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
```
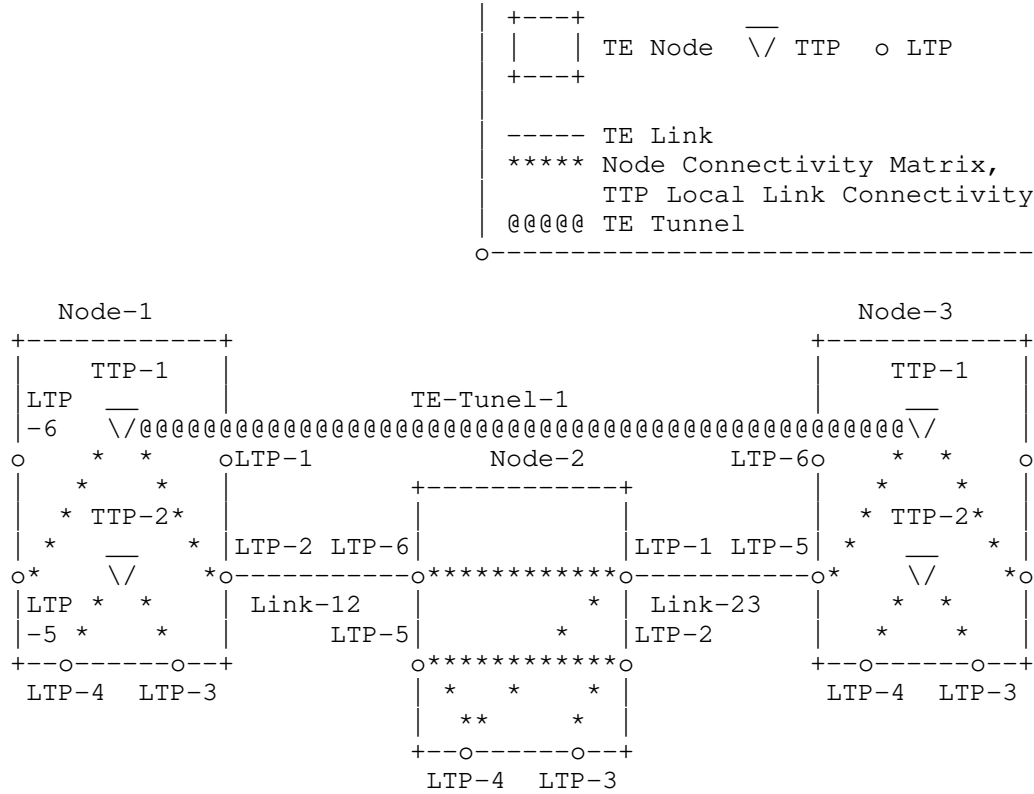
              Figure 1: TE Topology Modeling Abstractions

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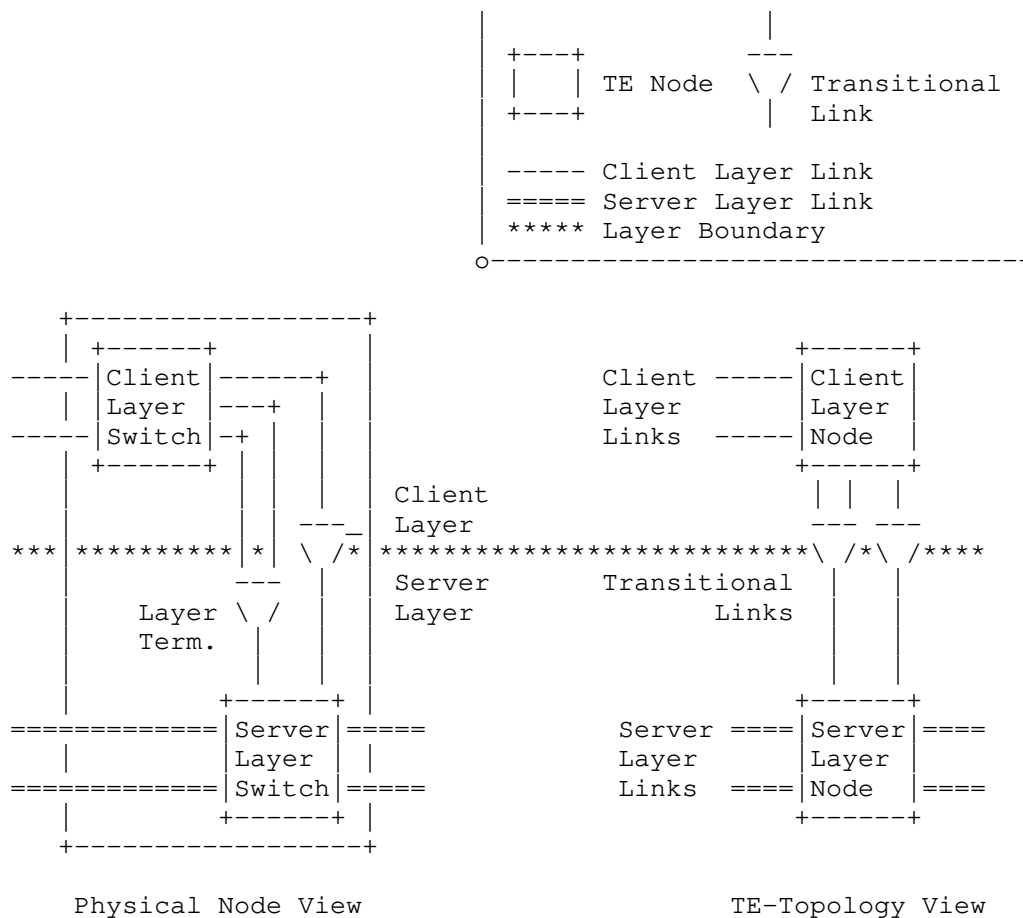
                Figure 2: Modeling a Multi-Layer Node (Dual-Layer Example)

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
```
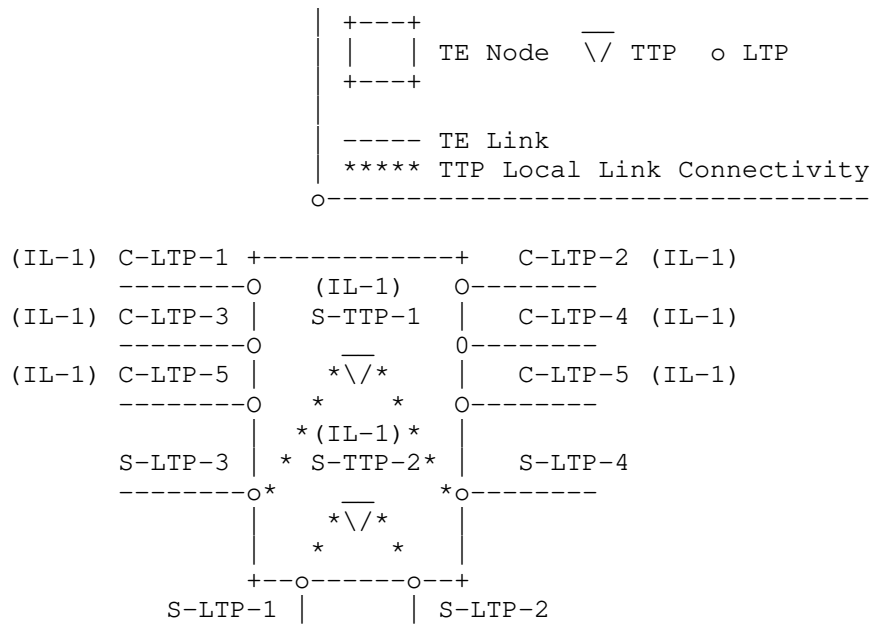
               Figure 3: TE Inter-Layer Lock ID Associations

