i.e. keeping peer bindings without established or recovered peering -- a "stale" peer. When used in this document, the above terms will refer strictly to the semantics and definitions defined for them.

A graphical representation of LDP YANG data model is presented in Figure 3, Figure 5, Figure 11, and Figure 12. Whereas, the actual model definition in YANG is captured in Section 6.

While presenting the YANG tree view and actual .yang specification, this document assumes the reader is familiar with the concepts of YANG modeling, its presentation and its compilation.

3.2. Configuration

This specification defines the configuration parameters for base LDP as specified in [RFC5036] and LDP IPv6 [RFC7552]. Moreover, it incorporates provisions to enable LDP Capabilities [RFC5561], and defines some of the most significant and commonly used capabilities such as Typed Wildcard FEC [RFC5918], End-of-LIB [RFC5919], and LDP Upstream Label Assignment [RFC6389].

This specification supports VRF-centric configuration. For implementations that support protocol-centric configuration, with provision for inheritance and items that apply to all vrfs, we recommend an augmentation of this model such that any protocol-centric or all-vrf configuration is defined under their designated containers within the standard network-instance (please see Section 3.2.2)

This model augments /rt:routing/rt:control-plane-protocols that is defined in [I-D.ietf-netmod-routing-cfg]. For LDP interfaces, this model refers the MPLS interface as defined under MPLS base specification [I-D.saad-mpls-base-yang]. Furthermore, as mentioned earlier, the configuration tree presents read-write intended configuration leave/items as well as read-only state of the applied configuration. The former is listed under "config" container and latter under "state" container.

Following is high-level configuration organization for LDP/mLDP:

```
       module: ietf-mpls-ldp
          +-- routing
           +-- control-plane-protocols
                 +-- mpls-ldp
                     +-- global
                     |    +-- ...
                     |    +-- ...
                     |    +-- address-family* [afi]
                     |        +-- . . .
                     |        +-- . . .
                     |    +-- discovery
                     |        +-- . . .
                     +-- peers
                          +-- ...
                          +-- ...
```

Figure 2

Given the configuration hierarchy, the model allows inheritance such that an item in a child tree is able to derive value from a similar or related item in one of the parent. For instance, hello holdtime can be configured per-VRF or per-VRF-interface, thus allowing inheritance as well flexibility to override with a different value at any child level.

Following is a simplified graphical representation of the data model for LDP configuration

```
   +--rw mpls-ldp!
      +--rw global
      |  +--rw config
      |  |  +--rw capability
      |  |  |  +--rw end-of-lib {capability-end-of-lib}?
      |  |  |  |  +--rw enable?   boolean
      |  |  |  +--rw typed-wildcard-fec {capability-typed-wildcard-fec}?
      |  |  |  |  +--rw enable?   boolean
      |  |  |  +--rw upstream-label-assignment {capability-upstream-label-assign
ment}?
      |  |  |     +--rw enable?   boolean
      |  |  +--rw graceful-restart
      |  |  |  +--rw enable?              boolean
      |  |  |  +--rw helper-enable?       boolean {graceful-restart-helper-mod
e}?
      |  |  |  +--rw reconnect-time?      uint16
      |  |  |  +--rw recovery-time?       uint16
      |  |  |  +--rw forwarding-holdtime?   uint16
      |  |  +--rw igp-synchronization-delay?   uint16
      |  |  +--rw lsr-id?                      yang:dotted-quad
```

```
     |  +--rw address-family* [afi]
     |  |  +--rw afi       ldp-address-family
     |  |  +--rw config
     |  |     +--rw enable?         boolean
     |  |     +--rw label-policy
     |  |     |  +--rw independent-mode
     |  |     |  |  +--rw assign {policy-label-assignment-config}?
     |  |     |  |  |  +--rw (prefix-option)?
     |  |     |  |  |     |  +--rw prefix-list?        prefix-list-ref
     |  |     |  |  |        +--rw host-routes-only?   boolean
     |  |     |  |  +--rw advertise
     |  |     |  |  |  +--rw explicit-null
     |  |     |  |  |  |  +--rw enable?        boolean
     |  |     |  |  |  |  +--rw prefix-list?   prefix-list-ref
     |  |     |  |  |  +--rw prefix-list?     prefix-list-ref
     |  |     |  |  +--rw accept
     |  |     |  |     +--rw prefix-list?   prefix-list-ref
     |  |     |  +--rw ordered-mode {policy-ordered-label-config}?
     |  |     |     +--rw egress-lsr
     |  |     |     |  +--rw prefix-list?   prefix-list-ref
     |  |     |     +--rw advertise
     |  |     |     |  +--rw prefix-list?   prefix-list-ref
     |  |     |     +--rw accept
     |  |     |        +--rw prefix-list?   prefix-list-ref
     |  |     +--rw ipv4
     |  |     |  +--rw transport-address?   inet:ipv4-address
     |  |     +--rw ipv6
     |  |        +--rw transport-address?   inet:ipv6-address
     |  +--rw discovery
     |  |  +--rw interfaces
     |  |  |  +--rw config
     |  |  |  |  +--rw hello-holdtime?   uint16
     |  |  |  |  +--rw hello-interval?   uint16
     |  |  |  +--rw interface* [interface]
     |  |  |     +--rw interface         mpls-interface-ref
     |  |  |     +--rw config
     |  |  |     |  +--rw hello-holdtime?             uint16
     |  |  |     |  +--rw hello-interval?             uint16
     |  |  |     |  +--rw igp-synchronization-delay?   uint16 {per-interface-ti
mer-config}?
     |  |  |     +--rw address-family* [afi]
     |  |  |        +--rw afi       ldp-address-family
     |  |  |        +--rw config
     |  |  |           +--rw enable?   boolean
     |  |  |           +--rw ipv4
     |  |  |           |  +--rw transport-address?   union
     |  |  |           +--rw ipv6
     |  |  |              +--rw transport-address?   union
     |  |  +--rw targeted
```

```
   |  |     +--rw config
   |  |     |  +--rw hello-holdtime?   uint16
   |  |     |  +--rw hello-interval?   uint16
   |  |     |  +--rw hello-accept {policy-extended-discovery-config}?
   |  |     |     +--rw enable?          boolean
   |  |     |     +--rw neighbor-list?   neighbor-list-ref
   |  |     +--rw address-family* [afi]
   |  |        +--rw afi      ldp-address-family
   |  |        +--rw ipv4
   |  |        |  +--rw target* [adjacent-address]
   |  |        |     +--rw adjacent-address    inet:ipv4-address
   |  |        |     +--rw config
   |  |        |        +--rw enable?          boolean
   |  |        |        +--rw local-address?   inet:ipv4-address
   |  |        +--rw ipv6
   |  |           +--rw target* [adjacent-address]
   |  |              +--rw adjacent-address    inet:ipv6-address
   |  |              +--rw config
   |  |                 +--rw enable?          boolean
   |  |                 +--rw local-address?   inet:ipv6-address
   |  +--rw forwarding-nexthop {forwarding-nexthop-config}?
   |  |  +--rw interfaces
   |  |     +--rw interface* [interface]
   |  |        +--rw interface         mpls-interface-ref
   |  |        +--rw address-family* [afi]
   |  |           +--rw afi       ldp-address-family
   |  |           +--rw config
   |  |              +--rw ldp-disable?    boolean
   |  +--rw label-policy
   |     +--rw independent-mode
   |     |  +--rw assign {policy-label-assignment-config}?
   |     |  |  +--rw (prefix-option)?
   |     |  |        +--rw prefix-list?        prefix-list-ref
   |     |  |        +--rw host-routes-only?   boolean
   |     |  +--rw advertise
   |     |  |  +--rw explicit-null
   |     |  |  |  +--rw enable?        boolean
   |     |  |  |  +--rw prefix-list?   prefix-list-ref
   |     |  |  +--rw prefix-list?     prefix-list-ref
   |     |  +--rw accept
   |     |     +--rw prefix-list?   prefix-list-ref
   |     +--rw ordered-mode {policy-ordered-label-config}?
   |        +--rw egress-lsr
   |        |  +--rw prefix-list?   prefix-list-ref
   |        +--rw advertise
   |        |  +--rw prefix-list?   prefix-list-ref
   |        +--rw accept
   |           +--rw prefix-list?   prefix-list-ref
```

```
      +--rw peers
         +--rw config
         |  +--rw session-authentication-md5-password?   string
         |  +--rw session-ka-holdtime?                    uint16
         |  +--rw session-ka-interval?                    uint16
         |  +--rw session-downstream-on-demand {session-downstream-on-demand-con
fig}?
         |     +--rw enable?      boolean
         |     +--rw peer-list?   peer-list-ref
         +--rw peer* [lsr-id]
            +--rw lsr-id    yang:dotted-quad
            +--rw config
               +--rw admin-down?                            boolean
               +--rw capability
               +--rw label-policy
               |  +--rw advertise
               |  |  +--rw prefix-list?   prefix-list-ref
               |  +--rw accept
               |     +--rw prefix-list?   prefix-list-ref
               +--rw session-authentication-md5-password?   string
               +--rw graceful-restart
               |  +--rw enable?           boolean
               |  +--rw reconnect-time?   uint16
               |  +--rw recovery-time?    uint16
               +--rw session-ka-holdtime?                   uint16
               +--rw session-ka-interval?                   uint16
               +--rw address-family
                  +--rw ipv4
                  |  +--rw label-policy
                  |     +--rw advertise
                  |     |  +--rw prefix-list?   prefix-list-ref
                  |     +--rw accept
                  |        +--rw prefix-list?   prefix-list-ref
                  +--rw ipv6
                     +--rw label-policy
                        +--rw advertise
                        |  +--rw prefix-list?   prefix-list-ref
                        +--rw accept
                           +--rw prefix-list?   prefix-list-ref
```

Figure 3

3.2.1. Configuration Hierarchy

The LDP configuration container is logically divided into following high-level config areas:

```
      Per-VRF parameters
          o Global parameters
          o Per-address-family parameters
          o LDP Capabilities parameters
          o Hello Discovery parameters
              - interfaces
                 - Per-interface:
                    Global
                    Per-address-family
              - targeted
                 - Per-target
         o Peer parameters
              - Global
              - Per-peer
                 Per-address-family
                 Capabilities parameters
         o Forwarding parameters
```

Figure 4

Following subsections briefly explain these configuration areas.

3.2.1.1. Per-VRF parameters

LDP module resides under an network-instance and the scope of any LDP configuration defined under this tree is per network-instance (per-VRF). This configuration is further divided into sub categories as follows.

3.2.1.1.1. Per-VRF global parameters

There are configuration items that are available directly under a VRF instance and do not fall under any other sub tree. Example of such a parameter is LDP LSR id that is typically configured per VRF. To keep legacy LDP features and applications working in an LDP IPv4 networks with this model, this document recommends an operator to pick a routable IPv4 unicast address as an LSR Id.

3.2.1.1.2. Per-VRF Capabilities parameters

This container falls under global tree and holds the LDP capabilities that are to be enabled for certain features. By default, an LDP capability is disabled unless explicitly enabled. These capabilities are typically used to negotiate with LDP peer(s) the support/non-support related to a feature and its parameters. The scope of a capability enabled under this container applies to all LDP peers in the given VRF instance. There is also a peer level capability container that is provided to override a capability that is enabled/ specified at VRF level.

3.2.1.1.3. Per-VRF Per-Address-Family parameters

Any LDP configuration parameter related to IP address family (AF) whose scope is VRF wide is configured under this tree. The examples of per-AF parameters include enabling LDP for an address family, prefix-list based label policies, and LDP transport address.

3.2.1.1.4. Per-VRF Hello Discovery parameters

This container is used to hold LDP configuration related to Hello and discovery process for both basic (link) and extended (targeted) discovery.

The "interfaces" is a container to configure parameters related to VRF interfaces. There are parameters that apply to all interfaces (such as hello timers), as well as parameters that can be configured per-interface. Hence, an interface list is defined under "interfaces" container. The model defines parameters to configure per-interface non AF related items, as well as per-interface per-AF items. The example of former is interface hello timers, and example of latter is enabling hellos for a given AF under an interface.

The "targeted" container under a VRF instance allows to configure LDP targeted discovery related parameters. Within this container, the "target" list provides a mean to configure multiple target addresses to perform extended discovery to a specific destination target, as well as to fine-tune the per-target parameters.

3.2.1.1.5. Per-VRF Peer parameters

This container is used to hold LDP configuration related to LDP sessions and peers under a VRF instance. This container allows to configure parameters that either apply on VRF’s all peers or a subset (peer-list) of VRF peers. The example of such parameters include authentication password, session KA timers etc. Moreover, the model also allows per-peer parameter tuning by specifying a "peer" list under the "peers" container. A peer is uniquely identified using its LSR Id and hence LSR Id is the key for peer list

Like per-interface parameters, some per-peer parameters are AF-agnostic (i.e. either non AF related or apply to both IP address families), and some that belong to an AF. The example of former is per-peer session password configuration, whereas the example of latter is prefix-list based label policies (inbound and outbound) that apply to a given peer.

3.2.1.1.6. Per-VRF Forwarding parameters

This container is used to hold configuration used to control LDP forwarding behavior under a VRF instance. One example of a configuration under this container is when a user wishes to enable neighbor discovery on an interface but wishes to disable use of the same interface as forwarding nexthop. This example configuration makes sense only when there are more than one LDP enabled interfaces towards the neighbor.

3.2.2. All-VRFs Configuration

[Ed note: TODO]

3.3. Operational State

Operational state of LDP can be queried and obtained from read-only state containers that fall under the same tree (/rt:routing/ rt:control-plane-protocols/) as the configuration.

Please note this state tree refers both the configuration "applied" state as well as the "derived" state related to the protocol. [Ed note: This is where this model differs presently from [I-D.openconfig-netmod-opstate] and subject to alignment in later revisions]