   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.
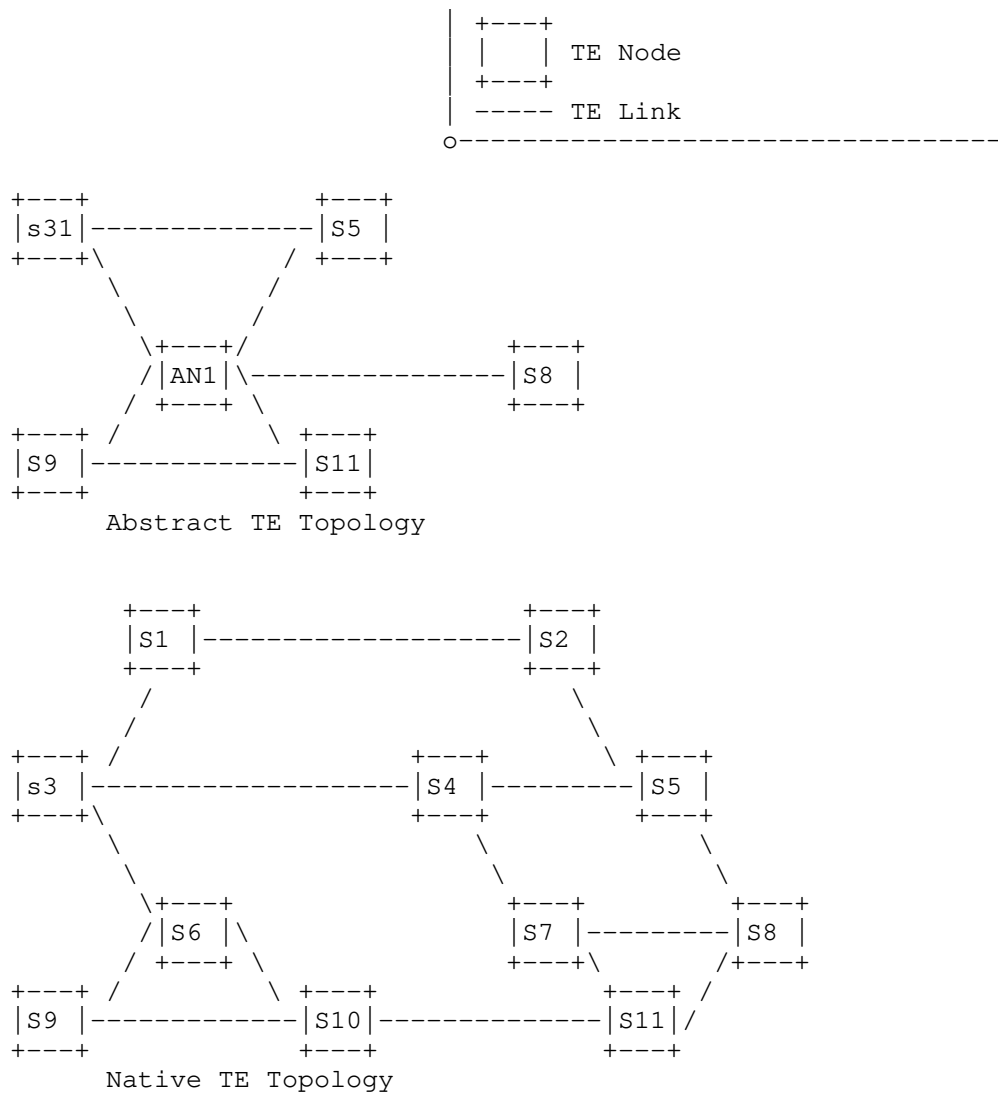
```
                         |  +---+
                         |  |   |  TE Node
                         |  +---+
                         |  -----  TE Link
                         o-------------------------------

  +---+                    +---+
  |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|
  +---+          +---+          +---+          +---+
```
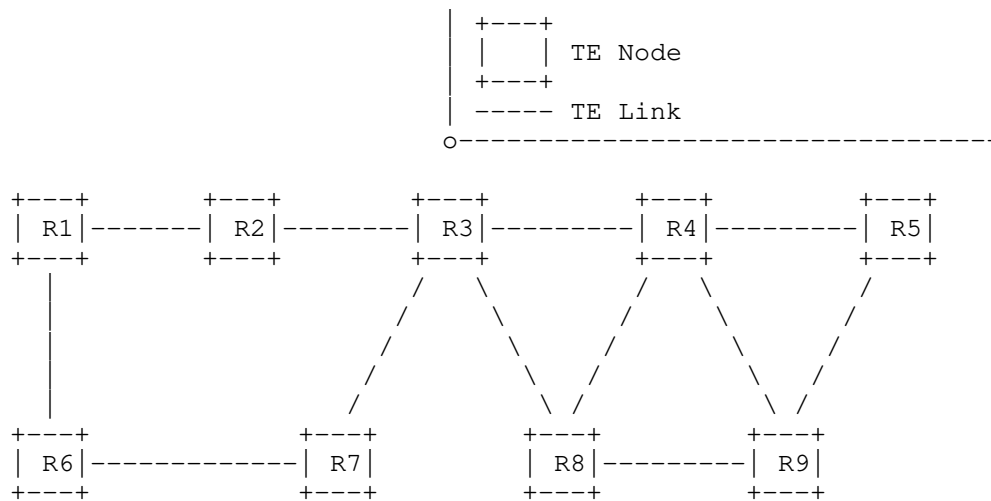
                 Figure 5a: Example Network Topology

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.

```
         ---------------
        |  Native       |                      |  [ ] TE Node
        |  TE-Topology  |                      |  +++ TE Link
         ---------------                       o-------------

   [R1] ++++ [R2] ++++ [R3] ++++ [R4] ++++ [R5]
     +                  +     +    +    +    +
     +                  +     +  +      +  +
     +                  +       ++       ++
   [R6] +++++++++ [R7]      [R8]  ++++ [R9]
```
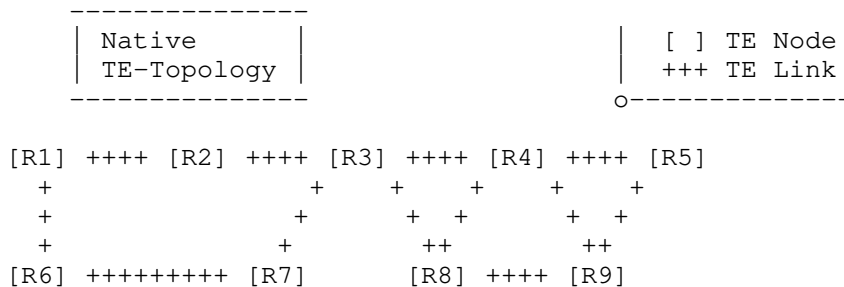
              Figure 5b: Native TE Topology as seen on Node R3

Consider the case of the topology being split in a way that some
nodes participate in OSPF-TE while others participate in ISIS-TE
(Figure 6a). An implementation MAY choose to construct separate TE
Topologies based on the information source. The native TE Topologies
constructed using only nodes and links that were learnt via a
specific information source are depicted in Figure 6b. The data model
proposed in this document can be used to retrieve/represent these TE
topologies.