Following is a simplified graphical representation of the data model for LDP operational state.

```
module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
   +--rw mpls-ldp!
      +--rw global
      |  +--ro state
      |  |  +--ro capability
      |  |  |  +--ro end-of-lib {capability-end-of-lib}?
      |  |  |  |  +--ro enable?   boolean
      |  |  |  +--ro typed-wildcard-fec {capability-typed-wildcard-fec}?
      |  |  |  |  +--ro enable?   boolean
      |  |  |  +--ro upstream-label-assignment {capability-upstream-label-assign
ment}?
      |  |  |  |  +--ro enable?   boolean
      |  |  +--ro graceful-restart
      |  |  |  +--ro enable?                 boolean
      |  |  |  +--ro helper-enable?          boolean {graceful-restart-helper-mod
e}?
      |  |  |  +--ro reconnect-time?         uint16
      |  |  |  +--ro recovery-time?          uint16
      |  |  |  +--ro forwarding-holdtime?    uint16
```

```
      |  |  +--ro igp-synchronization-delay?   uint16
      |  |  +--ro lsr-id?                      yang:dotted-quad
      |  +--rw address-family* [afi]
      |  |  +--rw afi       ldp-address-family
      |  |  +--ro state
      |  |     +--ro enable?         boolean
      |  |     +--ro label-policy
      |  |     |  +--ro independent-mode
      |  |     |  |  +--ro assign {policy-label-assignment-config}?
      |  |     |  |  |  +--ro (prefix-option)?
      |  |     |  |  |     +--:(prefix-list)
      |  |     |  |  |     |  +--ro prefix-list?        prefix-list-ref
      |  |     |  |  |     +--:(host-routes-only)
      |  |     |  |  |        +--ro host-routes-only?   boolean
      |  |     |  |  +--ro advertise
      |  |     |  |  |  +--ro explicit-null
      |  |     |  |  |  |  +--ro enable?        boolean
      |  |     |  |  |  |  +--ro prefix-list?   prefix-list-ref
      |  |     |  |  |  +--ro prefix-list?     prefix-list-ref
      |  |     |  |  +--ro accept
      |  |     |  |     +--ro prefix-list?   prefix-list-ref
      |  |     |  +--ro ordered-mode {policy-ordered-label-config}?
      |  |     |     +--ro egress-lsr
      |  |     |     |  +--ro prefix-list?   prefix-list-ref
      |  |     |     +--ro advertise
      |  |     |     |  +--ro prefix-list?   prefix-list-ref
      |  |     |     +--ro accept
      |  |     |        +--ro prefix-list?   prefix-list-ref
      |  |     +--ro ipv4
      |  |     |  +--ro transport-address?   inet:ipv4-address
      |  |     |  +--ro bindings
      |  |     |     +--ro address* [address]
      |  |     |     |  +--ro address      inet:ipv4-address
      |  |     |     |  +--ro advertisement-type?   advertised-received
      |  |     |     |  +--ro peer?        leafref
      |  |     |     +--ro fec-label* [fec]
      |  |     |        +--ro fec     inet:ipv4-prefix
      |  |     |        +--ro peer* [peer advertisement-type]
      |  |     |           +--ro peer                    leafref
      |  |     |           +--ro advertisement-type    advertised-received
      |  |     |           +--ro label?                mpls:mpls-label
      |  |     |           +--ro used-in-forwarding?   boolean
      |  |     +--ro ipv6
      |  |        +--ro transport-address?   inet:ipv6-address
      |  |        +--ro binding
      |  |           +--ro address* [address]
      |  |           |  +--ro address      inet:ipv6-address
      |  |           |  +--ro advertisement-type?   advertised-received
```

```
      |  |           |  +--ro peer?        leafref
      |  |           +--ro fec-label* [fec]
      |  |              +--ro fec     inet:ipv6-prefix
      |  |              +--ro peer* [peer advertisement-type]
      |  |                 +--ro peer                  leafref
      |  |                 +--ro advertisement-type    advertised-received
      |  |                 +--ro label?                mpls:mpls-label
      |  |                 +--ro used-in-forwarding?   boolean
      |  +--rw discovery
      |  |  +--rw interfaces
      |  |  |  +--ro state
      |  |  |  |  +--ro hello-holdtime?   uint16
      |  |  |  |  +--ro hello-interval?   uint16
      |  |  |  +--rw interface* [interface]
      |  |  |     +--ro state
      |  |  |     |  +--ro hello-holdtime?              uint16
      |  |  |     |  +--ro hello-interval?              uint16
      |  |  |     |  +--ro igp-synchronization-delay?   uint16 {per-interface-ti
mer-config}?
      |  |  |     |  +--ro next-hello?                  uint16
      |  |  |     +--rw address-family* [afi]
      |  |  |        +--rw afi       ldp-address-family
      |  |  |        +--ro state
      |  |  |           +--ro enable?   boolean
      |  |  |           +--ro ipv4
      |  |  |           |  +--ro transport-address?   union
      |  |  |           |  +--ro hello-adjacencies* [adjacent-address]
      |  |  |           |     +--ro adjacent-address    inet:ipv4-address
      |  |  |           |     +--ro flag*               identityref
      |  |  |           |     +--ro hello-holdtime
      |  |  |           |     |  +--ro adjacent?     uint16
      |  |  |           |     |  +--ro negotiated?   uint16
      |  |  |           |     |  +--ro remaining?    uint16
      |  |  |           |     +--ro next-hello?         uint16
      |  |  |           |     +--ro statistics
      |  |  |           |     |  +--ro discontinuity-time    yang:date-and-time
      |  |  |           |     |  +--ro hello-received?       yang:counter64
      |  |  |           |     |  +--ro hello-dropped?        yang:counter64
      |  |  |           |     +--ro peer?               leafref
      |  |  |           +--ro ipv6
      |  |  |              +--ro transport-address?   union
      |  |  |              +--ro hello-adjacencies* [adjacent-address]
      |  |  |                 +--ro adjacent-address    inet:ipv6-address
      |  |  |                 +--ro flag*               identityref
      |  |  |                 +--ro hello-holdtime
      |  |  |                 |  +--ro adjacent?     uint16
      |  |  |                 |  +--ro negotiated?   uint16
      |  |  |                 |  +--ro remaining?    uint16
      |  |  |                 +--ro next-hello?         uint16
```

```
      |  |  |                 +--ro statistics
      |  |  |                 |  +--ro discontinuity-time    yang:date-and-time
      |  |  |                 |  +--ro hello-received?       yang:counter64
      |  |  |                 |  +--ro hello-dropped?        yang:counter64
      |  |  |                 +--ro peer?               leafref
      |  |  +--rw targeted
      |  |     +--ro state
      |  |     |  +--ro hello-holdtime?   uint16
      |  |     |  +--ro hello-interval?   uint16
      |  |     |  +--ro hello-accept {policy-extended-discovery-config}?
      |  |     |     +--ro enable?           boolean
      |  |     |     +--ro neighbor-list?   neighbor-list-ref
      |  |     +--rw address-family* [afi]
      |  |        +--rw afi      ldp-address-family
      |  |        +--ro state
      |  |        |  +--ro ipv4
      |  |        |  |  +--ro hello-adjacencies* [local-address adjacent-address
]
      |  |        |  |     +--ro local-address       inet:ipv4-address
      |  |        |  |     +--ro adjacent-address    inet:ipv4-address
      |  |        |  |     +--ro flag*               identityref
      |  |        |  |     +--ro hello-holdtime
      |  |        |  |     |  +--ro adjacent?     uint16
      |  |        |  |     |  +--ro negotiated?   uint16
      |  |        |  |     |  +--ro remaining?    uint16
      |  |        |  |     +--ro next-hello?         uint16
      |  |        |  |     +--ro statistics
      |  |        |  |     |  +--ro discontinuity-time    yang:date-and-time
      |  |        |  |     |  +--ro hello-received?       yang:counter64
      |  |        |  |     |  +--ro hello-dropped?        yang:counter64
      |  |        |  |     +--ro peer?               leafref
      |  |        |  +--ro ipv6
      |  |        |     +--ro hello-adjacencies* [local-address adjacent-address
]
      |  |        |        +--ro local-address       inet:ipv6-address
      |  |        |        +--ro adjacent-address    inet:ipv6-address
      |  |        |        +--ro flag*               identityref
      |  |        |        +--ro hello-holdtime
      |  |        |        |  +--ro adjacent?     uint16
      |  |        |        |  +--ro negotiated?   uint16
      |  |        |        |  +--ro remaining?    uint16
      |  |        |        +--ro next-hello?         uint16
      |  |        |        +--ro statistics
      |  |        |        |  +--ro discontinuity-time    yang:date-and-time
      |  |        |        |  +--ro hello-received?       yang:counter64
      |  |        |        |  +--ro hello-dropped?        yang:counter64
      |  |        |        +--ro peer?               leafref
      |  |        +--rw ipv4
      |  |        |  +--rw target* [adjacent-address]
      |  |        |     +--rw adjacent-address    inet:ipv4-address
```

```
      |  |        |     +--ro state
      |  |        |        +--ro enable?          boolean
      |  |        |        +--ro local-address?   inet:ipv4-address
      |  |        +--rw ipv6
      |  |           +--rw target* [adjacent-address]
      |  |              +--rw adjacent-address    inet:ipv6-address
      |  |              +--ro state
      |  |                 +--ro enable?          boolean
      |  |                 +--ro local-address?   inet:ipv6-address
      |  +--rw forwarding-nexthop {forwarding-nexthop-config}?
      |  |  +--rw interfaces
      |  |     +--rw interface* [interface]
      |  |        +--rw interface         mpls-interface-ref
      |  |        +--rw address-family* [afi]
      |  |           +--rw afi       ldp-address-family
      |  |           +--ro state
      |  |              +--ro ldp-disable?    boolean
      +--rw peers
         +--ro state
         |  +--ro session-authentication-md5-password?   string
         |  +--ro session-ka-holdtime?                   uint16
         |  +--ro session-ka-interval?                   uint16
         |  +--ro session-downstream-on-demand {session-downstream-on-demand-con
fig}?
         |     +--ro enable?      boolean
         |     +--ro peer-list?   peer-list-ref
         +--rw peer* [lsr-id]
            +--rw lsr-id    yang:dotted-quad
            +--ro state
               +--ro admin-down?                            boolean
               +--ro capability
               +--ro label-policy
               |  +--ro advertise
               |  |  +--ro prefix-list?   prefix-list-ref
               |  +--ro accept
               |     +--ro prefix-list?   prefix-list-ref
               +--ro session-authentication-md5-password?   string
               +--ro graceful-restart
               |  +--ro enable?           boolean
               |  +--ro reconnect-time?   uint16
               |  +--ro recovery-time?    uint16
               +--ro session-ka-holdtime?                   uint16
               +--ro session-ka-interval?                   uint16
               +--ro address-family
               |  +--ro ipv4
               |  |  +--ro label-policy
               |  |  |  +--ro advertise
               |  |  |  |  +--ro prefix-list?   prefix-list-ref
               |  |  |  +--ro accept
```

```
|  |  |     +--ro prefix-list?   prefix-list-ref
|  |  +--ro hello-adjacencies* [local-address adjacent-address]
|  |     +--ro local-address       inet:ipv4-address
|  |     +--ro adjacent-address    inet:ipv4-address
|  |     +--ro flag*               identityref
|  |     +--ro hello-holdtime
|  |     |  +--ro adjacent?     uint16
|  |     |  +--ro negotiated?   uint16
|  |     |  +--ro remaining?    uint16
|  |     +--ro next-hello?         uint16
|  |     +--ro statistics
|  |     |  +--ro discontinuity-time    yang:date-and-time
|  |     |  +--ro hello-received?       yang:counter64
|  |     |  +--ro hello-dropped?        yang:counter64
|  |     +--ro interface?          mpls-interface-ref
|  +--ro ipv6
|     +--ro label-policy
|     |  +--ro advertise
|     |  |  +--ro prefix-list?   prefix-list-ref
|     |  +--ro accept
|     |     +--ro prefix-list?   prefix-list-ref
|     +--ro hello-adjacencies* [local-address adjacent-address]
|        +--ro local-address       inet:ipv6-address
|        +--ro adjacent-address    inet:ipv6-address
|        +--ro flag*               identityref
|        +--ro hello-holdtime
|        |  +--ro adjacent?     uint16
|        |  +--ro negotiated?   uint16
|        |  +--ro remaining?    uint16
|        +--ro next-hello?         uint16
|        +--ro statistics
|        |  +--ro discontinuity-time    yang:date-and-time
|        |  +--ro hello-received?       yang:counter64
|        |  +--ro hello-dropped?        yang:counter64
|        +--ro interface?          mpls-interface-ref
+--ro label-advertisement-mode
|  +--ro local?        label-adv-mode
|  +--ro peer?         label-adv-mode
|  +--ro negotiated?   label-adv-mode
+--ro next-keep-alive?                           uint16
+--ro peer-ldp-id?                               yang:dotted-quad
+--ro received-peer-state
|  +--ro graceful-restart
|  |  +--ro enable?           boolean
|  |  +--ro reconnect-time?   uint16
|  |  +--ro recovery-time?    uint16
|  +--ro capability
|     +--ro end-of-lib
```

```
|     |  +--ro enable?   boolean
|     +--ro typed-wildcard-fec
|     |  +--ro enable?   boolean
|     +--ro upstream-label-assignment
|        +--ro enable?   boolean
+--ro session-holdtime
|  +--ro peer?         uint16
|  +--ro negotiated?   uint16
|  +--ro remaining?    uint16
+--ro session-state?                           enumeration
+--ro tcp-connection
|  +--ro local-address?    inet:ip-address
|  +--ro local-port?       inet:port-number
|  +--ro remote-address?   inet:ip-address
|  +--ro remote-port?      inet:port-number
+--ro up-time?                                 string
+--ro statistics
   +--ro discontinuity-time          yang:date-and-time
   +--ro received
   |  +--ro total-octets?          yang:counter64
   |  +--ro total-messages?        yang:counter64
   |  +--ro address?               yang:counter64
   |  +--ro address-withdraw?      yang:counter64
   |  +--ro initialization?        yang:counter64
   |  +--ro keepalive?             yang:counter64
   |  +--ro label-abort-request?   yang:counter64
   |  +--ro label-mapping?         yang:counter64
   |  +--ro label-release?         yang:counter64
   |  +--ro label-request?         yang:counter64
   |  +--ro label-withdraw?        yang:counter64
   |  +--ro notification?          yang:counter64
   +--ro sent
   |  +--ro total-octets?          yang:counter64
   |  +--ro total-messages?        yang:counter64
   |  +--ro address?               yang:counter64
   |  +--ro address-withdraw?      yang:counter64
   |  +--ro initialization?        yang:counter64
   |  +--ro keepalive?             yang:counter64
   |  +--ro label-abort-request?   yang:counter64
   |  +--ro label-mapping?         yang:counter64
   |  +--ro label-release?         yang:counter64
   |  +--ro label-request?         yang:counter64
   |  +--ro label-withdraw?        yang:counter64
   |  +--ro notification?          yang:counter64
   +--ro total-addresses?            uint32
   +--ro total-labels?               uint32
   +--ro total-fec-label-bindings?   uint32
```

Figure 5

3.3.1. Derived States

Following are main areas for which LDP operational "derived" state is defined:

Neighbor Adjacencies

Peer

Bindings (FEC-label and address)

Capabilities

3.3.1.1. Adjacency state

Neighbor adjacencies are per address-family hello adjacencies that are formed with neighbors as result of LDP basic or extended discovery. In terms of organization, there is a source of discovery (e.g. interface or target address) along with its associated parameters and one or more discovered neighbors along with neighbor discovery related parameters. For the basic discovery, there could be more than one discovered neighbor for a given source (interface), whereas there is at most one discovered neighbor for an extended discovery source (local-address and target-address). This is also to be noted that the reason for a targeted neighbor adjacency could be either an active source (locally configured targeted) or passive source (to allow any incoming extended/targeted hellos). A neighbor/adjacency record also contains session-state that helps highlight whether a given adjacency has progressed to subsequent session level or to eventual peer level.

Following captures high level tree hierarchy for neighbor adjacency state.

```
   +--rw mpls-ldp!
      +--rw discovery
         +--rw interfaces
         |  +--rw interface* [interface]
         |     +--rw address-family* [af]
         |        +--ro state
         |           +--ro ipv4 (or ipv6)
         |              +--ro hello-adjacencies* [adjacent-address]
         |                 +--ro adjacent-address
         |                    . . . .
         |                    . . . .
         +--rw targeted
            +--rw address-family* [afi]
                +--rw afi      address-family
                   +--ro state
                      +--ro ipv4 (or ipv6)
                         +--ro hello-adjacencies* [local-address adjacent-addres
s]
                            +--ro local-address
                            +--ro adjacent-address
                               . . . .
                               . . . .
```

Figure 6

3.3.1.2. Peer state

Peer related derived state is presented under peers tree. This is one of the core state that provides info on the session related parameters (mode, authentication, KA timeout etc.), TCP connection info, hello adjacencies for the peer, statistics related to messages and bindings, and capabilities exchange info.

Following captures high level tree hierarchy for peer state.

```
  +--rw mpls-ldp!
      +--rw peers
         +--rw peer* [lsr-id]
            +--rw lsr-id
            +--ro state
               +--ro session-ka-holdtime?
               +-- . . . .
               +-- . . . .
               +--ro capability
               +  +ro -- . . .
               +--ro address-family
               |  +--ro ipv4 (or ipv6)
               |     +--ro hello-adjacencies* [local-address adjacent-address]
               |        . . . .
               |        . . . .
               +--ro received-peer-state
               |  +--ro . . . .
               |  +--ro capability
               |     +--ro . . . .
               +--ro statistics
                  +-- . . . .
                  +-- . . . .
```

Figure 7

3.3.1.3. Bindings state

Binding state provides information on LDP FEC-label bindings as well as address binding for both inbound (received) as well as outbound (advertised) direction. FEC-label bindings are presented as a FEC-centric view, and address bindings are presented as an address-centric view:

```
        FEC-Label bindings:
            FEC 200.1.1.1/32:
              advertised: local-label 16000
                peer 192.168.0.2:0
                peer 192.168.0.3:0
                peer 192.168.0.4:0
              received:
                peer 192.168.0.2:0, label 16002, used-in-forwarding=Yes
                peer 192.168.0.3:0, label 17002, used-in-forwarding=No
            FEC 200.1.1.2/32:
               . . . .
            FEC 201.1.0.0/16:
               . . . .


        Address bindings:
            Addr 1.1.1.1:
              advertised
            Addr 1.1.1.2:
              advertised
            Addr 2.2.2.2:
              received, peer 192.168.0.2
            Addr 2.2.2.22:
              received, peer 192.168.0.2
            Addr 3.3.3.3:
              received, peer 192.168.0.3
            Addr 3.3.3.33:
              received, peer 192.168.0.3
```

Figure 8

Note that all local addresses are advertised to all peers and hence no need to provide per-peer information for local address advertisement. Furthermore, note that it is easy to derive a peer-centric view for the bindings from the information already provided in this model.

Following captures high level tree hierarchy for bindings state.

```
   +--rw mpls-ldp!
      +--rw global
         +--rw address-family* [afi]
            +--rw afi       address-family
            +--ro state
               +--ro ipv4 (or ipv6)
                  +--ro bindings
                     +--ro address* [address]
                     |  +--ro address
                     |  +--ro direction?   advertised-received
                     |  +--ro peer?        leafref
                     +--ro fec-label* [fec]
                        +--ro fec     inet:ipv4-prefix
                        +--ro peer* [peer advertisement-type]
                           +--ro peer                  leafref
                           +--ro advertisement-type    advertised-received
                           +--ro label?                mpls:mpls-label
                           +--ro used-in-forwarding?   boolean
```

Figure 9

3.3.1.4. Capabilities state

LDP capabilities state comprise two types of information - global information (such as timer etc.), and per-peer information.

Following captures high level tree hierarchy for LDP capabilities state.

```
      +--rw mpls-ldp!
         +--rw global
         |  +--ro state
         |     +--ro capability
         |        +--ro . . . .
         |        +--ro . . . .
         +--rw peers
            +--rw peer* [lsr-id]
               +--rw lsr-id    yang:dotted-quad
               +--ro state
                  +--ro received-peer-state
                     +--ro capability
                        +--ro . . . .
                        +--ro . . . .
```

Figure 10

3.4. Notifications

This model defines a list of notifications to inform client of important events detected during the protocol operation. These events include events related to changes in the operational state of an LDP peer, hello adjacency, and FEC etc. It is to be noted that an LDP FEC is treated as operational (up) as long as it has at least 1 NHLFE with outgoing label.

Following is a simplified graphical representation of the data model for LDP notifications.

```
module: ietf-mpls-ldp
notifications:
   +---n mpls-ldp-peer-event
   |  +--ro event-type?   oper-status-event-type
   |  +--ro peer-ref?     leafref
   +---n mpls-ldp-hello-adjacency-event
   |  +--ro event-type?   oper-status-event-type
   |  +--ro (hello-adjacency-type)?
   |     +--:(targeted)
   |     |  +--ro targeted
   |     |     +--ro target-address?   inet:ip-address
   |     +--:(link)
   |        +--ro link
   |           +--ro next-hop-interface?   mpls-interface-ref
   |           +--ro next-hop-address?     inet:ip-address
   +---n mpls-ldp-fec-event
      +--ro event-type?   oper-status-event-type
      +--ro prefix?       inet:ip-prefix
```

Figure 11

3.5. Actions

This model defines a list of rpcs that allow performing an action or executing a command on the protocol. For example, it allows to clear (reset) LDP peers, hello-adjacencies, and statistics. The model makes an effort to provide different level of control so that a user is able to either clear all, or clear all for a given type, or clear a specific entity.

Following is a simplified graphical representation of the data model for LDP actions.

```
module: ietf-mpls-ldp
rpcs:
   +---x mpls-ldp-clear-peer
   |  +---w input
   |     +---w lsr-id?   union
   +---x mpls-ldp-clear-hello-adjacency
   |  +---w input
   |     +---w hello-adjacency
   |        +---w (hello-adjacency-type)?
   |           +--:(targeted)
   |           |  +---w targeted!
   |           |     +---w target-address?   inet:ip-address
   |           +--:(link)
   |              +---w link!
   |                 +---w next-hop-interface?   mpls-interface-ref
   |                 +---w next-hop-address?     inet:ip-address
   +---x mpls-ldp-clear-peer-statistics
      +---w input
         +---w lsr-id?   union
```

Figure 12

4. mLDP YANG Model

4.1. Overview

Due to tight dependency of mLDP on LDP, mLDP model builds on top of LDP model defined earlier in the document. Following are the main mLDP areas and documents that are within the scope of this model:

- o mLDP Base Specification [RFC6388]
- o mLDP Recursive FEC [RFC6512]
- o Targeted mLDP [RFC7060]
- o mLDP Fast-Reroute (FRR)
  - * Node Protection [RFC7715]
  - * Multicast-only
- o Hub-and-Spoke Multipoint LSPs [RFC7140]
- o mLDP In-band Signaling [RFC6826] (future revision)
- o mLDP In-band signaling in a VRF [RFC7246]

- o mLDP In-band Signaling with Wildcards [RFC7438] (future revision)
- o Configured Leaf LSPs (manually provisioned)

[Ed Note: Some of the topics in the above list are to be addressed/ added in later revision of this document].

4.2. Configuration

4.2.1. Configuration Hierarchy

In terms of overall configuration layout, following figure highlights extensions to LDP configuration model to incorporate mLDP:

```
      +-- mpls-ldp
         +-- ...
         +-- ...
         +-- mldp
         |  +-- ...
         |  +-- ...
         |  +-- address-family* [af]
         |     +-- af
         |        +-- ...
         |        +-- ...
         +-- global
         |  +-- ...
         |  +-- capability
         |     +-- ...
         |     +-- ...
         |     +-- mldp
         |        +-- ...
         |        +-- ...
         +-- discovery
         |  +-- ...
         |  +-- ...
         +-- forwarding-nexthop
         |  +-- interfaces
         |     +-- interface* [interface]
         |        +-- interface
         |        +-- address-family* [af]
         |           +-- af
         |           +-- ...
         |           +-- mldp-disable
         +-- peers
            +-- ...
            +-- ...
            +-- peer* [lsr-id]
                +-- ...
                +-- ...
                +-- capability
                    +-- ...
                    +-- ...
                    +-- mldp
                        +-- ...
                        +-- ...
```

Figure 13

From above hierarchy, we can categorize mLDP configuration parameters into two types:

- o Parameters that leverage/extend LDP containers and parameters

- o Parameters that are mLDP specific

Following subsections first describe mLDP specific configuration parameters, followed by those leveraging LDP.

4.2.2. mldp container

mldp container resides directly under "mpls-ldp" and holds the configuration related to items that are mLDP specific. The main items under this container are:

- o mLDP enabling: To enable mLDP under a (VRF) routing instance, mldp container is enabled under LDP. Given that mLDP requires LDP signalling, it is not sensible to allow disabling LDP control plane under a (VRF) network-instance while requiring mLDP to be enabled for the same. However, if a user wishes only to allow signalling for multipoint FECs on an LDP/mLDP enabled VRF instance, he/she can use LDP label-policies to disable unicast FECs under the VRF.

- o mLDP per-AF features: mLDP manages its own list of IP address-families and the features enabled underneath. The per-AF mLDP configuration items include:

  - * Multicast-only FRR: This enables Multicast-only FRR functionality for a given AF under mLDP. The feature allows route-policy to be configured for finer control/applicability of the feature.

  - * Recursive FEC: The recursive-fec feature [RFC6512] can be enabled per AF with a route-policy.

  - * Configured Leaf LSPs: To provision multipoint leaf LSP manually, a container is provided per-AF under LDP. The configuration is flexible and allows a user to specify MP LSPs of type p2mp or mp2mp with IPv4 or IPv6 root address(es) by using either LSP-Id or (S,G).

Targeted mLDP feature specification [RFC7060] do not require any mLDP specific configuration. It, however, requires LDP upstream-label-assignment capability [RFC6389] to be enabled.

4.2.3. Leveraging LDP containers

mLDP configuration model leverages following configuration areas and containers that are already defined for LDP:

- Capabilities: A new container "mldp" is defined under Capabilities container. This new container specifies any mLDP specific capabilities and their parameters. Moreover, a new "mldp" container is also added under per-peer capability container to override/control mLDP specific capabilities on a peer level. In the scope of this document, the most important capabilities related to mLDP are p2mp, mp2mp, make-before-break, hub-and-spoke, and node-protection.

- Discovery and Peer: mLDP requires LDP discovery and peer procedures to form mLDP peering. A peer is treated as mLDP peer only when either P2MP or MP2MP capabilities have been successfully exchanged with the peer. If a user wish to selectively enable or disable mLDP with a LDP-enabled peer, he/she may use per-peer mLDP capabilities configuration. [Ed Note: The option to control mLDP enabling/disabling on a peer-list is being explored for future]. In most common deployments, it is desirable to disable mLDP (capabilities announcements) on a targeted-only LDP peering, where targeted-only peer is the one whose discovery sources are targeted only. In future revision, a configuration option for this support will also be provided.

- Forwarding: By default, mLDP is allowed to select any of the LDP enabled interface as a downstream interface towards a nexthop (LDP/mLDP peer) for MP LSP programming. However, a configuration option is provided to allow mLDP to exclude a given interface from such a selection. Note that such a configuration option will be useful only when there are more than one interfaces available for the downstream selection.

This goes without saying that mLDP configuration tree follows the same approach as LDP, where the tree comprise leafs for intended configuration.

4.2.4. YANG tree

The following figure captures the YANG tree for mLDP configuration. To keep the focus, the figure has been simplified to display only mLDP items without any LDP items.

```
module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
   +--rw mpls-ldp!
```

```
      +--rw global
      |  +--rw config
      |  |  +--rw capability
      |  |     +--rw mldp {mldp}?
      |  |        +--rw p2mp
      |  |        |  +--rw enable?   boolean
      |  |        +--rw mp2mp
      |  |        |  +--rw enable?   boolean
      |  |        +--rw make-before-break
      |  |        |  +--rw enable?             boolean
      |  |        |  +--rw switchover-delay?   uint16
      |  |        |  +--rw timeout?            uint16
      |  |        +--rw hub-and-spoke {capability-mldp-hsmp}?
      |  |        |  +--rw enable?   boolean
      |  |        +--rw node-protection {capability-mldp-node-protection}?
      |  |           +--rw plr?           boolean
      |  |           +--rw merge-point
      |  |              +--rw enable?                              boolean
      |  |              +--rw targeted-session-teardown-delay?   uint16
      |  +--rw mldp {mldp}?
      |  |  +--rw config
      |  |  |  +--rw enable?   boolean
      |  |  +--rw address-family* [afi]
      |  |     +--rw afi                     ldp-address-family
      |  |     +--rw config
      |  |     |  +--rw multicast-only-frr {mldp-mofrr}?
      |  |     |  |  +--rw prefix-list?   prefix-list-ref
      |  |     |  +--rw recursive-fec
      |  |     |     +--rw prefix-list?   prefix-list-ref
      |  |     +--rw configured-leaf-lsps
      |  |        +--rw p2mp
      |  |        |  +--rw roots-ipv4
      |  |        |  |  +--rw root* [root-address]
      |  |        |  |     +--rw root-address    inet:ipv4-address
      |  |        |  |     +--rw lsp* [lsp-id source-address group-address]
      |  |        |  |        +--rw lsp-id            uint16
      |  |        |  |        +--rw source-address    inet:ipv4-address
      |  |        |  |        +--rw group-address     inet:ipv4-address-no-zone
      |  |        |  +--rw roots-ipv6
      |  |        |     +--rw root* [root-address]
      |  |        |        +--rw root-address    inet:ipv6-address
      |  |        |        +--rw lsp* [lsp-id source-address group-address]
      |  |        |           +--rw lsp-id            uint16
      |  |        |           +--rw source-address    inet:ipv6-address
      |  |        |           +--rw group-address     inet:ipv6-address-no-zone
      |  |        +--rw mp2mp
      |  |           +--rw roots-ipv4
      |  |           |  +--rw root* [root-address]
```

```
      |  |           |     +--rw root-address    inet:ipv4-address
      |  |           |     +--rw lsp* [lsp-id source-address group-address]
      |  |           |        +--rw lsp-id           uint16
      |  |           |        +--rw source-address    inet:ipv4-address
      |  |           |        +--rw group-address     inet:ipv4-address-no-zone
      |  |           +--rw roots-ipv6
      |  |              +--rw root* [root-address]
      |  |                 +--rw root-address    inet:ipv6-address
      |  |                 +--rw lsp* [lsp-id source-address group-address]
      |  |                    +--rw lsp-id           uint16
      |  |                    +--rw source-address    inet:ipv6-address
      |  |                    +--rw group-address     inet:ipv6-address-no-zone
      |  +--rw forwarding-nexthop {forwarding-nexthop-config}?
      |     +--rw interfaces
      |        +--rw interface* [interface]
      |           +--rw interface         if:interface-ref
      |           +--rw address-family* [afi]
      |              +--rw afi       address-family
      |              +--rw config
      |                 +--rw mldp-disable?   boolean {mldp}?
      +--rw peers
         +--rw peer* [lsr-id]
               +--rw capability
                  +--rw mldp {mldp}?
                     +--rw p2mp
                     |  +--rw enable?   boolean
                     +--rw mp2mp
                     |  +--rw enable?   boolean
                     +--rw make-before-break
                     |  +--rw enable?             boolean
                     |  +--rw switchover-delay?   uint16
                     |  +--rw timeout?            uint16
                     +--rw hub-and-spoke {capability-mldp-hsmp}?
                     |  +--rw enable?   boolean
                      +--rw node-protection {capability-mldp-node-protection}?
                        +--rw plr?          boolean
                        +--rw merge-point
                           +--rw enable?                            boolean
                           +--rw targeted-session-teardown-delay?   uint16
```

Figure 14

4.3. Operational State

Operational state of mLDP can be queried and obtained from this read-only container "mldp" which resides under mpls-ldp container.

Please note this state tree refers both the configuration "applied" state as well as the "derived" state related to the mLDP protocol.

Following is a simplified graphical representation of the data model for mLDP operational state:

```
module: ietf-mpls-ldp
augment /rt:routing/rt:control-plane-protocols:
   +--rw mpls-ldp!
      +--rw global
      |  +--ro state
      |  |  +--ro capability
      |  |     +--ro mldp {mldp}?
      |  |        +--ro p2mp
      |  |        |  +--ro enable?   boolean
      |  |        +--ro mp2mp
      |  |        |  +--ro enable?   boolean
      |  |        +--ro make-before-break
      |  |        |  +--ro enable?             boolean
      |  |        |  +--ro switchover-delay?   uint16
      |  |        |  +--ro timeout?            uint16
      |  |        +--ro hub-and-spoke {capability-mldp-hsmp}?
      |  |        |  +--ro enable?   boolean
      |  |        +--ro node-protection {capability-mldp-node-protection}?
      |  |           +--ro plr?           boolean
      |  |           +--ro merge-point
      |  |              +--ro enable?                              boolean
      |  |              +--ro targeted-session-teardown-delay?   uint16
      |  |
      |  +--rw mldp {mldp}?
      |  |  +--ro state
      |  |     +--ro enable?   boolean
      |  |  +--rw address-family* [afi]
      |  |     +--rw afi                      ldp-address-family
      |  |     +--ro state
      |  |        +--ro multicast-only-frr {mldp-mofrr}?
      |  |        |  +--ro prefix-list?   prefix-list-ref
      |  |        +--ro recursive-fec
      |  |        |  +--ro prefix-list?   prefix-list-ref
      |  |        +--ro ipv4
      |  |        |  +--ro roots
      |  |        |  |  +--ro root* [root-address]
      |  |        |  |     +--ro root-address    inet:ipv4-address
      |  |        |  |     +--ro is-self?        boolean
      |  |        |  |     +--ro reachability* [address interface]
      |  |        |  |        +--ro address      inet:ipv4-address
      |  |        |  |        +--ro interface    mpls-interface-ref
```

```
      |  |        |  |        +--ro peer?        leafref
      |  |        |  +--ro bindings
      |  |        |     +--ro opaque-type-lspid
      |  |        |     |  +--ro fec-label* [root-address lsp-id recur-root-addr
ess recur-rd]
      |  |        |     |     +--ro root-address          inet:ipv4-address
      |  |        |     |     +--ro lsp-id                uint32
      |  |        |     |     +--ro recur-root-address    inet:ip-address
      |  |        |     |     +--ro recur-rd              route-distinguisher
      |  |        |     |     +--ro multipoint-type?      multipoint-type
      |  |        |     |     +--ro peer* [direction peer advertisement-type]
      |  |        |     |        +--ro direction             downstream-upstream
      |  |        |     |        +--ro peer                  leafref
      |  |        |     |        +--ro advertisement-type    advertised-received
      |  |        |     |        +--ro label?                mpls:mpls-label
      |  |        |     |        +--ro mbb-role?             enumeration
      |  |        |     |        +--ro mofrr-role?           enumeration
      |  |        |     +--ro opaque-type-src
      |  |        |     |  +--ro fec-label* [root-address source-address group-a
ddress rd recur-root-address recur-rd]
      |  |        |     |     +--ro root-address          inet:ipv4-address
      |  |        |     |     +--ro source-address        inet:ip-address
      |  |        |     |     +--ro group-address         inet:ip-address-no-zon
e
      |  |        |     |     +--ro rd                    route-distinguisher
      |  |        |     |     +--ro recur-root-address    inet:ip-address
      |  |        |     |     +--ro recur-rd              route-distinguisher
      |  |        |     |     +--ro multipoint-type?      multipoint-type
      |  |        |     |     +--ro peer* [direction peer advertisement-type]
      |  |        |     |        +--ro direction             downstream-upstream
      |  |        |     |        +--ro peer                  leafref
      |  |        |     |        +--ro advertisement-type    advertised-received
      |  |        |     |        +--ro label?                mpls:mpls-label
      |  |        |     |        +--ro mbb-role?             enumeration
      |  |        |     |        +--ro mofrr-role?           enumeration
      |  |        |     +--ro opaque-type-bidir
      |  |        |        +--ro fec-label* [root-address rp group-address rd re
cur-root-address recur-rd]
      |  |        |           +--ro root-address          inet:ipv4-address
      |  |        |           +--ro rp                    inet:ip-address
      |  |        |           +--ro group-address         inet:ip-address-no-zon
e
      |  |        |           +--ro rd                    route-distinguisher
      |  |        |           +--ro recur-root-address    inet:ip-address
      |  |        |           +--ro recur-rd              route-distinguisher
      |  |        |           +--ro multipoint-type?      multipoint-type
      |  |        |           +--ro peer* [direction peer advertisement-type]
      |  |        |              +--ro direction             downstream-upstream
      |  |        |              +--ro peer                  leafref
      |  |        |              +--ro advertisement-type    advertised-received
      |  |        |              +--ro label?                mpls:mpls-label
      |  |        |              +--ro mbb-role?             enumeration
      |  |        |              +--ro mofrr-role?           enumeration
```

```
      |  |        +--ro ipv6
      |  |           +--ro roots
      |  |           |  +--ro root* [root-address]
      |  |           |     +--ro root-address    inet:ipv6-address
      |  |           |     +--ro is-self?        boolean
      |  |           |     +--ro reachability* [address interface]
      |  |           |        +--ro address      inet:ipv6-address
      |  |           |        +--ro interface    mpls-interface-ref
      |  |           |        +--ro peer?        leafref
      |  |           +--ro bindings
      |  |              +--ro opaque-type-lspid
      |  |              |  +--ro fec-label* [root-address lsp-id recur-root-addr
ess recur-rd]
      |  |              |     +--ro root-address          inet:ipv6-address
      |  |              |     +--ro lsp-id                uint32
      |  |              |     +--ro recur-root-address    inet:ip-address
      |  |              |     +--ro recur-rd              route-distinguisher
      |  |              |     +--ro multipoint-type?      multipoint-type
      |  |              |     +--ro peer* [direction peer advertisement-type]
      |  |              |        +--ro direction             downstream-upstream
      |  |              |        +--ro peer                  leafref
      |  |              |        +--ro advertisement-type    advertised-received
      |  |              |        +--ro label?                mpls:mpls-label
      |  |              |        +--ro mbb-role?             enumeration
      |  |              |        +--ro mofrr-role?           enumeration
      |  |              +--ro opaque-type-src
      |  |              |  +--ro fec-label* [root-address source-address group-a
ddress rd recur-root-address recur-rd]
      |  |              |     +--ro root-address          inet:ipv6-address
      |  |              |     +--ro source-address        inet:ip-address
      |  |              |     +--ro group-address         inet:ip-address-no-zon
e
      |  |              |     +--ro rd                    route-distinguisher
      |  |              |     +--ro recur-root-address    inet:ip-address
      |  |              |     +--ro recur-rd              route-distinguisher
      |  |              |     +--ro multipoint-type?      multipoint-type
      |  |              |     +--ro peer* [direction peer advertisement-type]
      |  |              |        +--ro direction             downstream-upstream
      |  |              |        +--ro peer                  leafref
      |  |              |        +--ro advertisement-type    advertised-received
      |  |              |        +--ro label?                mpls:mpls-label
      |  |              |        +--ro mbb-role?             enumeration
      |  |              |        +--ro mofrr-role?           enumeration
      |  |              +--ro opaque-type-bidir
      |  |                 +--ro fec-label* [root-address rp group-address rd re
cur-root-address recur-rd]
      |  |                    +--ro root-address          inet:ipv6-address
      |  |                    +--ro rp                    inet:ip-address
      |  |                    +--ro group-address         inet:ip-address-no-zon
e
      |  |                    +--ro rd                    route-distinguisher
      |  |                    +--ro recur-root-address    inet:ip-address
      |  |                    +--ro recur-rd              route-distinguisher
```

```
      |  |                    +--ro multipoint-type?      multipoint-type
      |  |                    +--ro peer* [direction peer advertisement-type]
      |  |                       +--ro direction             downstream-upstream
      |  |                       +--ro peer                  leafref
      |  |                       +--ro advertisement-type    advertised-received
      |  |                       +--ro label?                mpls:mpls-label
      |  |                       +--ro mbb-role?             enumeration
      |  |                       +--ro mofrr-role?           enumeration
      |  +--rw forwarding-nexthop {forwarding-nexthop-config}?
      |     +--rw interfaces
      |        +--rw interface* [interface]
      |           +--rw address-family* [afi]
      |              +--ro state
      |                 +--ro mldp-disable?   boolean {mldp}?
      +--rw peers
         +--rw peer* [lsr-id]
            +--ro state
               +--ro capability
               |   +--ro mldp {mldp}?
               |     +--ro p2mp
               |     |  +--ro enable?   boolean
               |     +--ro mp2mp
               |     |  +--ro enable?   boolean
               |     +--ro make-before-break
               |     |  +--ro enable?             boolean
               |     |  +--ro switchover-delay?   uint16
               |     |  +--ro timeout?            uint16
               |     +--ro hub-and-spoke {capability-mldp-hsmp}?
               |     |  +--ro enable?   boolean
               |     +--ro node-protection {capability-mldp-node-protection}?
               |        +--ro plr?           boolean
               |        +--ro merge-point
               |           +--ro enable?                              boolean
               |           +--ro targeted-session-teardown-delay?   uint16
               +--ro received-peer-state
                  +--ro capability
                     +--ro mldp {mldp}?
                        +--ro p2mp
                        |  +--ro enable?   boolean
                        +--ro mp2mp
                        |  +--ro enable?   boolean
                        +--ro make-before-break
                        |  +--ro enable?   boolean
                        +--ro hub-and-spoke
                        |  +--ro enable?   boolean
                        +--ro node-protection
                           +--ro plr?           boolean
                           +--ro merge-point?   boolean
```

Figure 15

4.3.1. Derived states

Following are main areas for which mLDP operational derived state is defined:

- o Root
- o Bindings (FEC-label)
- o Capabilities

4.3.1.1. Root state

Root address is a fundamental construct for MP FEC bindings and LSPs. The root state provides information on all the known roots in a given address-familty, and their information on the root reachability (as learnt from RIB). In case of multi-path reachability to a root, the selection of upstream path is done on per-LSP basis at the time of LSP setup. Similarly, when protection mechanisms like MBB or MoFRR are in place, the path designation as active/standby or primary/ backup is also done on per LSP basis. It is to be noted that a given root can be shared amongst multiple P2MP and/or MP2MP LSPs. Moreover, an LSP can be signaled to more than one root for RNR purposes.