Similarly, the data model can be used to represent/retrieve a TE
Topology that is constructed using only nodes and links that belong
to a particular technology layer. The data model is flexible enough
to retrieve and represent many such native TE Topologies.

```
                                 :
TE info distributed via ISIS-TE  :  TE info distributed via OSPF-TE
                                 :
   +---+        +---+        +---+        +---+        +---+
   | R1|-------| R2|--------| R3|--------| R4|--------| R5|
   +---+        +---+        +---+        +---+        +---+
     |              / : \      / \         /
     |             / : \    / \       /
     |            / : \  / \     /
     |           / : \ /   \   /
     |          / : \ /     \ /
   +---+        +---+  :   +---+      +---+
   | R6|-------------| R7|  :   | R8|--------| R9|
   +---+        +---+  :   +---+      +---+
                                 :
```
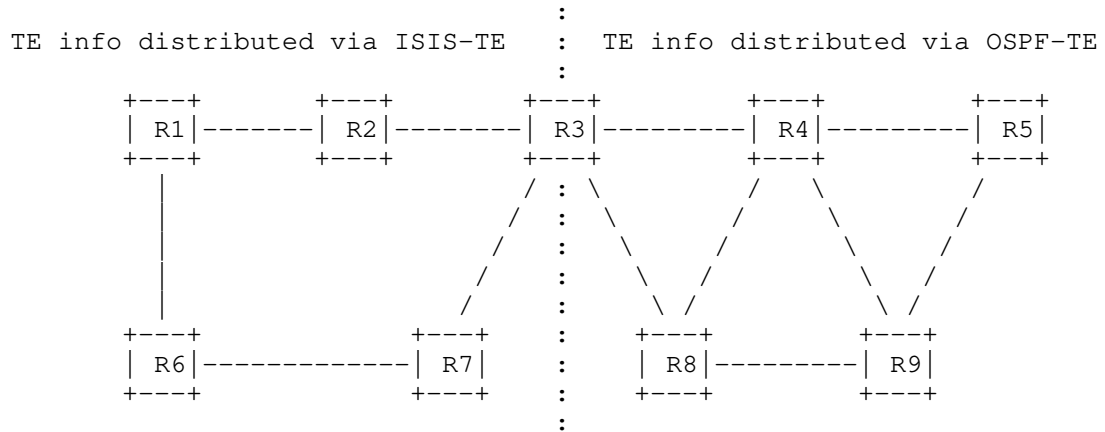
                Figure 6a: Example Network Topology

```
   ---------------------   :  ---------------------
   |Native TE Topology  |   :  |Native TE Topology  |
   |Info-Source: ISIS-TE|   :  |Info-Source: OSPF-TE|
   ---------------------   :  ---------------------
                                 :
    [R1] ++++ [R2] ++++ [R3] : [R3'] ++++ [R4] ++++ [R5]
     +              +     :    +    +    +    +
     +              +     :     +  +      +  +
     +              +     :      ++         ++
    [R6] ++++++++ [R7]    :       [R8]  ++++ [R9]
```
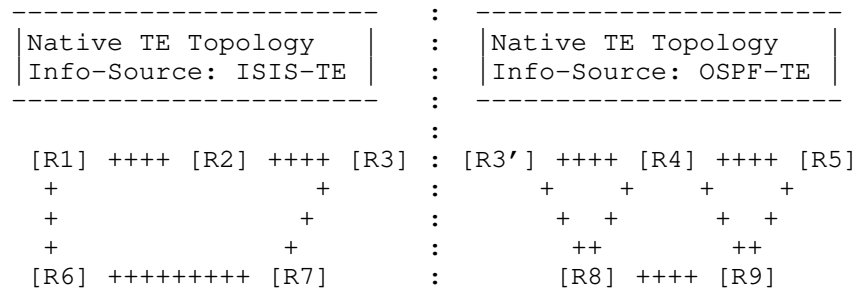
            Figure 6b: Native TE Topologies as seen on Node R3

4.2. Customized TE Topologies

Customized TE topology is a topology that was modified by the
provider to honor a particular client's requirements or preferences.
The model discussed in this draft can be used to represent, retrieve
and manipulate customized TE Topologies. The model allows the
provider to present the network in abstract TE Terms on a per client

basis. These customized topologies contain sufficient information for
the path computing client to select paths according to its policies.

```
                        |  +---+                /-\
                        |  |   |  Router      (   ) WDM
                        |  +---+  Node          \-/  node
                        |
                        o---------------------------

  +---+      /-\          /-\            /-\          +---+
  | R1|-------( A )--------( C )---------( E )---------| R3|
  +---+      \-/          \-/            \-/          +---+
                         /  \          /   \
                        /    \        /     \
                       /      \      /       \
                      /        \    /         \
                     /          \  /           \
  +---+      /-\    /   /-\       \/   /-\        /-\    +---+
  | R2|---------( B )---------( D )---------( F )---------| R4|
  +---+      \-/          \-/            \-/          +---+
```
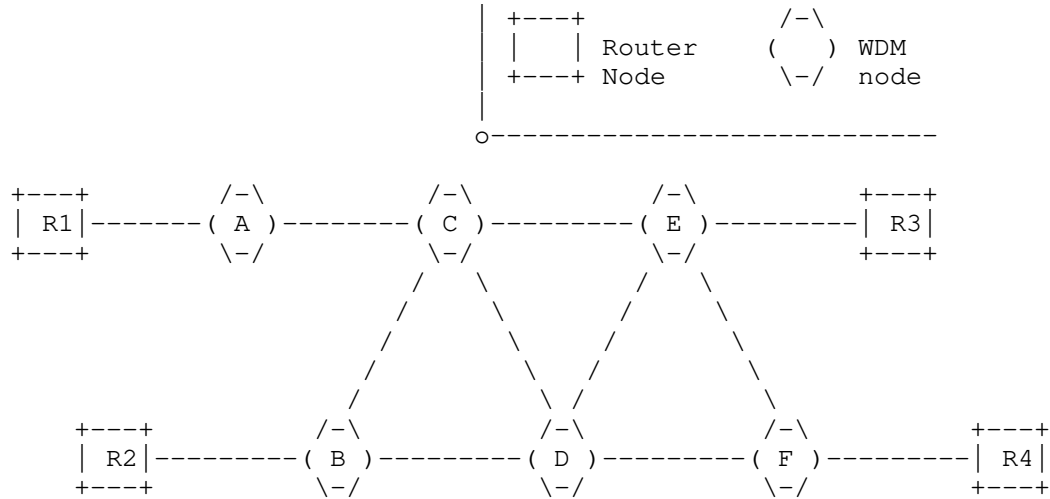
                Figure 7: Example packet optical topology