The following diagram illustrates a root database on a branch/transit LSR:

```
        root 1.1.1.1:
            path1:
                RIB: GigEthernet 1/0, 12.1.0.2;
                LDP: peer 192.168.0.1:0
            path2:
                RIB: GigEthernet 2/0, 12.2.0.2;
                LDP: peer 192.168.0.3:0

        root 2.2.2.2:
            path1:
                RIB: 3.3.3.3;             (NOTE: This is a recursive path)
                LDP: peer 192.168.0.3:0   (NOTE: T-mLDP peer)

         root 9.9.9.9:
            . . . .
```

Figure 16

A root entry on a root LSR itself will be presented as follows:

```
          root 9.9.9.9:
              is-self
```

Figure 17

4.3.1.2. Bindings state

Binding state provides information on mLDP FEC-label bindings for both P2MP and MP2MP FEC types. Like LDP, the FEC-label binding derived state is presented in a FEC-centric view per address-family, and provides information on both inbound (received) and outbound (advertised) bindings. The FEC is presented as (root-address, opaque-type-data) and the direction (upstream or downstream) is picked with respect to root reachability. In case of MBB or/and MoFRR, the role of a given peer binding is also provided with respect to MBB (active or standby) or/and MoFRR (primary or backup).

This document covers following type of opaque values with their keys in the operational model of mLDP bindings:

Opaque Type	Key	RFC
Generic LSP Identifier	LSP Id	[RFC6388]
Transit IPv4 Source	Source, Group	[RFC6826]
Transit IPv6 Source	Source, Group	[RFC6826]
Transit IPv4 Bidir	RP, Group	[RFC6826]
Transit IPv6 Bidir	RP, Group	[RFC6826]
Transit VPNv4 Source	Source, Group, RD	[RFC7246]
Transit VPNv6 Source	Source, Group, RD	[RFC7246]
Transit VPNv4 Bidir	RP, Group, RD	[RFC7246]
Transit VPNv6 Bidir	RP, Group, RD	[RFC7246]
Recursive Opaque	Root	[RFC6512]
VPN-Recursive Opaque	Root, RD	[RFC6512]

Table 1: MP Opaque Types and keys

It is to be noted that there are three basic types (LSP Id, Source, and Bidir) and then there are variants (VPN, recursive, VPN-recursive) on top of these basic types.

Following captures high level tree hierarchy for mLDP bindings state:

```
+--rw mpls-ldp!
    +--rw mldp
      +--rw address-family* [afi]
         +--rw afi       address-family
         +--ro state
            +--ro ipv4 (or ipv6)
               +--ro bindings
                  +--ro opaque-type-xxx [root-address, type-specific-key]
                     +--ro root-address
                     +--ro ...
                     +--ro recur-root-address    inet:ipv4-address
                     +--ro recur-rd              route-distinguisher
                     +--ro multipoint-type?      multipoint-type
                     +--ro peer* [direction peer advertisement-type]
                        +--ro direction             downstream-upstream
                        +--ro peer                  leafref
                        +--ro advertisement-type    advertised-received
                        +--ro label?                mpls:mpls-label
                        +--ro mbb-role?             enumeration
                        +--ro mofrr-role?           enumeration
```

Figure 18

In the above tree, the type-specific-key varies with the base type as listed in earlier Table 1. For example, if the opaque type is Generic LSP Identifier, then the type-specific-key will be a uint32 value corresponding to the LSP. Please see the complete model for all other types.

Moreover, the binding tree defines only three types of sub-trees (i.e. lspid, src, and bidir) which is able to map the respective variants (vpn, recursive, and vpn-recusrive) accordingly. For example, the key for opaque-type-src is [R, S, G, rd, recur-R, recur-RD], where basic type will specify (R, S,G,-, -, -), VPN type will specify (R, S,G, rd, -, -), recursive type will specify [R, S,G, -, recur-R, -] and VPN-recursive type will specify [R, S,G, -, recur-R, recur-rd].

It is important to take note of the following:

- o The address-family ipv4/ipv4 applies to "root" address in the mLDP binding tree. The other addresses (source, group, RP etc) do not have to be of the same address family type as the root.

- o The "recur-root-address" field applies to Recursive opaque type, and (recur-root-address, recur-rd) fields applies to VPN-Recursive opaque types as defined in [RFC6512]

- o In case of a recursive FEC, the address-family of the recur-root-address could be different than the address-family of the root address of original encapsulated MP FEC

The following diagram illustrates the FEC-label binding information structure for a P2MP (Transit IPv4 Source type) LSP on a branch/ transit LSR:

```
         FEC (root 2.2.2.2, S=192.168.1.1, G=224.1.1.1):
           type: p2mp
           upstream:
             advertised:
               peer 192.168.0.1:0, label 16000 (local)
           downstream:
             received:
               peer 192.168.0.2:0, label 17000 (remote)
               peer 192.168.0.3:0, label 18000  (remote)
```

Figure 19

The following diagram illustrates the FEC-label binding information structure for a similar MP2MP LSP on a branch/transit LSR:

```
FEC (root 2.2.2.2, RP=192.168.9.9, G=224.1.1.1):
  type: mp2mp
  upstream:
    advertised:
      peer 192.168.0.1:0, label 16000 (local)
    received:
      peer 192.168.0.1:0, label 17000 (remote)
  downstream:
    advertised:
      peer 192.168.0.2:0, label 16001 (local), MBB role=active
      peer 192.168.0.3:0, label 16002 (local), MBB role=standby
    received:
      peer 192.168.0.2:0, label 17001 (remote)
      peer 192.168.0.3:0, label 18001 (remote)
```

Figure 20

4.3.1.3. Capabilities state

Like LDP, mLDP capabilities state comprise two types of information - global information and per-peer information.

4.4. Notifications

mLDP notification module consists of notification related to changes in the operational state of an mLDP FEC. Following is a simplified graphical representation of the data model for mLDP notifications:

```
notifications:
   +---n mpls-mldp-fec-event
      +--ro event-type?       oper-status-event-type
      +--ro tree-type?        multipoint-type
      +--ro root?             inet:ip-address
      +--ro (lsp-key-type)?
         +--:(lsp-id-based)
         |  +--ro lsp-id?           uint16
         +--:(source-group-based)
            +--ro source-address?   inet:ip-address
            +--ro group-address?    inet:ip-address
```

Figure 21

4.5. Actions

Currently, no RPCs/actions are defined for mLDP.

5. Open Items

Following is a list of open items that are to be discussed and addressed in future revisions of this document:

- o Close on augmentation off "mpls" list in "ietf-mpls" defined in [I-D.saad-mpls-base-yang]
- o Align operational state modeling with other routing procols and [I-D.openconfig-netmod-opstate]
- o Complete the section on Protocol-centric implementations and all-vrfs
- o Specify default values for configuration parameters
- o Revisit and cut down on the scope of the document and number of features it is trying to cover
- o Split the model into a base and extended items
- o Add statistics for mLDP root LSPs and bindings
- o Extend the "Configured Leaf LSPs" for various type of opaque-types
- o Extend mLDP notifications for other types of opaque values as well
- o Close on single vs separate document for mLDP Yang

6. YANG Specification

Following are actual YANG definition for LDP and mLDP constructs defined earlier in the document.

```
<CODE BEGINS> file "ietf-mpls-ldp@2016-07-08.yang" -->

module ietf-mpls-ldp {
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-ldp";
  // replace with IANA namespace when assigned
  prefix ldp;

  import ietf-inet-types {
```

```
     prefix "inet";
   }

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

   import ietf-interfaces {
     prefix "if";
   }

   import ietf-ip {
     prefix "ip";
   }

   import ietf-routing {
     prefix "rt";
   }

   import ietf-mpls {
     prefix "mpls";
   }

   organization
     "IETF MPLS Working Group";
   contact
     "WG Web:   <http://tools.ietf.org/wg/teas/>
      WG List:  <mailto:teas@ietf.org>

      WG Chair: Loa Andersson
                <mailto:loa@pi.nu>

      WG Chair: Ross Callon
                <mailto:rcallon@juniper.net>

      WG Chair: George Swallow
                <mailto:swallow.ietf@gmail.com>

      Editor:   Kamran Raza
                <mailto:skraza@cisco.com>

      Editor:   Rajiv Asati
                <mailto:rajiva@cisco.com>

      Editor:   Xufeng Liu
                <mailto:xliu@kuatrotech.com>

      Editor:   Santosh Esale
```

```
                <mailto:sesale@juniper.net>

      Editor:   Xia Chen
                <mailto:jescia.chenxia@huawei.com>

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

   description
     "This YANG module defines the essential components for the
      management of Multi-Protocol Label Switching (MPLS) Label
      Distribution Protocol (LDP) and Multipoint LDP (mLDP).";

   revision 2016-07-08 {
     description
       "Initial revision.";
     reference
       "RFC XXXX: YANG Data Model for MPLS LDP and mLDP.";
   }

   /*
    * Features
    */

   feature admin-down-config {
     description
       "This feature indicates that the system allows to configure
        administrative down on a VRF instance and a peer.";
   }

   feature all-af-policy-config {
     description
       "This feature indicates that the system allows to configure
        policies that are applied to all address families.";
   }

   feature capability-end-of-lib {
     description
       "This feature indicates that the system allows to configure
        LDP end-of-lib capability.";
   }

   feature capability-mldp-hsmp {
     description
       "This feature indicates that the system allows to configure
        mLDP hub-and-spoke-multipoint capability.";
   }
```

```
   feature capability-mldp-node-protection {
     description
       "This feature indicates that the system allows to configure
        mLDP node-protection capability.";
   }

   feature capability-typed-wildcard-fec {
     description
       "This feature indicates that the system allows to configure
        LDP typed-wildcard-fec capability.";
   }

   feature capability-upstream-label-assignment {
     description
       "This feature indicates that the system allows to configure
        LDP upstream label assignment capability.";
   }

   feature forwarding-nexthop-config {
     description
       "This feature indicates that the system allows to configure
        forwarding nexthop on interfaces.";
   }

   feature global-session-authentication {
     description
       "This feature indicates that the system allows to configure
        authentication at global level.";
   }

   feature graceful-restart-helper-mode {
     description
       "This feature indicates that the system supports graceful
        restart helper mode.";
   }

   feature mldp {
     description
       "This feature indicates that the system supports Multicast
        LDP (mLDP).";
   }

   feature mldp-mofrr {
     description
       "This feature indicates that the system supports mLDP
        Multicast only FRR (MoFRR).";
   }
```

```
   feature per-interface-timer-config {
     description
       "This feature indicates that the system allows to configure
        interface hello timers at the per-interface level.";
   }

   feature per-peer-graceful-restart-config {
     description
       "This feature indicates that the system allows to configure
        graceful restart at the per-peer level.";
   }

   feature per-peer-session-attributes-config {
     description
       "This feature indicates that the system allows to configure
        session attributes at the per-peer level.";
   }

   feature policy-extended-discovery-config {
     description
       "This feature indicates that the system allows to configure
        policies to control the acceptance of extended neighbor
        discovery hello messages.";
   }

   feature policy-label-assignment-config {
     description
       "This feature indicates that the system allows to configure
        policies to assign labels according to certain prefixes.";
   }

   feature policy-ordered-label-config {
     description
       "This feature indicates that the system allows to configure
        ordered label policies.";
   }

   feature session-downstream-on-demand-config {
     description
       "This feature indicates that the system allows to configure
        session downstream-on-demand";
   }

   /*
    * Typedefs
    */
   typedef ldp-address-family {
     type identityref {
```

```
        base rt:address-family;
      }
      description
        "LDP address family type.";
    }

    typedef duration32-inf {
      type union {
        type uint32;
        type enumeration {
          enum "infinite" {
            description "The duration is infinite.";
          }
        }
      }
      units seconds;
      description
        "Duration represented as 32 bit seconds with infinite.";
    }

    typedef advertised-received {
      type enumeration {
        enum advertised {
          description "Advertised information.";
        }
        enum received {
          description "Received information.";
        }
      }
      description
        "Received or advertised.";
    }

    typedef downstream-upstream {
      type enumeration {
        enum downstream {
          description "Downstream information.";
        }
        enum upstream {
          description "Upstream information.";
        }
      }
      description
        "Received or advertised.";
    }

    typedef label-adv-mode {
      type enumeration {
```

```
        enum downstream-unsolicited {
          description "Downstream Unsolicited.";
        }
        enum downstream-on-demand {
          description "Downstream on Demand.";
        }
      }
      description
        "Label Advertisement Mode.";
    }

    typedef mpls-interface-ref {
      type leafref {
        path "/rt:routing/mpls:mpls/mpls:interface/mpls:name";
      }
      description
        "This type is used by data models that need to reference
         mpls interfaces.";
    }

    typedef multipoint-type {
      type enumeration {
        enum p2mp {
          description "Point to multipoint.";
        }
        enum mp2mp {
          description "Multipoint to multipoint.";
        }
      }
      description
        "p2mp or mp2mp.";
    }

    typedef neighbor-list-ref {
      type string;
      description
        "A type for a reference to a neighbor list.";
    }

    typedef peer-list-ref {
      type string;
      description
        "A type for a reference to a peer list.";
    }

    typedef prefix-list-ref {
      type string;
      description
```

```
       "A type for a reference to a prefix list.";
   }

   typedef oper-status-event-type {
     type enumeration {
       enum up {
         value 1;
         description
           "Operational status changed to up.";
       }
       enum down {
         value 2;
         description
           "Operational status changed to down.";
       }
     }
     description "Operational status event type for notifications.";
   }

   typedef route-distinguisher {
     type string {
     }
     description
       "Type definition for route distinguisher.";
     reference
       "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs).";
   }

   /*
    * Identities
    */
   identity adjacency-flag-base {
     description "Base type for adjacency flags.";
   }

   identity adjacency-flag-active {
     base "adjacency-flag-base";
     description
       "This adjacency is configured and actively created.";
   }

   identity adjacency-flag-passive {
     base "adjacency-flag-base";
     description
       "This adjacency is not configured and passively accepted.";
   }

   /*
```

```
    * Groupings
    */

   grouping adjacency-state-attributes {
     description
       "Adjacency state attributes.";

     leaf-list flag {
       type identityref {
         base "adjacency-flag-base";
       }
       description "Adjacency flags.";
     }
     container hello-holdtime {
       description "Hello holdtime state.";
       leaf adjacent {
         type uint16;
         units seconds;
         description "Peer holdtime.";
       }
       leaf negotiated {
         type uint16;
         units seconds;
         description "Negotiated holdtime.";
       }
       leaf remaining {
         type uint16;
         units seconds;
         description "Remaining holdtime.";
       }
     }

     leaf next-hello {
       type uint16;
       units seconds;
       description "Time to send the next hello message.";
     }

     container statistics {
       description
         "Statistics objects.";

       leaf discontinuity-time {
         type yang:date-and-time;
         mandatory true;
         description
           "The time on the most recent occasion at which any one or
            more of this interface’s counters suffered a
```

```
            discontinuity.  If no such discontinuities have occurred
            since the last re-initialization of the local management
            subsystem, then this node contains the time the local
            management subsystem re-initialized itself.";
       }

       leaf hello-received {
         type yang:counter64;
         description
           "The number of hello messages received.";
       }
       leaf hello-dropped {
         type yang:counter64;
         description
           "The number of hello messages received.";
       }
     } // statistics
   } // adjacency-state-attributes

   grouping basic-discovery-timers {
     description
       "Basic discovery timer attributes.";
     leaf hello-holdtime {
       type uint16 {
         range 15..3600;
       }
       units seconds;
       description
         "The time interval for which a LDP link Hello adjacency
          is maintained in the absence of link Hello messages from
          the LDP neighbor";
     }
     leaf hello-interval {
       type uint16 {
         range 5..1200;
       }
       units seconds;
       description
         "The interval between consecutive LDP link Hello messages
          used in basic LDP discovery";
     }
   } // basic-discovery-timers

   grouping binding-address-state-attributes {
     description
       "Address binding attributes";
     leaf advertisement-type {
       type advertised-received;
```

```
        description
          "Received or advertised.";
      }
      leaf peer {
        type leafref {
          path "../../../../../../../peers/peer/lsr-id";
        }
        must "../advertisement-type = 'received'" {
          description
            "Applicable for received address.";
        }
        description
          "LDP peer from which this address is received.";
      } // peer
    } // binding-address-state-attributes

    grouping binding-label-state-attributes {
      description
        "Label binding attributes";
      list peer {
        key "peer advertisement-type";
        description
          "List of advertised and received peers.";
        leaf peer {
          type leafref {
            path "../../../../../../../../peers/peer/lsr-id";
          }
          description
            "LDP peer from which this binding is received,
             or to which this binding is advertised.";
        }
        leaf advertisement-type {
          type advertised-received;
          description
            "Received or advertised.";
        }
        leaf label {
          type mpls:mpls-label;
          description
            "Advertised (outbound) or received (inbound)
             label.";
        }
        leaf used-in-forwarding {
          type boolean;
          description
            "'true' if the lable is used in forwarding.";
        }
      } // peer
```

```
   } // binding-label-state-attributes

   grouping extended-discovery-policy-attributes {
     description
       "LDP policy to control the acceptance of extended neighbor
        discovery hello messages.";
     container hello-accept {
       if-feature policy-extended-discovery-config;
       description
         "Extended discovery acceptance policies.";

       leaf enable {
         type boolean;
         description
           "'true' to accept; 'false' to deny.";
       }
       leaf neighbor-list {
         type neighbor-list-ref;
         description

           "The name of a peer ACL.";
       }
     } // hello-accept
   } // extended-discovery-policy-attributes

   grouping extended-discovery-timers {
     description
       "Extended discovery timer attributes.";
     leaf hello-holdtime {
       type uint16 {
         range 15..3600;
       }
       units seconds;
       description
         "The time interval for which LDP targeted Hello adjacency

          is maintained in the absence of targeted Hello messages
          from an LDP neighbor.";
     }
     leaf hello-interval {
       type uint16 {
         range 5..3600;
       }
       units seconds;
       description
         "The interval between consecutive LDP targeted Hello
          messages used in extended LDP discovery.";
     }
```

```
   } // extended-discovery-timers

   grouping global-attributes {
     description "Configuration attributes at global level.";

     uses instance-attributes;
   } // global-attributes

   grouping graceful-restart-attributes {
     description
       "Graceful restart configuration attributes.";
     container graceful-restart {
       description
         "Attributes for graceful restart.";
       leaf enable {
         type boolean;
         description
           "Enable or disable graceful restart.";
       }
       leaf helper-enable {
         if-feature graceful-restart-helper-mode;
         type boolean;
         description
           "Enable or disable graceful restart helper mode.";
       }
       leaf reconnect-time {
         type uint16 {
           range 10..1800;
         }
         units seconds;
         description
           "Specifies the time interval that the remote LDP peer
            must wait for the local LDP peer to reconnect after the
            remote peer detects the LDP communication failure.";
       }
       leaf recovery-time {
         type uint16 {
           range 30..3600;
         }
         units seconds;
         description
           "Specifies the time interval, in seconds, that the remote
            LDP peer preserves its MPLS forwarding state after
            receiving the Initialization message from the restarted
            local LDP peer.";
       }
       leaf forwarding-holdtime {
         type uint16 {
```

```
            range 30..3600;
          }
          units seconds;
          description
            "Specifies the time interval, in seconds, before the
             termination of the recovery phase.";
        }
      } // graceful-restart
    } // graceful-restart-attributes

    grouping graceful-restart-attributes-per-peer {
      description
        "Per peer graceful restart configuration attributes.";
      container graceful-restart {
        description
          "Attributes for graceful restart.";
        leaf enable {
          type boolean;
          description
            "Enable or disable graceful restart.";
        }
        leaf reconnect-time {
          type uint16 {
            range 10..1800;
          }
          units seconds;
          description
            "Specifies the time interval that the remote LDP peer
             must wait for the local LDP peer to reconnect after the
             remote peer detects the LDP communication failure.";
        }
        leaf recovery-time {
          type uint16 {
            range 30..3600;
          }
          units seconds;
          description
            "Specifies the time interval, in seconds, that the remote
             LDP peer preserves its MPLS forwarding state after
             receiving the Initialization message from the restarted
             local LDP peer.";
        }
      } // graceful-restart
    } // graceful-restart-attributes-per-peer

    grouping instance-attributes {
      description "Configuration attributes at instance level.";
```

```
     container capability {
       description "Configure capability.";
       container end-of-lib {
         if-feature capability-end-of-lib;
         description
           "Configure end-of-lib capability.";
         leaf enable {
           type boolean;
           description
             "Enable end-of-lib capability.";
         }
       }
       container typed-wildcard-fec {
         if-feature capability-typed-wildcard-fec;
         description
           "Configure typed-wildcard-fec capability.";
         leaf enable {
           type boolean;
           description
             "Enable typed-wildcard-fec capability.";
         }
       }
       container upstream-label-assignment {
         if-feature capability-upstream-label-assignment;
         description
           "Configure upstream label assignment capability.";
         leaf enable {
           type boolean;
           description
             "Enable upstream label assignment.";
         }
       }
       container mldp {
         if-feature mldp;

         description
           "Multipoint capabilities.";
         uses mldp-capabilities;
       }
     } // capability

     uses graceful-restart-attributes;

     leaf igp-synchronization-delay {
       type uint16 {
         range 3..60;
       }
       units seconds;
```

```
       description
         "Sets the interval that the LDP waits before notifying the
          Interior Gateway Protocol (IGP) that label exchange is
          completed so that IGP can start advertising the normal
          metric for the link.";
     }
     leaf lsr-id {
       type yang:dotted-quad;
       description "Router ID.";
     }
   } // instance-attributes

   grouping ldp-adjacency-ref {
     description
       "An absolute reference to an LDP adjacency.";
     choice hello-adjacency-type {
       description
         "Interface or targeted adjacency.";
       case targeted {
         container targeted {
           description "Targeted adjacency.";
           leaf target-address {
             type inet:ip-address;
             description
               "The target address.";
           }
         } // targeted
       }
       case link {
         container link {
           description "Link adjacency.";
           leaf next-hop-interface {
             type mpls-interface-ref;
             description
               "Interface connecting to next-hop.";
           }
           leaf next-hop-address {
             type inet:ip-address;
             must "../next-hop-interface" {
               description
                 "Applicable when interface is specified.";

             }
             description
               "IP address of next-hop.";
           }
         } // link
       }
```

```
      }
    } // ldp-adjacency-ref

    grouping ldp-fec-event {
      description
        "A LDP FEC event.";
      leaf prefix {
        type inet:ip-prefix;
        description
          "FEC.";
      }
    } // ldp-fec-event

    grouping ldp-peer-ref {
      description
        "An absolute reference to an LDP peer.";
      leaf peer-ref {
        type leafref {
          path "/rt:routing/rt:control-plane-protocols/mpls-ldp/"
            + "peers/peer/lsr-id";
        }
        description
          "Reference to an LDP peer.";
      }
    } // ldp-peer-ref

    grouping mldp-capabilities {
      description
        "mLDP capabilities.";
      container p2mp {
        description
          "Configure point-to-multipoint capability.";
        leaf enable {
          type boolean;
          description
            "Enable point-to-multipoint.";
        }
      }
      container mp2mp {
        description
          "Configure multipoint-to-multipoint capability.";
        leaf enable {
          type boolean;
          description
            "Enable multipoint-to-multipoint.";
        }
      }
      container make-before-break {
```

```
        description
          "Configure make-before-break capability.";
        leaf enable {
          type boolean;
          description
            "Enable make-before-break.";
        }
        leaf switchover-delay {
          type uint16;
          units seconds;
          description
            "Switchover delay in seconds.";
        }
        leaf timeout {
          type uint16;
          units seconds;
          description
            "Timeout in seconds.";
        }
      }
      container hub-and-spoke {
        if-feature capability-mldp-hsmp;
        description
          "Configure hub-and-spoke-multipoint capability.";
        reference
          "RFC7140: LDP Extensions for Hub and Spoke Multipoint
           Label Switched Path";
        leaf enable {
          type boolean;
          description
            "Enable hub-and-spoke-multipoint.";
        }
      }
      container node-protection {
        if-feature capability-mldp-node-protection;
        description
          "Configure node-protection capability.";
        reference
          "RFC7715: mLDP Node Protection.";
        leaf plr {
          type boolean;
          description
            "Point of Local Repair capable for MP LSP node
             protection.";
        }
        container merge-point {
          description
            "Merge Point capable for MP LSP node protection.";
```

```
         leaf enable {
           type boolean;
           description
             "Enable merge point capability.";
         }
         leaf targeted-session-teardown-delay {
           type uint16;
           units seconds;
           description
             "Targeted session teardown delay.";
         }
       } // merge-point
     }
   } // mldp-capabilities

   grouping  mldp-configured-lsp-roots {
     description
       "mLDP roots containers.";

     container roots-ipv4 {

       when "../../../af = 'ipv4'" {
         description
           "Only for IPv4.";
       }
       description
         "Configured IPv4 multicast LSPs.";
       list root {
         key "root-address";
         description
           "List of roots for configured multicast LSPs.";

         leaf root-address {
           type inet:ipv4-address;
           description
             "Root address.";
         }

         list lsp {
           must "(lsp-id = 0 and source-address != '0.0.0.0' and "
             + "group-address != '0.0.0.0') or "
             + "(lsp-id != 0 and source-address = '0.0.0.0' and "
             + "group-address = '0.0.0.0')" {
             description
               "A LSP can be identified by either <lsp-id> or
                <source-address, group-address>.";
           }
           key "lsp-id source-address group-address";
```

```
          description
            "List of LSPs.";
          leaf lsp-id {
            type uint16;
            description "ID to identify the LSP.";
          }
          leaf source-address {
            type inet:ipv4-address;
            description
              "Source address.";
          }
          leaf group-address {
            type inet:ipv4-address-no-zone;
            description
              "Group address.";
          }
        } // list lsp
      } // list root
    } // roots-ipv4

    container roots-ipv6 {

      when "../../../af = 'ipv6'" {
        description
          "Only for IPv6.";
      }
      description
        "Configured IPv6 multicast LSPs.";

      list root {
        key "root-address";
        description
          "List of roots for configured multicast LSPs.";

        leaf root-address {
          type inet:ipv6-address;
          description
            "Root address.";
        }

        list lsp {
          must "(lsp-id = 0 and source-address != '::' and "
            + "group-address != '::') or "
            + "(lsp-id != 0 and source-address = '::' and "
            + "group-address = '::')" {
            description
              "A LSP can be identified by either <lsp-id> or
               <source-address, group-address>.";
```

```
           }
           key "lsp-id source-address group-address";
           description
             "List of LSPs.";
           leaf lsp-id {
             type uint16;
             description "ID to identify the LSP.";
           }
           leaf source-address {
             type inet:ipv6-address;
             description
               "Source address.";
           }
           leaf group-address {
             type inet:ipv6-address-no-zone;
             description
               "Group address.";
           }
         } // list lsp
       } // list root
     } // roots-ipv6
   } // mldp-configured-lsp-roots

   grouping mldp-fec-event {
     description
       "A mLDP FEC event.";
     leaf tree-type {
       type multipoint-type;
       description
         "p2mp or mp2mp.";
     }
     leaf root {
       type inet:ip-address;
       description
         "Root address.";
     }
     choice lsp-key-type {
       description
         "LSP ID based or source-group based .";
       case lsp-id-based {
         leaf lsp-id {
           type uint16;
           description
             "ID to identify the LSP.";
         }
       }
       case source-group-based {
         leaf source-address {
```

```
          type inet:ip-address;
          description

            "LSP source address.";
        }
        leaf group-address {
          type inet:ip-address;
          description
            "Multicast group address.";
        }
      } // case source-group-based
    }
  } // mldp-fec-event

  grouping  mldp-binding-label-state-attributes {
    description
      "mLDP label binding attributes.";

    leaf multipoint-type {
      type multipoint-type;
      description
        "The type of mutipoint, p2mp or mp2mp.";
    }
    list peer {
      key "direction peer advertisement-type";
      description
        "List of advertised and received peers.";
      leaf direction {
        type downstream-upstream;
        description
          "Downstream or upstream.";
      }
      leaf peer {
        type leafref {
          path
            "../../../../../../../../../../../peers/peer/lsr-id";
        }
        description
          "LDP peer from which this binding is received,
           or to which this binding is advertised.";
      }
      leaf advertisement-type {
        type advertised-received;
        description
          "Advertised or received.";
      }
      leaf label {
        type mpls:mpls-label;
```

```
         description
           "Advertised (outbound) or received (inbound) label.";
       }
       leaf mbb-role {
         when "../direction = 'upstream'" {
           description
             "For upstream.";
         }
         type enumeration {
           enum none {
             description "MBB is not enabled.";
           }
           enum active {
             description "This LSP is active.";
           }
           enum inactive {
             description "This LSP is inactive.";
           }
         }
         description
           "The MBB status of this LSP.";
       }
       leaf mofrr-role {
         when "../direction = 'upstream'" {
           description
             "For upstream.";
         }
         type enumeration {
           enum none {
             description "MOFRR is not enabled.";
           }
           enum primary {
             description "This LSP is primary.";
           }
           enum backup {
             description "This LSP is backup.";
           }
         }
         description
           "The MOFRR status of this LSP.";
       }
     } // peer
   } // mldp-binding-label-state-attributes

   grouping peer-af-policy-container {
     description
       "LDP policy attribute container under peer address-family.";
     container label-policy {
```

```
        description
          "Label policy attributes.";
        container advertise {
          description
            "Label advertising policies.";
          leaf prefix-list {
            type prefix-list-ref;
              description
                "Applies the prefix list to outgoing label
                 advertisements.";
          }
        }
        container accept {
          description
            "Label advertisement acceptance policies.";
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Applies the prefix list to incoming label
               advertisements.";
          }
        } // accept
      } // label-policy
    } // peer-af-policy-container

    grouping peer-attributes {
      description "Peer configuration attributes.";

      leaf session-ka-holdtime {
        type uint16 {
          range 45..3600;
        }
        units seconds;
        description
          "The time interval after which an inactive LDP session
           terminates and the corresponding TCP session closes.
           Inactivity is defined as not receiving LDP packets from the
           peer.";
      }
      leaf session-ka-interval {
        type uint16 {
          range 15..1200;
        }
        units seconds;
        description
          "The interval between successive transmissions of keepalive
           packets. Keepalive packets are only sent in the absence of
           other LDP packets transmitted over the LDP session.";
```

```
      }
    } // peer-attributes

    grouping peer-authentication {
      description
        "Peer authentication attributes.";
      leaf session-authentication-md5-password {
        type string {
          length "1..80";
        }
        description
          "Assigns an encrypted MD5 password to an LDP
           peer";
      } // md5-password
    } // peer-authentication

    grouping peer-state-derived {
      description "Peer derived state attributes.";

      container label-advertisement-mode {
        description "Label advertisement mode state.";
        leaf local {
          type label-adv-mode;
          description
            "Local Label Advertisement Mode.";
        }
        leaf peer {
          type label-adv-mode;
          description
            "Peer Label Advertisement Mode.";
        }
        leaf negotiated {
          type label-adv-mode;
          description
            "Negotiated Label Advertisement Mode.";
        }
      }
      leaf next-keep-alive {
        type uint16;
        units seconds;
        description "Time to send the next KeepAlive message.";
      }

      leaf peer-ldp-id {
        type yang:dotted-quad;
        description "Peer LDP ID.";
      }
```

```
     container received-peer-state {
       description "Peer features.";

       uses graceful-restart-attributes-per-peer;

       container capability {
         description "Configure capability.";
         container end-of-lib {
           description
             "Configure end-of-lib capability.";
           leaf enable {
             type boolean;
             description
               "Enable end-of-lib capability.";
           }
         }
         container typed-wildcard-fec {
           description
             "Configure typed-wildcard-fec capability.";
           leaf enable {
             type boolean;
             description
               "Enable typed-wildcard-fec capability.";
           }
         }
         container upstream-label-assignment {
           description
             "Configure upstream label assignment capability.";
           leaf enable {
             type boolean;
             description
               "Enable upstream label assignment.";
           }
         }
         container mldp {
           if-feature mldp;
           description
             "Multipoint capabilities.";

           container p2mp {
             description
               "Configure point-to-multipoint capability.";
             leaf enable {
               type boolean;
               description
                 "Enable point-to-multipoint.";
             }
           }
```

```
           container mp2mp {
             description
               "Configure multipoint-to-multipoint capability.";
             leaf enable {
               type boolean;
               description
                 "Enable multipoint-to-multipoint.";
             }
           }
           container make-before-break {
             description
               "Configure make-before-break capability.";
             leaf enable {
               type boolean;
               description
                 "Enable make-before-break.";
             }
           }
           container hub-and-spoke {
             description
               "Configure hub-and-spoke-multipoint capability.";
             reference
               "RFC7140: LDP Extensions for Hub and Spoke Multipoint
                Label Switched Path";
             leaf enable {
               type boolean;
               description
                 "Enable hub-and-spoke-multipoint.";
             }
           }
           container node-protection {
             description
               "Configure node-protection capability.";
             reference
               "RFC7715: mLDP Node Protection.";
             leaf plr {
               type boolean;
               description
                 "Point of Local Repair capable for MP LSP node
                  protection.";
             }
             leaf merge-point {
               type boolean;
               description
                 "Merge Point capable for MP LSP node protection.";
             } // merge-point
           } // node-protection
         } // mldp
```

```
      } // capability
    } // received-peer-state

    container session-holdtime {
      description "Session holdtime state.";
      leaf peer {
        type uint16;
        units seconds;
        description "Peer holdtime.";
      }
      leaf negotiated {
        type uint16;
        units seconds;
        description "Negotiated holdtime.";
      }
      leaf remaining {
        type uint16;
        units seconds;
        description "Remaining holdtime.";
      }
    } // session-holdtime

    leaf session-state {
      type enumeration {
        enum non-existent {
          description "NON EXISTENT state. Transport disconnected.";
        }
        enum initialized {
          description "INITIALIZED state.";
        }
        enum openrec {
          description "OPENREC state.";
        }
        enum opensent {
          description "OPENSENT state.";
        }
        enum operational {
          description "OPERATIONAL state.";
        }
      }
      description
        "Representing the operational status.";
    }

    container tcp-connection {
      description "TCP connection state.";
      leaf local-address {
        type inet:ip-address;
```

```
          description "Local address.";
        }
        leaf local-port {
          type inet:port-number;
          description "Local port.";
        }
        leaf remote-address {
          type inet:ip-address;
          description "Remote address.";
        }
        leaf remote-port {
          type inet:port-number;
          description "Remote port.";
        }
      } // tcp-connection

      leaf up-time {
        type string;
        description "Up time. The interval format in ISO 8601.";
      }

      container statistics {
        description
          "Statistics objects.";

        leaf discontinuity-time {
          type yang:date-and-time;
          mandatory true;
          description
            "The time on the most recent occasion at which any one or
             more of this interface's counters suffered a
             discontinuity.  If no such discontinuities have occurred
             since the last re-initialization of the local management
             subsystem, then this node contains the time the local
             management subsystem re-initialized itself.";
        }

        container received {
          description "Inbound statistics.";
          uses statistics-peer-received-sent;
        }
        container sent {
          description "Outbound statistics.";
          uses statistics-peer-received-sent;
        }

        leaf total-addresses {
          type uint32;
```

```
        description
          "The number of learned addresses.";
      }
      leaf total-labels {
        type uint32;
        description
          "The number of learned labels.";
      }
      leaf total-fec-label-bindings {
        type uint32;
        description
          "The number of learned label-address bindings.";
      }
    } // statistics
  } // peer-state-derived

  grouping policy-container {
    description
      "LDP policy attributes.";
    container label-policy {
      description
        "Label policy attributes.";
      container independent-mode {
        description
          "Independent label policy attributes.";
        container assign {

          if-feature policy-label-assignment-config;
          description
            "Label assignment policies";
          choice prefix-option {
            description
              "Use either prefix-list or host-routes-only.";
            case prefix-list {
              leaf prefix-list {
                type prefix-list-ref;
                description
                  "Assign labels according to certain prefixes.";
              }
            }
            case host-routes-only {
              leaf host-routes-only {
                type boolean;
                description
                  "'true' to apply host routes only.";
              }
            }
          } // prefix-option
```

```
        }
        container advertise {
          description
            "Label advertising policies.";
          container explicit-null {
            description
              "Enables an egress router to advertise an
               explicit null label (value 0) in place of an
               implicit null label (value 3) to the
               penultimate hop router.";
            leaf enable {
              type boolean;
              description
                "'true' to enable explicit null.";
            }
            leaf prefix-list {
              type prefix-list-ref;
              description
                "Prefix list name. Applies the filters in the
                 specified prefix list to label
                 advertisements.
                 If the prefix list is not specified, explicit
                 null label advertisement is enabled for all
                 directly connected prefixes.";
            }
          }
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Applies the prefix list to outgoing label
               advertisements.";

          }
        }
        container accept {
          description
            "Label advertisement acceptance policies.";
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Applies the prefix list to incoming label
               advertisements.";
          }
        }
      } // independent-mode
      container ordered-mode {
        if-feature policy-ordered-label-config;
        description
```

```
          "Ordered label policy attributes.";
        container egress-lsr {
          description
            "Egress LSR label assignment policies";
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Assign labels according to certain prefixes.";
          }
        }
        container advertise {
          description
            "Label advertising policies.";
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Applies the prefix list to outgoing label
               advertisements.";
          }
        }
        container accept {
          description
            "Label advertisement acceptance policies.";
          leaf prefix-list {
            type prefix-list-ref;
            description
              "Applies the prefix list to incoming label
               advertisements.";
          }
        }
      } // ordered-mode
    } // label-policy
  } // policy-container

  grouping statistics-peer-received-sent {
    description
      "Inbound and outbound statistic counters.";
    leaf total-octets {
      type yang:counter64;
      description
        "The total number of octets sent or received.";
    }
    leaf total-messages {
      type yang:counter64;
      description
        "The number of messages sent or received.";
    }
    leaf address {
```

```
        type yang:counter64;
        description
          "The number of address messages sent or received.";
      }
      leaf address-withdraw {
        type yang:counter64;
        description
          "The number of address-withdraw messages sent or received.";
      }
      leaf initialization {
        type yang:counter64;
        description
          "The number of initialization messages sent or received.";
      }
      leaf keepalive {
        type yang:counter64;
        description
          "The number of keepalive messages sent or received.";
      }
      leaf label-abort-request {
        type yang:counter64;
        description
          "The number of label-abort-request messages sent or
           received.";
      }
      leaf label-mapping {
        type yang:counter64;
        description
          "The number of label-mapping messages sent or received.";
      }
      leaf label-release {
        type yang:counter64;
        description
          "The number of label-release messages sent or received.";
      }
      leaf label-request {
        type yang:counter64;
        description
          "The number of label-request messages sent or received.";
      }
      leaf label-withdraw {
        type yang:counter64;
        description
          "The number of label-withdraw messages sent or received.";
      }
      leaf notification {
        type yang:counter64;
        description
```

```
         "The number of messages sent or received.";
     }
   } // statistics-peer-received-sent

   /*
    * Configuration data nodes
    */

   augment "/rt:routing/rt:control-plane-protocols" {
     description "LDP augmentation.";

     container mpls-ldp {
       presence "Container for LDP protocol.";
       description
         "Container for LDP protocol.";

       container global {
         description
           "Global attributes for LDP.";
         container config {
           description
             "Configuration data.";
           uses global-attributes;
         }
         container state {
           config false;
           description
             "Operational state data.";
           uses global-attributes;
         }

         container mldp {
           if-feature mldp;
           description
             "mLDP attributes at per instance level. Defining
              attributes here does not enable any MP capabilities.
              MP capabilities need to be explicitly enabled under
              container capability.";

           container config {
             description
               "Configuration data.";
             leaf enable {
               type boolean;
               description
                 "Enable mLDP.";
             }
           }
```

```
          container state {
            config false;
            description

              "Operational state data.";
            leaf enable {
              type boolean;
              description
                "Enable mLDP.";
            }
          }

          list address-family {
            key "afi";
            description
              "Per-af params.";
            leaf afi {
              type ldp-address-family;
              description
                "Address family type value.";
            }

            container config {
              description
                "Configuration data.";
              container multicast-only-frr {
                if-feature mldp-mofrr;
                description
                  "Multicast only FRR (MoFRR) policy.";
                leaf prefix-list {
                  type prefix-list-ref;
                  description
                    "Enables MoFRR for the specified access list.";
                }
              } // multicast-only-frr
              container recursive-fec {
                description
                  "Recursive FEC policy.";
                leaf prefix-list {
                  type prefix-list-ref;
                  description
                    "Enables recursive FEC for the specified access
                     list.";
                }
              } // recursive-for
            }
            container state {
              config false;
```

```
               description
                 "Operational state data.";
               container multicast-only-frr {
                 if-feature mldp-mofrr;

                 description
                   "Multicast only FRR (MoFRR) policy.";
                 leaf prefix-list {
                   type prefix-list-ref;
                   description
                     "Enables MoFRR for the specified access list.";
                 }
               } // multicast-only-frr
               container recursive-fec {
                 description
                   "Recursive FEC policy.";
                 leaf prefix-list {
                   type prefix-list-ref;
                   description
                     "Enables recursive FEC for the specified access
                      list.";
                 }
               } // recursive-fec

               container ipv4 {
                 when "../../afi = 'ipv4'" {
                   description
                     "Only for IPv4.";
                 }
                 description
                   "IPv4 state information.";
                 container roots {
                   description
                     "IPv4 multicast LSP roots.";
                   list root {
                     key "root-address";
                     description
                       "List of roots for configured multicast LSPs.";

                     leaf root-address {
                       type inet:ipv4-address;
                       description
                         "Root address.";
                     }

                     leaf is-self {
                       type boolean;
                       description
```

```
                 "This is the root.";
             }

             list reachability {
               key "address interface";
               description
                 "A next hop for reachability to root,
                  as a RIB view.";
               leaf address {
                 type inet:ipv4-address;
                 description
                   "The next hop address to reach root.";
               }
               leaf interface {
                 type mpls-interface-ref;
                 description
                   "Interface connecting to next-hop.";
               }
               leaf peer {
                 type leafref {
                   path
                     "../../../../../../../../../../peers/peer/"
                     + "lsr-id";
                 }
                 description
                   "LDP peer from which this next hop can be
                    reached.";
               }
             }
           } // list root
         } // roots
         container bindings {
           description
             "mLDP FEC to label bindings.";
           container opaque-type-lspid {
             description
               "The type of opaque value element is
                the generic LSP identifier";
             reference
               "RFC6388: Label Distribution Protocol
                Extensions for Point-to-Multipoint and
                Multipoint-to-Multipoint Label Switched
                Paths.";
             list fec-label {
               key
                 "root-address lsp-id "