Consider the network topology depicted in Figure 7. This is a typical
packet optical transport deployment scenario where the WDM layer
network domain serves as a Server Network Domain providing transport
connectivity to the packet layer network Domain (Client Network
Domain). Nodes R1, R2, R3 and R4 are IP routers that are connected to
an Optical WDM transport network. A, B, C, D, E and F are WDM nodes
that constitute the Server Network Domain.

```
                          |  *****   B-F WDM Path
                          |  @@@@@   B-E WDM Path
                          |  $$$$$   A-E WDM Path
                          o-------------------

  +---+          /-\ $$$$$$$$ /-\ $$$$$$$$$ /-\          +---+
  | R1|-------( A )--------( C )---------( E )---------| R3|
  +---+      \-/          @\-/ @@@@@@@@@ \-/          +---+
                        @/       \          /    \
                       @/         \        /      \
                      @/           \      /        \
                     @/             \    /          \
                    @/               \  /            \
  +---+          /-\ ********* /-\ ********* /-\          +---+
  | R2|---------( B )---------( D )---------( F )---------| R4|
  +---+      \-/          \-/            \-/          +---+
```
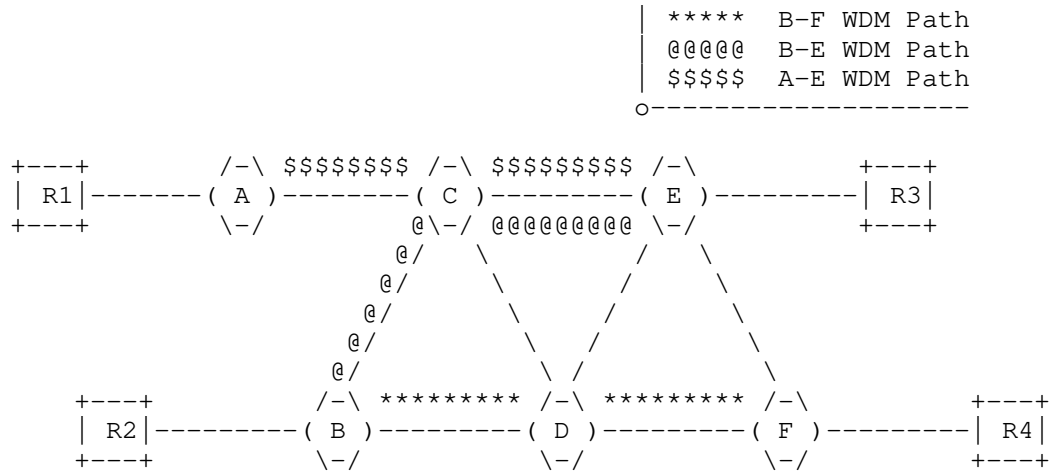
                  Figure 8a: Paths within the provider domain

        ++++++++ [A] +++++++++++++++++++ [E] +++++++++
                                     +++++
                                   ++++
                                 ++++
                               ++++
                             ++++
        ++++++++ [B] +++++++++++++++++++ [F] +++++++++

           Figure 8b: Customized TE Topology provided to the Client

   The goal here is to augment the Client TE Topology with a customized
   TE Topology provided by the WDM network. Given the availability of
   the paths A-E, B-F and B-E (Figure 8a), a customized TE Topology as
   depicted in Figure 8b is provided to the Client. This customized TE
   Topology is merged with the Client's Native TE Topology and the
   resulting topology is depicted in Figure 8c.

       [R1] ++++++++ [A] +++++++++++++++++++ [E] +++++++++ [R3]
                                         +++++
                                       ++++
                                     ++++
                                   ++++
                                 ++++
       [R2] ++++++++ [B] +++++++++++++++++++ [F] +++++++++ [R4]

     Figure 8c: Customized TE Topology merged with the Client's Native TE
                                  Topology

   The data model proposed in this document can be used to
   retrieve/represent/manipulate the customized TE Topology depicted in
   Figure 8b.

   A customized TE topology is not necessarily an abstract TE topology.
   The provider may produce, for example, an abstract TE topology of
   certain type (e.g. single-abstract-node-with-connectivity-matrix
   topology, a border-nodes-connected-via-mesh-of-abstract-links
   topology, etc.) and expose it to all/some clients in expectation that
   the clients will use it without customization.
   On the other hand, a client may request a customized version of the
   provider's native TE topology (e.g. by requesting removal of TE links

which belong to certain layers, are too slow, not protected and/or
have a certain affinity). Note that the resulting TE topology will
not be abstract (because it will not contain abstract elements), but
customized (modified upon client's instructions).

The client ID field in the TE topology identifier (Section 5.4.)
indicates which client the TE topology is customized for. Although an
authorized client MAY receive a TE topology with the client ID field
matching some other client, the client can customize only TE
topologies with the client ID field either 0 or matching the ID of
the client in question. If the client starts reconfiguration of a
topology its client ID will be automatically set in the topology ID
field for all future configurations and updates wrt. the topology in
question.

The provider MAY tell the client that a given TE topology cannot be
re-negotiated, by setting its own (provider's) ID in the client ID
field of the topology ID.

Even though this data model allows to access TE topology information
across clients, implementations MAY restrict access for particular
clients to particular data fields. The Network Configuration Access
Control Model (NACM) [RFC8341] provides such a mechanism.

4.3. Merging TE Topologies Provided by Multiple Providers

A client may receive TE topologies provided by multiple providers,
each of which managing a separate domain of multi-domain network. In
order to make use of said topologies, the client is expected to merge
the provided TE topologies into one or more client's native TE
topologies, each of which homogeneously representing the multi-domain
network. This makes it possible for the client to select end-to-end
TE paths for its services traversing multiple domains.