                 + "recur-root-address recur-rd";
               description
```

```
                 "List of FEC to label bindings.";
               leaf root-address {
                 type inet:ipv4-address;
                 description
                   "Root address.";
               }
               leaf lsp-id {
                 type uint32;
                 description "ID to identify the LSP.";
               }
               leaf recur-root-address {
                 type inet:ip-address;
                 description
                   "Recursive root address.";
                 reference
                   "RFC6512: Using Multipoint LDP When the
                    Backbone Has No Route to the Root";
               }
               leaf recur-rd {
                 type route-distinguisher;
                 description
                   "Route Distinguisher in the VPN-Recursive
                    Opaque Value.";
                 reference
                   "RFC6512: Using Multipoint LDP When the
                    Backbone Has No Route to the Root";
               }
               uses mldp-binding-label-state-attributes;
             } // fec-label
           } // opaque-type-lspid

           container opaque-type-src {
             description
               "The type of opaque value element is
                the transit source TLV";
             reference
               "RFC6826: Multipoint LDP In-Band Signaling for
                Point-to-Multipoint and
                Multipoint-to-Multipoint Label Switched
                Paths.";
             list fec-label {
               key
                 "root-address source-address group-address "
                 + "rd recur-root-address recur-rd";
               description
                 "List of FEC to label bindings.";
               leaf root-address {
                 type inet:ipv4-address;
```

```
                  description
                    "Root address.";
                }
                leaf source-address {
                  type inet:ip-address;
                  description
                    "Source address.";
                }
                leaf group-address {
                  type inet:ip-address-no-zone;
                  description
                    "Group address.";
                }
                leaf rd {
                  type route-distinguisher;
                  description
                    "Route Distinguisher.";
                  reference
                    "RFC7246: Multipoint Label Distribution
                     Protocol In-Band Signaling in a Virtual
                     Routing and Forwarding (VRF) Table
                     Context.";
                }
                leaf recur-root-address {
                  type inet:ip-address;
                  description
                    "Recursive root address.";
                  reference
                    "RFC6512: Using Multipoint LDP When the
                     Backbone Has No Route to the Root";
                }
                leaf recur-rd {
                  type route-distinguisher;
                  description
                    "Route Distinguisher in the VPN-Recursive
                     Opaque Value.";
                  reference
                    "RFC6512: Using Multipoint LDP When the
                     Backbone Has No Route to the Root";
                }
                uses mldp-binding-label-state-attributes;
              } // fec-label
            } // opaque-type-src

            container opaque-type-bidir {
              description
                "The type of opaque value element is
                 the generic LSP identifier";
```

```
                     reference
                       "RFC6826: Multipoint LDP In-Band Signaling for
                        Point-to-Multipoint and
                        Multipoint-to-Multipoint Label Switched
                        Paths.";
                     list fec-label {
                       key
                         "root-address rp group-address "
                         + "rd recur-root-address recur-rd";
                       description
                         "List of FEC to label bindings.";
                       leaf root-address {
                         type inet:ipv4-address;
                         description
                           "Root address.";
                       }
                       leaf rp {
                         type inet:ip-address;
                         description
                           "RP address.";
                       }
                       leaf group-address {
                         type inet:ip-address-no-zone;
                         description
                           "Group address.";
                       }
                       leaf rd {
                         type route-distinguisher;
                         description
                           "Route Distinguisher.";
                         reference
                           "RFC7246: Multipoint Label Distribution
                            Protocol In-Band Signaling in a Virtual
                            Routing and Forwarding (VRF) Table
                            Context.";
                       }
                       leaf recur-root-address {
                         type inet:ip-address;
                         description
                           "Recursive root address.";
                         reference
                           "RFC6512: Using Multipoint LDP When the
                            Backbone Has No Route to the Root";
                       }
                       leaf recur-rd {
                         type route-distinguisher;
                         description
                           "Route Distinguisher in the VPN-Recursive
```

```
                        Opaque Value.";
                     reference
                       "RFC6512: Using Multipoint LDP When the
                        Backbone Has No Route to the Root";
                   }
                   uses mldp-binding-label-state-attributes;
                 } // fec-label
               } // opaque-type-bidir
             } // bindings
           } // ipv4

           container ipv6 {
             when "../../afi = 'ipv6'" {
               description
                 "Only for IPv6.";
             }
             description
               "IPv6 state information.";
             container roots {
               description
                 "IPv6 multicast LSP roots.";
               list root {
                 key "root-address";
                 description
                   "List of roots for configured multicast LSPs.";

                 leaf root-address {
                   type inet:ipv6-address;
                   description
                     "Root address.";
                 }

                 leaf is-self {
                   type boolean;
                   description
                     "This is the root.";
                 }

                 list reachability {
                   key "address interface";
                   description
                     "A next hop for reachability to root,
                      as a RIB view.";
                   leaf address {
                     type inet:ipv6-address;
                     description
                       "The next hop address to reach root.";
                   }
```

```
                         leaf interface {
                           type mpls-interface-ref;
                           description
                             "Interface connecting to next-hop.";
                         }
                         leaf peer {
                           type leafref {
                             path
                               "../../../../../../../../../peers/peer/"
                               + "lsr-id";
                           }
                           description
                             "LDP peer from which this next hop can be
                              reached.";
                         }
                       }
                     } // list root
                   } // roots
                   container bindings {
                     description
                       "mLDP FEC to label bindings.";
                     container opaque-type-lspid {
                       description
                         "The type of opaque value element is
                          the generic LSP identifier";
                       reference
                         "RFC6388: Label Distribution Protocol
                          Extensions for Point-to-Multipoint and
                          Multipoint-to-Multipoint Label Switched
                          Paths.";
                       list fec-label {
                         key
                           "root-address lsp-id "
                           + "recur-root-address recur-rd";
                         description
                           "List of FEC to label bindings.";
                         leaf root-address {
                           type inet:ipv6-address;
                           description
                             "Root address.";
                         }
                         leaf lsp-id {
                           type uint32;
                           description "ID to identify the LSP.";
                         }
                         leaf recur-root-address {
                           type inet:ip-address;
                           description
```

```
                "Recursive root address.";
              reference
                "RFC6512: Using Multipoint LDP When the
                 Backbone Has No Route to the Root";
            }
            leaf recur-rd {
              type route-distinguisher;
              description
                "Route Distinguisher in the VPN-Recursive
                 Opaque Value.";
              reference
                "RFC6512: Using Multipoint LDP When the
                 Backbone Has No Route to the Root";
            }
            uses mldp-binding-label-state-attributes;
          } // fec-label
        } // opaque-type-lspid

        container opaque-type-src {
          description
            "The type of opaque value element is
             the transit Source TLV";
          reference
            "RFC6826: Multipoint LDP In-Band Signaling for
             Point-to-Multipoint and
             Multipoint-to-Multipoint Label Switched
             Paths.";
          list fec-label {
            key
              "root-address source-address group-address "
              + "rd recur-root-address recur-rd";
            description
              "List of FEC to label bindings.";
            leaf root-address {
              type inet:ipv6-address;
              description
                "Root address.";
            }
            leaf source-address {
              type inet:ip-address;
              description
                "Source address.";
            }
            leaf group-address {
              type inet:ip-address-no-zone;
              description
                "Group address.";
            }
```

```
            leaf rd {
              type route-distinguisher;
              description
                "Route Distinguisher.";
              reference
                "RFC7246: Multipoint Label Distribution
                 Protocol In-Band Signaling in a Virtual
                 Routing and Forwarding (VRF) Table
                 Context.";
            }
            leaf recur-root-address {
              type inet:ip-address;
              description
                "Recursive root address.";
              reference
                "RFC6512: Using Multipoint LDP When the
                 Backbone Has No Route to the Root";
            }
            leaf recur-rd {
              type route-distinguisher;
              description
                "Route Distinguisher in the VPN-Recursive
                 Opaque Value.";
              reference
                "RFC6512: Using Multipoint LDP When the
                 Backbone Has No Route to the Root";
            }
            uses mldp-binding-label-state-attributes;
          } // fec-label
        } // opaque-type-src

        container opaque-type-bidir {
          description
            "The type of opaque value element is
             the generic LSP identifier";
          reference
            "RFC6826: Multipoint LDP In-Band Signaling for
             Point-to-Multipoint and
             Multipoint-to-Multipoint Label Switched
             Paths.";
          list fec-label {
            key
              "root-address rp group-address "
              + "rd recur-root-address recur-rd";
            description
              "List of FEC to label bindings.";
            leaf root-address {
              type inet:ipv6-address;
```

```
                    description
                      "Root address.";
                  }
                  leaf rp {
                    type inet:ip-address;
                    description
                      "RP address.";
                  }
                  leaf group-address {
                    type inet:ip-address-no-zone;
                    description
                      "Group address.";
                  }
                  leaf rd {
                    type route-distinguisher;
                    description
                      "Route Distinguisher.";
                    reference
                      "RFC7246: Multipoint Label Distribution
                       Protocol In-Band Signaling in a Virtual
                       Routing and Forwarding (VRF) Table
                       Context.";
                  }
                  leaf recur-root-address {
                    type inet:ip-address;
                    description
                      "Recursive root address.";
                    reference
                      "RFC6512: Using Multipoint LDP When the
                       Backbone Has No Route to the Root";
                  }
                  leaf recur-rd {
                    type route-distinguisher;
                    description
                      "Route Distinguisher in the VPN-Recursive
                       Opaque Value.";
                    reference
                      "RFC6512: Using Multipoint LDP When the
                       Backbone Has No Route to the Root";
                  }
                  uses mldp-binding-label-state-attributes;
                } // fec-label
              } // opaque-type-bidir
            } // bindings
          } // ipv6
        } // state

        container configured-leaf-lsps {
```

```
               description
                 "Configured multicast LSPs.";

               container p2mp {
                 description
                   "Configured point-to-multipoint LSPs.";
                 uses mldp-configured-lsp-roots;
               }
               container mp2mp {
                 description
                   "Configured multipoint-to-multipoint LSPs.";
                 uses mldp-configured-lsp-roots;
               }
             } // configured-leaf-lsps
           } // list address-family
         } // mldp

         list address-family {
           key "afi";
           description
             "Per-vrf per-af params.";
           leaf afi {
             type ldp-address-family;
             description
               "Address family type value.";
           }

           container config {
             description
               "Configuration data.";
             leaf enable {
               type boolean;
               description
                 "'true' to enable the address family.";
             }
             uses policy-container;

             container ipv4 {
               when "../../afi = 'ipv4'" {
                 description
                   "Only for IPv4.";
               }
               description
                 "IPv4 address family.";
               leaf transport-address {
                 type inet:ipv4-address;
                 description
                   "The transport address advertised in LDP Hello
```

```
                    messages.";
               }
             } // ipv4
             container ipv6 {
               when "../../afi = 'ipv6'" {
                 description
                   "Only for IPv6.";
               }
               description
                 "IPv6 address family.";
               leaf transport-address {
                 type inet:ipv6-address;
                 description
                   "The transport address advertised in LDP Hello
                    messages.";
               }
             } // ipv6
           }
           container state {
             config false;
             description
               "Operational state data.";
             leaf enable {
               type boolean;
               description
                 "'true' to enable the address family.";
             }

             uses policy-container;

             container ipv4 {
               when "../../afi = 'ipv4'" {
                 description
                   "Only for IPv4.";
               }
               description
                 "IPv4 address family.";
               leaf transport-address {
                 type inet:ipv4-address;
                 description
                   "The transport address advertised in LDP Hello
                    messages.";
               }

               container bindings {
                 description
                   "LDP address and label binding information.";
                 list address {
```

```
                   key "address";
                   description
                     "List of address bindings.";
                   leaf address {
                     type inet:ipv4-address;
                     description
                       "Binding address.";
                   }
                   uses binding-address-state-attributes;
                 } // binding-address

                 list fec-label {
                   key "fec";
                   description
                     "List of label bindings.";
                   leaf fec {
                     type inet:ipv4-prefix;
                     description
                       "Prefix FEC.";
                   }
                   uses binding-label-state-attributes;
                 } // fec-label
               } // binding
             } // ipv4
             container ipv6 {
               when "../../afi = 'ipv6'" {
                 description
                   "Only for IPv6.";
               }
               description
                 "IPv6 address family.";
               leaf transport-address {
                 type inet:ipv6-address;
                 description
                   "The transport address advertised in LDP Hello
                    messages.";
               }

               container binding {
                 description
                   "LDP address and label binding information.";
                 list address {
                   key "address";
                   description
                     "List of address bindings.";
                   leaf address {
                     type inet:ipv6-address;
                     description
```

```
                    "Binding address.";
                }
                uses binding-address-state-attributes;
              } // binding-address

              list fec-label {
                key "fec";
                description
                  "List of label bindings.";
                leaf fec {
                  type inet:ipv6-prefix;
                  description
                    "Prefix FEC.";
                }
                uses binding-label-state-attributes;
              } // fec-label
            } // binding
          } // ipv6
        } // state
      } // address-family

      container discovery {
        description
          "Neibgbor discovery configuration.";

        container interfaces {
          description
            "A list of interfaces for basic descovery.";
          container config {
            description
              "Configuration data.";
            uses basic-discovery-timers;
          }
          container state {
            config false;
            description

              "Operational state data.";
            uses basic-discovery-timers;
          }

          list interface {
            key "interface";
            description
              "List of LDP interfaces.";
            leaf interface {
              type mpls-interface-ref;
              description
```

```
                "Interface.";
            }
            container config {
              description
                "Configuration data.";
              uses basic-discovery-timers {
                if-feature per-interface-timer-config;
              }
              leaf igp-synchronization-delay {
                if-feature per-interface-timer-config;
                type uint16 {
                  range 3..60;
                }
                units seconds;
                description
                  "Sets the interval that the LDP waits before
                   notifying the Interior Gateway Protocol (IGP)
                   that label exchange is completed so that IGP
                   can start advertising the normal metric for
                   the link.";
              }
            }
            container state {
              config false;
              description
                "Operational state data.";
              uses basic-discovery-timers {
                if-feature per-interface-timer-config;
              }
              leaf igp-synchronization-delay {
                if-feature per-interface-timer-config;
                type uint16 {
                  range 3..60;
                }
                units seconds;
                description
                  "Sets the interval that the LDP waits before
                   notifying the Interior Gateway Protocol (IGP)
                   that label exchange is completed so that IGP
                   can start advertising the normal metric for
                   the link.";
              }
              leaf next-hello {
                type uint16;
                units seconds;
                description "Time to send the next hello message.";
              }
            } // state
```

```
               list address-family {
                 key "afi";
                 description
                   "Per-vrf per-af params.";
                 leaf afi {
                   type ldp-address-family;
                   description
                     "Address family type value.";
                 }
                 container config {
                   description
                     "Configuration data.";
                   leaf enable {
                     type boolean;
                     description
                       "Enable the address family on the interface.";
                   }

                   container ipv4 {
                     must "/if:interfaces/if:interface"
                       + "[name = current()/../../../interface]/"
                       + "ip:ipv4" {
                       description
                         "Only if IPv4 is enabled on the interface.";
                     }
                     description
                       "IPv4 address family.";
                     leaf transport-address {
                       type union {
                         type enumeration {
                           enum "use-interface-address" {
                             description
                             "Use interface address as the transport
                              address.";
                           }
                         }
                         type inet:ipv4-address;
                       }
                       description
                         "IP address to be advertised as the LDP
                          transport address.";
                     }
                   }

                   container ipv6 {
                     must "/if:interfaces/if:interface"
                       + "[name = current()/../../../interface]/"
                       + "ip:ipv6" {
```

```
              description
                "Only if IPv6 is enabled on the interface.";
            }
            description
              "IPv6 address family.";
            leaf transport-address {
              type union {
                type enumeration {
                  enum "use-interface-address" {
                    description
                      "Use interface address as the transport
                       address.";
                  }
                }
                type inet:ipv4-address;
              }
              description
                "IP address to be advertised as the LDP
                 transport address.";
            }
          } // ipv6
        }
        container state {
          config false;
          description
            "Operational state data.";
          leaf enable {
            type boolean;
            description
              "Enable the address family on the interface.";
          }

          container ipv4 {
            must "/if:interfaces/if:interface"
              + "[name = current()/../../../interface]/"
              + "ip:ipv4" {
              description
                "Only if IPv4 is enabled on the interface.";
            }
            description
              "IPv4 address family.";
            leaf transport-address {
              type union {
                type enumeration {

                  enum "use-interface-address" {
                    description
                      "Use interface address as the transport
```

```
                                address.";
                           }
                         }
                         type inet:ipv4-address;
                       }
                       description
                         "IP address to be advertised as the LDP
                          transport address.";
                     }

                     list hello-adjacencies {
                       key "adjacent-address";
                       description "List of hello adjacencies.";

                       leaf adjacent-address {
                         type inet:ipv4-address;
                         description
                           "Neighbor address of the hello adjacency.";
                       }

                       uses adjacency-state-attributes;

                       leaf peer {
                         type leafref {
                           path "../../../../../../../../../../peers/peer/"
                             + "lsr-id";
                         }
                         description
                           "LDP peer from this adjacency.";
                       }
                     } // hello-adjacencies
                   }
                   container ipv6 {
                     must "/if:interfaces/if:interface"
                       + "[name = current()/../../../interface]/"
                       + "ip:ipv6" {
                       description
                         "Only if IPv6 is enabled on the interface.";
                     }
                     description
                       "IPv6 address family.";
                     leaf transport-address {
                       type union {
                         type enumeration {
                           enum "use-interface-address" {
                             description
                             "Use interface address as the transport
                              address.";
```

```
                          }
                        }
                        type inet:ipv4-address;
                      }
                      description
                        "IP address to be advertised as the LDP
                         transport address.";
                    }

                    list hello-adjacencies {
                      key "adjacent-address";
                      description "List of hello adjacencies.";

                      leaf adjacent-address {
                        type inet:ipv6-address;
                        description
                          "Neighbor address of the hello adjacency.";
                      }

                      uses adjacency-state-attributes;

                      leaf peer {
                        type leafref {
                          path "../../../../../../../../../../peers/peer/"
                            + "lsr-id";
                        }
                        description
                          "LDP peer from this adjacency.";
                      }
                    } // hello-adjacencies
                  } // ipv6
                }
              } // address-family
            } // list interface
          } // interfaces

          container targeted
          {
            description
              "A list of targeted neighbors for extended discovery.";
            container config {

              description
                "Configuration data.";
              uses extended-discovery-timers;
              uses extended-discovery-policy-attributes;
            }
            container state {
```

```
              config false;
              description
                "Operational state data.";
              uses extended-discovery-timers;
              uses extended-discovery-policy-attributes;
            }

            list address-family {
              key "afi";
              description
                "Per-af params.";
              leaf afi {
                type ldp-address-family;
                description
                  "Address family type value.";
              }

              container state {
                config false;
                description
                  "Operational state data.";

                container ipv4 {
                  when "../../afi = 'ipv4'" {
                    description
                      "For IPv4.";
                  }
                  description
                    "IPv4 address family.";
                  list hello-adjacencies {
                    key "local-address adjacent-address";
                    description "List of hello adjacencies.";

                    leaf local-address {
                      type inet:ipv4-address;
                      description
                        "Local address of the hello adjacency.";
                    }
                    leaf adjacent-address {
                      type inet:ipv4-address;
                      description
                        "Neighbor address of the hello adjacency.";
                    }

                    uses adjacency-state-attributes;

                    leaf peer {
                      type leafref {
```

```
                 path "../../../../../../../../peers/peer/"
                   + "lsr-id";
               }
               description
                 "LDP peer from this adjacency.";
             }
           } // hello-adjacencies
         } // ipv4

         container ipv6 {
           when "../../afi = 'ipv6'" {
             description
               "For IPv6.";
           }
           description
             "IPv6 address family.";
           list hello-adjacencies {
             key "local-address adjacent-address";
             description "List of hello adjacencies.";

             leaf local-address {
               type inet:ipv6-address;
               description
                 "Local address of the hello adjacency.";
             }
             leaf adjacent-address {
               type inet:ipv6-address;
               description
                 "Neighbor address of the hello adjacency.";
             }

             uses adjacency-state-attributes;

             leaf peer {
               type leafref {
                 path "../../../../../../../../peers/peer/"
                   + "lsr-id";
               }
               description
                 "LDP peer from this adjacency.";
             }
           } // hello-adjacencies
         } // ipv6
       } // state

       container ipv4 {
         when "../afi = 'ipv4'" {
           description
```

```
                    "For IPv4.";
                }
                description
                  "IPv4 address family.";
                list target {
                  key "adjacent-address";
                  description
                    "Targeted discovery params.";

                  leaf adjacent-address {
                    type inet:ipv4-address;
                    description
                      "Configures a remote LDP neighbor and enables
                       extended LDP discovery of the specified
                       neighbor.";
                  }
                  container config {
                    description
                      "Configuration data.";
                    leaf enable {
                      type boolean;
                      description
                        "Enable the target.";
                    }
                    leaf local-address {
                      type inet:ipv4-address;
                      description
                        "The local address.";
                    }
                  }
                  container state {
                    config false;
                    description
                      "Operational state data.";
                    leaf enable {
                      type boolean;
                      description
                        "Enable the target.";
                    }
                    leaf local-address {
                      type inet:ipv4-address;
                      description
                        "The local address.";
                    }
                  } // state
                }
              } // ipv4
              container ipv6 {
```

```
                when "../afi = 'ipv6'" {
                  description
                    "For IPv6.";
                }
                description
                  "IPv6 address family.";
                list target {
                  key "adjacent-address";
                  description
                    "Targeted discovery params.";

                  leaf adjacent-address {
                    type inet:ipv6-address;
                    description
                      "Configures a remote LDP neighbor and enables
                       extended LDP discovery of the specified
                       neighbor.";
                  }
                  container config {
                    description
                      "Configuration data.";
                    leaf enable {
                      type boolean;
                      description
                        "Enable the target.";
                    }
                    leaf local-address {
                      type inet:ipv6-address;
                      description
                        "The local address.";
                    }
                  }
                  container state {
                    config false;
                    description
                      "Operational state data.";
                    leaf enable {
                      type boolean;
                      description
                        "Enable the target.";
                    }
                    leaf local-address {
                      type inet:ipv6-address;
                      description
                        "The local address.";
                    }
                  } // state
                }
```

```
              } // ipv6
            } // address-family
          } // targeted
        } // discovery

        container forwarding-nexthop {
          if-feature forwarding-nexthop-config;
          description
            "Configuration for forwarding nexthop.";

          container interfaces {
            description
              "A list of interfaces on which forwarding is
               disabled.";

            list interface {
              key "interface";
              description
                "List of LDP interfaces.";
              leaf interface {
                type mpls-interface-ref;
                description
                  "Interface.";
              }
              list address-family {
                key "afi";
                description
                  "Per-vrf per-af params.";
                leaf afi {
                  type ldp-address-family;
                  description
                    "Address family type value.";
                }
                container config {
                  description
                    "Configuration data.";
                  leaf ldp-disable {
                    type boolean;
                    description
                      "Disable LDP forwarding on the interface.";
                  }
                  leaf mldp-disable {
                    if-feature mldp;
                    type boolean;
                    description
                      "Disable mLDP forwarding on the interface.";
                  }
                }
```

```
                 container state {
                   config false;
                   description
                     "Operational state data.";
                   leaf ldp-disable {
                     type boolean;
                     description
                       "Disable LDP forwarding on the interface.";
                   }
                   leaf mldp-disable {
                     if-feature mldp;

                     type boolean;
                     description
                       "Disable mLDP forwarding on the interface.";
                   }
                 }
               } // address-family
             } // list interface
           } // interfaces
         } // forwarding-nexthop
         uses policy-container {
           if-feature all-af-policy-config;
         }
       } // global

       container peers {
         description
           "Peers configuration attributes.";

         container config {
           description
             "Configuration data.";
           uses peer-authentication {
             if-feature global-session-authentication;
           }
           uses peer-attributes;

           container session-downstream-on-demand {
             if-feature session-downstream-on-demand-config;
             description
               "Session downstream-on-demand attributes.";
             leaf enable {
               type boolean;
               description
                 "'true' if session downstream-on-demand is enabled.";
             }
             leaf peer-list {
```

```
               type peer-list-ref;
               description
                 "The name of a peer ACL.";
             }
           }
         }
         container state {
           config false;
           description
             "Operational state data.";
           uses peer-authentication {
             if-feature global-session-authentication;
           }
           uses peer-attributes;

           container session-downstream-on-demand {
             if-feature session-downstream-on-demand-config;
             description
               "Session downstream-on-demand attributes.";
             leaf enable {
               type boolean;
               description
                 "'true' if session downstream-on-demand is enabled.";
             }
             leaf peer-list {
               type peer-list-ref;
               description
                 "The name of a peer ACL.";
             }
           }
         }

         list peer {
           key "lsr-id";
           description
             "List of peers.";

           leaf lsr-id {
             type yang:dotted-quad;
             description "LSR ID.";
           }

           container config {
             description
               "Configuration data.";
             leaf admin-down {
               type boolean;
               default false;
```

```
              description
                "'true' to disable the peer.";
            }

            container capability {
              description
                "Per peer capability";
              container mldp {
                if-feature mldp;
                description
                  "mLDP capabilities.";
                uses mldp-capabilities;
              }
            }

            uses peer-af-policy-container {
              if-feature all-af-policy-config;
            }

            uses peer-authentication;

            uses graceful-restart-attributes-per-peer {
              if-feature per-peer-graceful-restart-config;
            }

            uses peer-attributes {
              if-feature per-peer-session-attributes-config;
            }

            container address-family {
              description
                "Per-vrf per-af params.";
              container ipv4 {
                description
                  "IPv4 address family.";
                uses peer-af-policy-container;
              }
              container ipv6 {
                description
                  "IPv6 address family.";
                uses peer-af-policy-container;
              } // ipv6
            } // address-family
          }
          container state {
            config false;
            description
              "Operational state data.";
```

```
          leaf admin-down {
            type boolean;
            default false;
            description
              "'true' to disable the peer.";
          }

          container capability {
            description
              "Per peer capability";
            container mldp {
              if-feature mldp;
              description
                "mLDP capabilities.";
              uses mldp-capabilities;
            }
          }

          uses peer-af-policy-container {
            if-feature all-af-policy-config;
          }

          uses peer-authentication;

          uses graceful-restart-attributes-per-peer {
            if-feature per-peer-graceful-restart-config;
          }

          uses peer-attributes {
            if-feature per-peer-session-attributes-config;
          }

          container address-family {
            description
              "Per-vrf per-af params.";
            container ipv4 {
              description
                "IPv4 address family.";
              uses peer-af-policy-container;

              list hello-adjacencies {
                key "local-address adjacent-address";
                description "List of hello adjacencies.";

                leaf local-address {
                  type inet:ipv4-address;
                  description
                    "Local address of the hello adjacency.";
```

```
                   }
                   leaf adjacent-address {
                     type inet:ipv4-address;
                     description
                       "Neighbor address of the hello adjacency.";
                   }

                   uses adjacency-state-attributes;

                   leaf interface {
                     type mpls-interface-ref;
                     description "Interface for this adjacency.";
                   }
                 } // hello-adjacencies
               } // ipv4
               container ipv6 {
                 description
                   "IPv6 address family.";
                 uses peer-af-policy-container;

                 list hello-adjacencies {
                   key "local-address adjacent-address";
                   description "List of hello adjacencies.";

                   leaf local-address {
                     type inet:ipv6-address;
                     description
                       "Local address of the hello adjacency.";
                   }
                   leaf adjacent-address {
                     type inet:ipv6-address;
                     description
                       "Neighbor address of the hello adjacency.";
                   }

                   uses adjacency-state-attributes;

                   leaf interface {
                     type mpls-interface-ref;
                     description "Interface for this adjacency.";
                   }
                 } // hello-adjacencies
               } // ipv6
             } // address-family

             uses peer-state-derived;
           } // state
         } // list peer
```

```
        } // peers
      } // container mpls-ldp
    }

    /*
     * RPCs
     */
    rpc mpls-ldp-clear-peer {
      description
        "Clears the session to the peer.";
      input {
        leaf lsr-id {
          type union {
            type yang:dotted-quad;
            type uint32;
          }
          description
            "LSR ID of peer to be cleared. If this is not provided
             then all peers are cleared";
        }
      }
    }

    rpc mpls-ldp-clear-hello-adjacency {
      description
        "Clears the hello adjacency";
      input {
        container hello-adjacency {
          description
            "Link adjacency or targettted adjacency. If this is not
             provided then all hello adjacencies are cleared";
          choice hello-adjacency-type {
            description "Adjacency type.";
            case targeted {
              container targeted {
                presence "Present to clear targeted adjacencies.";
                description
                  "Clear targeted adjacencies.";
                leaf target-address {
                  type inet:ip-address;
                  description
                    "The target address. If this is not provided then
                     all targeted adjacencies are cleared";
                }
              } // targeted
            }
            case link {
              container link {
```

```
               presence "Present to clear link adjacencies.";
               description
                 "Clear link adjacencies.";
               leaf next-hop-interface {
                 type mpls-interface-ref;
                 description

                   "Interface connecting to next-hop. If this is not
                    provided then all link adjacencies are cleared.";
               }
               leaf next-hop-address {
                 type inet:ip-address;
                 must "../next-hop-interface" {
                   description
                     "Applicable when interface is specified.";
                 }
                 description
                   "IP address of next-hop. If this is not provided
                    then adjacencies to all next-hops on the given
                    interface are cleared.";
               } // next-hop-address
             } // link
           }
         }
       }
     }
   }

   rpc mpls-ldp-clear-peer-statistics {
     description
       "Clears protocol statistics (e.g. sent and received
        counters).";
     input {
       leaf lsr-id {
         type union {
           type yang:dotted-quad;
           type uint32;
         }
         description
           "LSR ID of peer whose statistic are to be cleared.
            If this is not provided then all peers statistics are
            cleared";
       }
     }
   }

   /*
    * Notifications
```

```
    */
   notification mpls-ldp-peer-event {

     description
       "Notification event for a change of LDP peer operational
        status.";
     leaf event-type {
       type oper-status-event-type;
       description "Event type.";
     }
     uses ldp-peer-ref;
   }

   notification mpls-ldp-hello-adjacency-event {
     description
       "Notification event for a change of LDP adjacency operational
        status.";
     leaf event-type {
       type oper-status-event-type;
       description "Event type.";
     }
     uses ldp-adjacency-ref;
   }

   notification mpls-ldp-fec-event {
     description
       "Notification event for a change of FEC status.";
     leaf event-type {
       type oper-status-event-type;
       description "Event type.";
     }
     uses ldp-fec-event;
   }

   notification mpls-mldp-fec-event {
     description
       "Notification event for a change of FEC status.";
     leaf event-type {
       type oper-status-event-type;
       description "Event type.";
     }
     uses mldp-fec-event;
   }
 }

 <CODE ENDS>
```