In particular, the process of merging TE topologies includes:

- Identifying neighboring domains and locking their topologies
  horizontally by connecting their inter-domain open-ended TE links;
- Renaming TE node, link, and SRLG IDs to ones allocated from a
  separate name space; this is necessary because all TE topologies
  are considered to be, generally speaking, independent with a
  possibility of clashes among TE node, link or SRLG IDs;
- Locking, vertically, TE topologies associated with different layer
  networks, according to provided topology inter-layer locks; this is
  to facilitate inter-layer path computations across multiple TE
  topologies provided by the same topology provider.

```
  /---\       +---+    +---+      +---+    +---+      /---\
  |s3 |------|S13|----|S15|------|S23|----|S25|------|C21|
  \---/       +---+\   +---+      +---+   /+---+      \---/
                    \                         /
                     \                       /
                      \+---+      +---+/    +---+      /---\
                       |S18|------|S24|     |S28|------|C22|
                       +---+      +---+\   /+---+      \---/
                                       \ /
                                       / \
  /---\       +---+    +---+      +---+/  \+---+      /---\
  |C12|------|S19|----|S17|------|S29|----|S27|------|C23|
  \---/       +---+    +---+      +---+    +---+      \---/


       Domain 1 TE Topology            Domain 2 TE Topology

       +---+    +---+                    +---+    +---+
   -----|S13|----|S15|----            ----|S23|----|S25|----
       +---+\   +---+                    +---+   /+---+
            \                                   /
             \                                 /
              \+---+                     +---+/    +---+
               |S18|----             ----|S24|     |S28|----
               +---+                     +---+\   /+---+
                                              \ /
                                              / \
       +---+    +---+                    +---+/  \+---+
   -----|S19|----|S17|----            ----|S29|----|S27|----
       +---+    +---+                    +---+    +---+
```
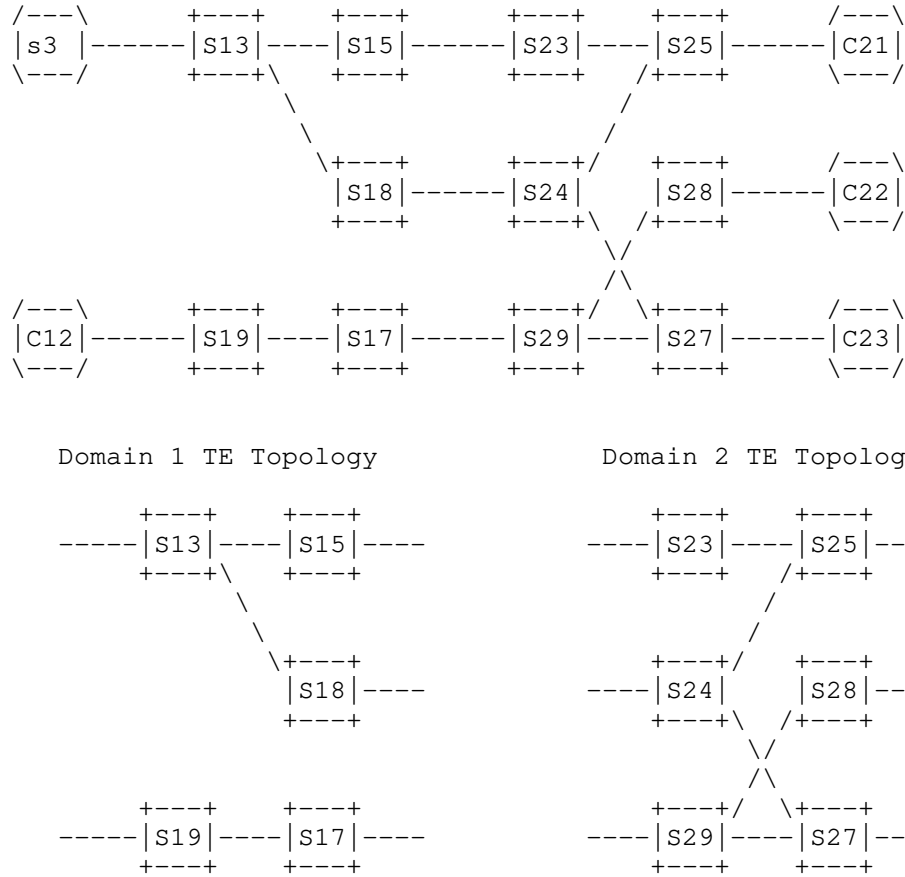
              Figure 9: Merging Domain TE Topologies

   Figure 9 illustrates the process of merging, by the client, of TE
   topologies provided by the client's providers. In the Figure, each of
   the two providers caters to the client (abstract or native) TE
   topology, describing the network domain under the respective
   provider's control. The client, by consulting the attributes of the
   inter-domain TE links - such as inter-domain plug IDs or remote TE
   node/link IDs (as defined by the TE Topology model) - is able to
   determine that:

   a) the two domains are adjacent and are inter-connected via three
      inter-domain TE links, and;

b) each domain is connected to a separate customer site, connecting
   the left domain in the Figure to customer devices C-11 and C-12,
   and the right domain to customer devices C-21, C-22 and C-23.

Therefore, the client inter-connects the open-ended TE links, as
shown on the upper part of the Figure.

As mentioned, one way to inter-connect the open-ended inter-domain TE
links of neighboring domains is to mandate the providers to specify
remote nodeID/linkID attribute in the provided inter-domain TE links.
This, however, may prove to be not flexible. For example, the
providers may not know the respective remote nodeIDs/ linkIDs. More
importantly, this option does not allow for the client to mix-n-match
multiple (more than one) topologies catered by the same providers
(see below). Another, more flexible, option to resolve the open-ended
inter-domain TE links is by annotating them with the inter-domain
plug ID attribute. Inter-domain plug ID is a network-wide unique
number that identifies on the network a connectivity supporting a
given inter-domain TE link. Instead of specifying remote node ID/link
ID, an inter-domain TE link may provide a non-zero inter-domain plug
ID. It is expected that two neighboring domain TE topologies
(provided by separate providers) will have each at least one open-
ended inter-domain TE link with an inter-domain plug ID matching to
one provided by its neighbor. For example, the inter-domain TE link
originating from node S15 of the Domain 1 TE topology (Figure 9) and
the inter-domain TE link coming from node S23 of Domain 2 TE topology
may specify matching inter-domain plug ID (e.g. 175344). This allows
for the client to identify adjacent nodes in the separate neighboring
TE topologies and resolve the inter-domain TE links connecting them
regardless of their respective nodeIDs/linkIDs (which, as mentioned,
could be allocated from independent name spaces). Inter-domain plug
IDs may be assigned and managed by a central network authority.
Alternatively, inter-domain plug IDs could be dynamically auto-
discovered (e.g. via LMP protocol).

Furthermore, the client renames the TE nodes, links and SRLGs offered
in the abstract TE topologies by assigning to them IDs allocated from
a separate name space managed by the client. Such renaming is
necessary, because the two abstract TE topologies may have their own
name spaces, generally speaking, independent one from another; hence,
ID overlaps/clashes are possible. For example, both TE topologies
have TE nodes named S7, which, after renaming, appear in the merged
TE topology as S17 and S27, respectively.

Once the merging process is complete, the client can use the merged
TE topology for path computations across both domains, for example,
to compute a TE path connecting C-11 to C-23.

4.4. Dealing with Multiple Abstract TE Topologies Provided by the Same
     Provider

```
       Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1


              +---+     +---+                  +---+     +---+
         -----|S13|----|S15|----         ----|S23|----|S25|----
              +---+\    +---+                  +---+   /+---+
                    \                                 /
                     \                               /
                      \+---+                   +---+/    +---+
                       |S18|----         ----|S24|     |S28|----
                       +---+                   +---+\  /+---+
                                                     \/
                                                     /\
              +---+     +---+                  +---+/  \+---+
         -----|S19|----|S17|----         ----|S29|----|S27|----
              +---+     +---+                  +---+     +---+



       Domain 1 Abstract TE Topology 1   Domain 2 Abstract TE Topology 1

              +-----------+                    +-----------+
         -----|           |----          ----|           |----
              |           |----          ----|           |----
              |    AN1    |----          ----|    AN1    |----
              |           |                  |           |
         -----|           |----          ----|           |----
              +-----------+                    +-----------+
```
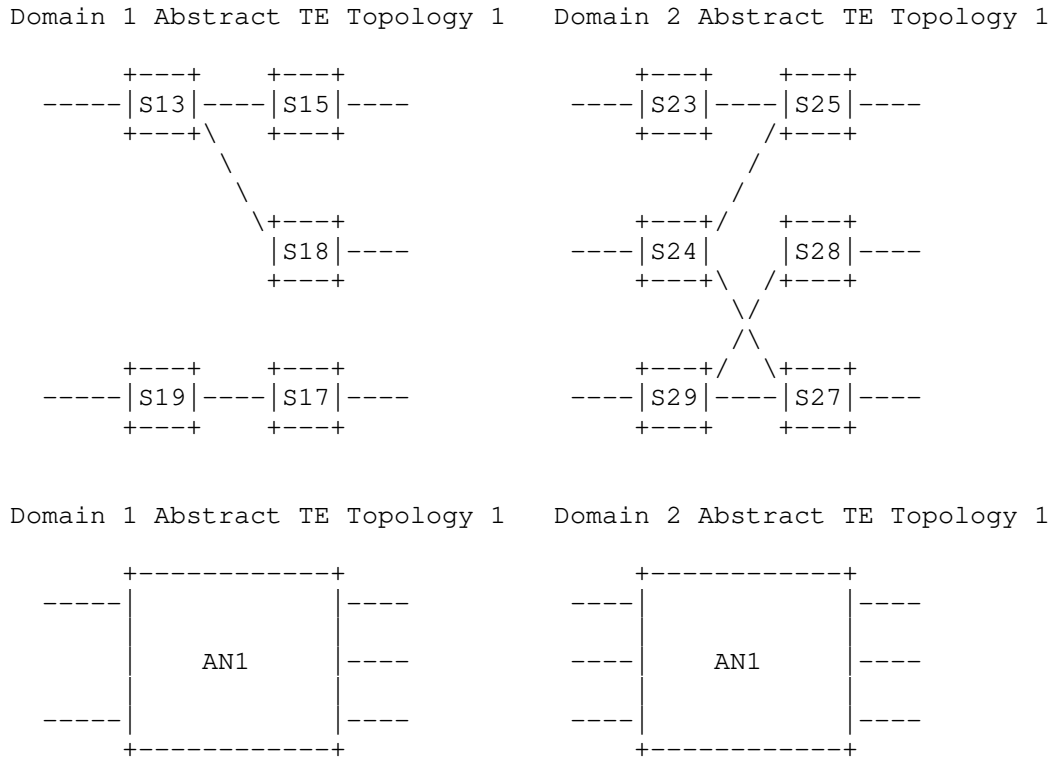
              Figure 10: Merging Domain TE Topologies

   Based on local configuration, templates and/or policies pushed by the
   client, a given provider may expose more than one abstract TE
   topology to the client. For example, one abstract TE topology could
   be optimized based on a lowest-cost criterion, while another one
   could be based on best possible delay metrics, while yet another one
   could be based on maximum bandwidth availability for the client
   services. Furthermore, the client may request all or some providers
   to expose additional abstract TE topologies, possibly of a different
   type and/or optimized differently, as compared to already-provided TE
   topologies. In any case, the client should be prepared for a provider
   to offer to the client more than one abstract TE topology.