Figure 22

7. Security Considerations

The configuration, state, action and notification data defined using YANG data models in this document are likely to be accessed via the protocols such as NETCONF [RFC6241] etc.

Hence, YANG implementations MUST comply with the security requirements specified in section 15 of [RFC6020]. Additionally, NETCONF implementations MUST comply with the security requirements specified in sections 2.2, 2.3 and 9 of [RFC6241] as well as section 3.7 of [RFC6536].

8. IANA Considerations

This document does not extend LDP or mLDP base protocol specifiction and hence there are no IANA considerations.

Note to the RFC Editor: Please remove IANA section before the publication.

9. Acknowledgments

The authors would like to acknowledge Eddie Chami, Nagendra Kumar, Mannan Venkatesan, Pavan Beeram for their contribution to this document. We also acknowledge Ladislav Lhotka for his useful comments as the YANG Doctor.

10. References

10.1. Normative References

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-22 (work in progress), July 2016.

[I-D.rtgyangdt-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., and D. Bogdanovic, "Network Instance Model", draft-rtgyangdt-rtgwg-ni-model-00 (work in progress), May 2016.

[I-D.saad-mpls-base-yang]
Raza, K., Gandhi, R., Liu, X., Beeram, V., Saad, T., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for MPLS Base", draft-saad-mpls-base-yang-00 (work in progress), May 2016.

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

[RFC3478] Leelanivas, M., Rekhter, Y., and R. Aggarwal, "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, DOI 10.17487/RFC3478, February 2003, <http://www.rfc-editor.org/info/rfc3478>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <http://www.rfc-editor.org/info/rfc5036>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <http://www.rfc-editor.org/info/rfc5331>.

[RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, DOI 10.17487/RFC5561, July 2009, <http://www.rfc-editor.org/info/rfc5561>.

[RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, DOI 10.17487/RFC5918, August 2010, <http://www.rfc-editor.org/info/rfc5918>.

[RFC5919] Asati, R., Mohapatra, P., Chen, E., and B. Thomas, "Signaling LDP Label Advertisement Completion", RFC 5919, DOI 10.17487/RFC5919, August 2010, <http://www.rfc-editor.org/info/rfc5919>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <http://www.rfc-editor.org/info/rfc6388>.

[RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, DOI 10.17487/RFC6389, November 2011, <http://www.rfc-editor.org/info/rfc6389>.

[RFC6512] Wijnands, IJ., Rosen, E., Napierala, M., and N. Leymann, "Using Multipoint LDP When the Backbone Has No Route to the Root", RFC 6512, DOI 10.17487/RFC6512, February 2012, <http://www.rfc-editor.org/info/rfc6512>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6826] Wijnands, IJ., Ed., Eckert, T., Leymann, N., and M. Napierala, "Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6826, DOI 10.17487/RFC6826, January 2013, <http://www.rfc-editor.org/info/rfc6826>.

[RFC7060] Napierala, M., Rosen, E., and IJ. Wijnands, "Using LDP Multipoint Extensions on Targeted LDP Sessions", RFC 7060, DOI 10.17487/RFC7060, November 2013, <http://www.rfc-editor.org/info/rfc7060>.

[RFC7140] Jin, L., Jounay, F., Wijnands, IJ., and N. Leymann, "LDP Extensions for Hub and Spoke Multipoint Label Switched Path", RFC 7140, DOI 10.17487/RFC7140, March 2014, <http://www.rfc-editor.org/info/rfc7140>.

[RFC7246] Wijnands, IJ., Ed., Hitchen, P., Leymann, N., Henderickx, W., Gulko, A., and J. Tantsura, "Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context", RFC 7246, DOI 10.17487/RFC7246, June 2014, <http://www.rfc-editor.org/info/rfc7246>.

[RFC7438] Wijnands, IJ., Ed., Rosen, E., Gulko, A., Joorde, U., and J. Tantsura, "Multipoint LDP (mLDP) In-Band Signaling with Wildcards", RFC 7438, DOI 10.17487/RFC7438, January 2015, <http://www.rfc-editor.org/info/rfc7438>.

[RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <http://www.rfc-editor.org/info/rfc7552>.

[RFC7715] Wijnands, IJ., Ed., Raza, K., Atlas, A., Tantsura, J., and Q. Zhao, "Multipoint LDP (mLDP) Node Protection", RFC 7715, DOI 10.17487/RFC7715, January 2016, <http://www.rfc-editor.org/info/rfc7715>.

10.2. Informative References

[I-D.ietf-rtgwg-policy-model] Shaikh, A., Shakir, R., D’Souza, K., and C. Chase, "Routing Policy Configuration Model for Service Provider Networks", draft-ietf-rtgwg-policy-model-01 (work in progress), April 2016.

[I-D.iwijnand-mpls-mldp-multi-topology] Wijnands, I. and K. Raza, "mLDP Extensions for Multi Topology Routing", draft-iwijnand-mpls-mldp-multi-topology-03 (work in progress), June 2013.

[I-D.openconfig-netmod-opstate] Shakir, R., Shaikh, A., and M. Hines, "Consistent Modeling of Operational State Data in YANG", draft-openconfig-netmod-opstate-01 (work in progress), July 2015.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[RFC7307] Zhao, Q., Raza, K., Zhou, C., Fang, L., Li, L., and D. King, "LDP Extensions for Multi-Topology", RFC 7307, DOI 10.17487/RFC7307, July 2014, <http://www.rfc-editor.org/info/rfc7307>.

Appendix A. Additional Contributors

Stephane Litkowski
Orange.
Email: stephane.litkowski@orange.com

Reshad Rahman
Cisco Systems Inc.
Email: rrahman@cisco.com

Danial Johari
Cisco Systems Inc.
Email: dajohari@cisco.com

Authors’ Addresses

Kamran Raza
Cisco Systems, Inc.
Email: skraza@cisco.com

Rajiv Asati
Cisco Systems, Inc.
Email: rajiva@cisco.com

Sowmya Krishnaswamy
Cisco Systems, Inc.
Email: sowkrish@cisco.com

Xufeng Liu
Ericsson
Email: xliu@kuatrotech.com

Jeff Tantsura
Ericsson
Email: jeff.tantsura@ericsson.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Xia Chen
Huawei Technologies
Email: jescia.chenxia@huawei.com

Loa Andersson
Huawei Technologies
Email: loa@pi.nu

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

Matthew Bocci
Alcatel-Lucent
Email: matthew.bocci@alcatel-lucent.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

X. Zhang, Ed.
Huawei Technologies
J. Ryoo
ETRI

October 31, 2016

A Service YANG Model for Connection-oriented Transport Networks
draft-zhang-teas-transport-service-model-01

Abstract

A transport network is a server-layer network designed to provide connectivity services for a client-layer network to carry the client traffic opaquely across the server-layer network resources. A transport network may be constructed from equipments utilizing any of a number of different transport technologies such as the evolving optical transport infrastructure (Synchronous Optical Networking (SONET) / Synchronous Digital Hierarchy (SDH) and Optical Transport Network (OTN)) or packet transport as epitomized by the MPLS Transport Profile (MPLS-TP).

This draft provides a transport service YANG model that can be used together with the RESTCONF protocol for a northbound client to initiate service requests toward the transport network controllers via the RESTful interface between them so as to enable automated service interations.

Requirements Language

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

Status of This Memo

This Internet-Draft will expire on May 4, 2017.

Table of Contents

1. Introduction

A transport network is a server-layer network designed to provide connectivity services, or more advanced services like Virtual Private Networks (VPN) for a client-layer network to carry the client traffic opaquely across the server-layer network resources. It acts as a pipe provider for upper-layer networks, such as IP network and mobile networks.

Transport networks, such as Synchronous Optical Networking (SONET) / Synchronous Digital Hierarchy (SDH), Optical Transport Network (OTN), Wavelength Division Multiplexing (WDM), and flexi-grid networks, are often built using equipments from a single vendor and are managed using private interfaces to dedicated Element Management Systems

(EMS) / Network Management Systems (NMS). All transport networks have high benchmarks for reliability and operational simplicity. This suggests a common, technology-independent management/control paradigm that is extended to represent and configure specific technology attributes.

The YANG language [RFC6020] is currently the data modeling language of choice within the IETF and has been adopted by a number of industry-wide open management and control initiatives. YANG may be used to model both configuration and operational states; it is vendor-neutral and supports extensible APIs for control and management of elements.

This document, exploting the YANG language, provides a transport service model that can be used by a northbound client (e.g., an orchestrator, a client network controller etc.) to initiate service requests, as well as retrieving service states, toward the transport network via the RESTful interface provided by the transport network controller(s). The model is currently scoped for connection-oriented transport networks and connection-less transport networks are out of scope. To understand better what interfaces a service model can apply to in the transport SDN architecture (i.e., Abstract Control of Transport Networks (ACTN))and how service model relates to other YANG models defined in IETF in general, please refer to [I-D.zhang-teas-actn-yang] for more detailed information.

2. Terminology

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [I-D.ietf-netmod-rfc6087bis]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

3. Service Model - YANG Tree

The service model YANG tree is provided below:

```
module: ietf-transport-service
   +--rw transport_service
      +--rw service* [service-name]
         +--rw service-name    -> ../config/service-name
         +--rw config
         |  +--rw service-name?        string
         |  +--rw service-id?          uint32
         |  +--rw service-endpoints* [node-id tp-id]
         |  |  +--rw type?            enumeration
         |  |  +--rw node-id          union
         |  |  +--rw tp-id            uint32
         |  |  +--rw endpoint-name?   string
         |  +--rw service-type         identityref
         |  +--rw supporting-tunnel
         |  |  +--rw tunnel-name?   string
         |  +--rw bandwidth?           decimal64
         |  +--rw protection-type?     identityref
         |  +--rw constraints
         |     +--rw delay-limit?            uint32
         |     +--rw delayvariation-limit?   uint32
         |     +--rw packetloss-limit?       decimal64
         |     +--rw objective?              identityref
         +--ro state
            +--ro service-name?        string
            +--ro service-id?          uint32
            +--ro service-endpoints* [node-id tp-id]
            |  +--ro type?            enumeration
            |  +--ro node-id          union
            |  +--ro tp-id            uint32
            |  +--ro endpoint-name?   string
            +--ro service-type         identityref
            +--ro supporting-tunnel
            |  +--ro tunnel-name?   string
            +--ro bandwidth?           decimal64
            +--ro protection-type?     identityref
            +--ro creation-time?       ytypes:date-and-time
            +--ro constraints
            |  +--ro delay-limit?            uint32
            |  +--ro delayvariation-limit?   uint32
            |  +--ro packetloss-limit?       decimal64
            |  +--ro objective?              identityref
            +--ro performance-info
            |  +--ro delay?            uint32
            |  +--ro delay-variation?   uint32
            |  +--ro packet-loss?       decimal64
            +--ro oper-status          tet:te-oper-status
```

4. Service Model - YANG Code

```
 <CODE BEGINS>file "ietf-transport-service@2016-10-31.yang"
module ietf-transport-service {
        yang-version 1;
        namespace "urn:ietf:params:xml:ns:yang:ietf-transport-service";
        prefix transservice;

        import ietf-inet-types {
                prefix inet;
        }

        import ietf-yang-types {
          prefix ytypes;
        }

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



        organization "Internet Engineering Task Force";
        contact
                "Editor: Xian ZHANG (zhang.xian@huawei.com)";
        description
                "this module describes a service module that is an essential
                API for a client to ask for a provider network for a path
                without the need to care about underlying technologies.
                Capability to specify constraints/policies are provided as
                optional features.";

        revision 2016-10-31 {
                description
                        "Initial revision.";
                        reference "to add the draft name";
        }

        grouping endpoints-grouping{
                description "end point grouping";
                leaf type {
                        type enumeration {
                                enum root {
                                        description "root";
                                }
                                enum leaf {
```

```
                                description "leaf";
                        }
                        enum unknown {
                                description "unknown";
                        }
                }
        default root;

        description
                "This attribute indicates whether a endpoint of the
                service is acting as a root or leaf, and takes on
                the value either root, leaf or unkown. If the service
                type is Point-to-Point or Multipoint-to-Multipoint,
                then the type is equal to root. If the service type
                is Point-to-Multipoint, then the type can be either
                root or leaf.";
        }


        leaf node-id {
                type union {
                        type inet:ip-address; //IPv4 or IPv6
                        type uint32;
                }
                description
                        "the node address of a service .";
        }

        leaf tp-id {
                type uint32;
                description
                        "the termination point of a service.";
        }

        leaf endpoint-name {
                type string;
                description
                        "the name of this service endpoint";
        }
}

identity service-type {
        description
                "base identity from which the specific service
                type can be derived.";
}

identity service-P2P {
```

```
                base service-type;
                description
                        "a point-to-point service type";
        }

        identity service-P2MP {
                base service-type;
                description
                        "a point to multi-point type";
        }

        identity service-MP2MP {
                base service-type;
                description
                        "a multi-point to multi-point type";
        }

//note: service type such as EPL, EVPL, EPLAN
//      EVPLAN, E-Tree can be added when augmenting
//      this model.

        identity service-prot-type {
                description
                        "base identity from which the specific
                         protection type can be derived.";
        }

        identity service-prot-unprotected {
                base service-prot-type;
                description
                        "service protection type: unprotected";
        }

        identity service-prot-1plusR {
                base service-prot-type;
                description
                        "service protection type: dynamic reroute";
                reference
                        "draft-ietf-teas-gmpls-resource-sharing-proc-04";
        }

        identity service-prot-1plus1 {
                base service-prot-type;
                description
                        "service protection type: static 1+1";
        }

        identity service-prot-1plus1plusR {
```

```
                base service-prot-type;
                description
                        "service protection type: 1+1+R";
                reference
                        "draft-ietf-teas-gmpls-resource-sharing-proc-04";
        }

        identity objective-type{
                description
                        "base identity from which the specific obective
                        funciton can be derived.";
        }
        identity objective-mindelay{
                base objective-type;
                description
                        "objective: minimized delay";
        }
        identity objective-mincost {
                base objective-type;
                description
                        "objective: minimized cost";
        }

        identity objective-mindistance {
                base objective-type;
                description
                        "objective: minimized distance";
        }

        grouping service-config-grouping {
                description "service-config-groupoing";

                leaf service-name{
                        type string;
                        description "the name for a service";
                }

                leaf service-id {
                        type uint32;
                        description
                                "an unique identificaiton of a service.";
                }

                list service-endpoints{
                        key "node-id tp-id";
                        description
                                "this provides the service endpoints and it can
be used
                                to support different types of services (includin
g P2P,
```

```
                                MP2MP, P2MP)";

                        uses endpoints-grouping;
                }

                leaf service-type {
                        type identityref {
                                base service-type;
                        }
                        mandatory true;
                        description
                                "the type of a service request";
                }

                container supporting-tunnel {
                        leaf tunnel-name{
                                type string;
                                description "the name of a tunnel (unique)";
                        }

                        description
                                "the tunnels that can be used to support the ser
vice";
                }

                leaf bandwidth {
                        type decimal64 {
                                fraction-digits 2;
                        }
                        description "the bandwidth requested by a service.";
                }

                leaf protection-type{
                        type identityref {
                                base service-prot-type;
                        }
                        description
                        "the type of protection expected for this
                        service";
                }
        }

        grouping constraints-grouping {
                leaf delay-limit {
                        type uint32;
                        description
                        "Delay variation in micro-seconds.";
                }
```

```
                leaf delayvariation-limit {
                        type uint32;
                        description
                        "Delay variation.";
                }

                leaf packetloss-limit {
                        type decimal64 {
                                fraction-digits 6;
                                range "0 .. 50.331642";
                        }
                        description
                        "Delay variation in %.";
                }
                leaf objective {
                        type identityref
                        {
                                base objective-type;
                        }
                        description
                                "objective function imposed for path computation
                                from this service.";
                }
                description "constraints gropuing";
        }

        container transport_service {
                description
                        "serves as a top-level container for a list of services"
;

                list service {
                        key "service-name";
                        description
                                "an unique identifier of a service";

                        leaf service-name {
                                type leafref {
                                        path "../config/service-name";
                                }
                                description "a unique service identifier.";
                        }
                        container config{
                                description "service config container";

                                uses service-config-grouping;

                                container constraints{
                                        uses constraints-grouping;
```

```
                                description
                                        "specify the constraints imposed
 by a
                                        service";
                        }
                }

                container state
                {
                        config false;
                        description "operational state of a service";

                        uses service-config-grouping;

                        leaf creation-time {
                                type ytypes:date-and-time;
                                description
                                        "the time when service is create
d";
                        }

                        container constraints{
                                uses constraints-grouping;
                                description
                                        "specify the constraints imposed
 by a
                                        service";
                        }

                        container performance-info{
                                description "service state-performance i
nfo";
                                leaf delay{
                                        type uint32;
                                        description "the latency of this
 service";
                                }

                                leaf delay-variation {
                                        type uint32;
                                        description "the service delay v
aration";
                                }
                                leaf packet-loss {
                                        type decimal64 {
                                                fraction-digits 6;
                                                range "0 .. 50.331642";
                                        }
                                        description "packet loss";
                                }
                        }
                        leaf oper-status {
                                type tet:te-oper-status;
                                mandatory true;
```

```
                                        description
                                                "the operational status of this
service";
                                }
                        }//end of state of a service
                }//end of service list (including config and state)
        }//service top container
}
```

<CODE ENDS>

5. General Considerations and Some Open Issues

[I-D.liu-netmod-yang-schedule] defines a YANG data model for configuration scheduling. A service model can also be scheduled, therefore the ietf-schedule model can be used to serve this purpose by specifying the appropriate object using xpath.

Functions that are under considerations for updating the YANG model:

1): Service delivery model augmentation: Information such as PIR/CIR pertaining to ETH service is needed and it is more appropriate to augment the base model; for OTN-based service model, signal-type information needs to be added;

2): To add service-related notification, e.g.: service up; service down, service degraded etc.; Currently, there are two options available, one is using the method provided in [RFC5277], the other one is using the new mechanism proposed in [I-D.ietf-netconf-yang-push].

3): To figure out how to deal with attributes such as bandwidth, schedule where it can be per node for non-P2P, but is per node pair for P2P;

4): to add XRO/IRO constraints;

6. IANA Considerations

TBD.

7. Manageability Considerations

TBD

8. Security Considerations

Since the data model defined in this draft is manipulated via, for example, the interface between an orchestrator and a transport network controller. The YANG module defined in this draft is designed to be accessed via the RESTCONF protocol defined in [I-D.ietf-netconf-restconf], or maybe via the NETCONF protocol [RFC6241]. As mentioned in Security Conserations Section of [I-D.ietf-netconf-restconf] that HTTPS defined in [RFC7230] can be used to provide security while accessing data.

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

Editor note: to List specific subtrees and data nodes and their sensitivity/vulnerability.

9. Acknowledgements

We would like to thank Baoquan Rao and Gang Xie for their initial discussions that trigger the generation of this draft. We would also like to thank Michael Scharf from Nokia for very useful comments to this draft and the model.

Aspects of the work in this Internet-Draft is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) under the TOUCAN project (EP/L020009/1).

10. Contributors

Zhe Liu
Huawei Technologies
Email: liuzhe123@huawei.com

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

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

11. References

11.1. Normative References

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-13 (work in progress), April 2016.

[I-D.ietf-netmod-rfc6087bis]
Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-06 (work in progress), March 2016.

[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-02 (work in progress), October 2016.

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

11.2. Informative References

[I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "Subscribing to YANG datastore push updates", draft-ietf-netconf-yang-push-04 (work in progress), October 2016.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-03 (work in progress), March 2016.

[I-D.zhang-teas-actn-yang]
Lee, Y., Zhang, X., Yoon, B., Dios, O., and J. Shin, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-zhang-teas-actn-yang-02 (work in progress), October 2016.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <http://www.rfc-editor.org/info/rfc5277>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

Authors' Addresses

Xian Zhang (editor)
Huawei Technologies
F3-5-B R&D Center, Huawei Industrial Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P.R.China

Email: zhang.xian@huawei.com

Jeong-dong Ryoo
ETRI

Email: ryoo@etri.re.kr