   It should be up to the client (based on the client's local
   configuration and/or policies conveyed to the client by the client's

clients) to decide how to mix-and-match multiple abstract TE
topologies provided by each or some of the providers, as well as how
to merge them into the client's native TE topologies. The client also
decides how many such merged TE topologies it needs to produce and
maintain. For example, in addition to the merged TE topology depicted
in the upper part of Figure 9, the client may merge the abstract TE
topologies received from the two providers, as shown in Figure 10,
into the client's additional native TE topologies, as shown in Figure
11.

Note that allowing for the client mix-n-matching of multiple TE
topologies assumes that inter-domain plug IDs (rather than remote
nodeID/linkID) option is used for identifying neighboring domains and
inter-domain TE link resolution.

                   Client's Merged TE Topology 2

```
/---\      +-----------+      +-----------+      /---\
|s3 |------|           |------|           |------|C21|
\---/      |           |      |           |      \---/
           |           |      |           |
           |           |      |           |      /---\
           |    AN11   |------|    AN21   |------|C22|
           |           |      |           |      \---/
           |           |      |           |
/---\      |           |      |           |      /---\
|C12|------|           |------|           |------|C23|
\---/      +-----------+      +-----------+      \---/
```

                   Client's Merged TE Topology 3

```
/---\      +-----------+      +---+     +---+      /---\
|s3 |------|           |------|S23|----|S25|------|C21|
\---/      |           |      +---+    /+---+      \---/
           |           |              /
           |           |             /
           |           |      +---+/    +---+      /---\
           |    AN11   |------|S24|     |S28|------|C22|
           |           |      +---+\   /+---+      \---/
           |           |            \ /
           |           |             \
           |           |             /\
/---\      |           |      +---+/  \+---+      /---\
|C12|------|           |------|S29|----|S27|------|C23|
\---/      +-----------+      +---+     +---+      \---/
```

        Figure 11: Multiple Native (Merged) Client's TE Topologies
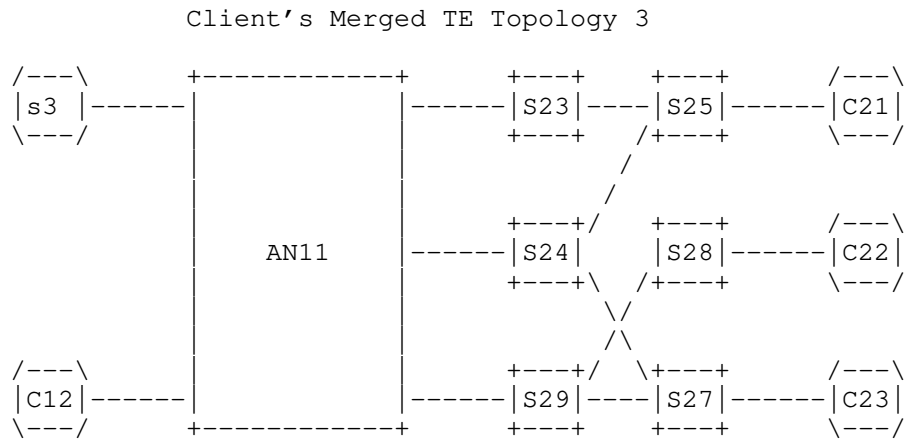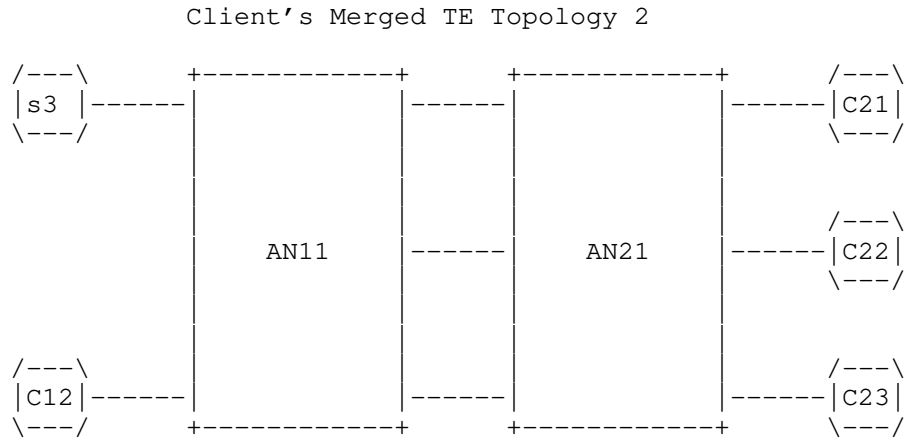
   It is important to note that each of the three native (merged) TE
   topologies could be used by the client for computing TE paths for any
   of the multi-domain services. The choice as to which topology to use
   for a given service depends on the service parameters/requirements
   and the topology's style, optimization criteria and the level of
   details.

5. Modeling Considerations

5.1. Network topology building blocks

   The network topology building blocks are discussed in [RFC8345]. The
   TE Topology model proposed in this document augments and uses the
   ietf-network-topology module defined in [RFC8345].

```
            +-----------------------+
            |                       |
            |  Network Topology Model |
            |  (ietf-network-topology)|
            +-----------------------+
                        |
                        |
                        |
                        V
            +-----------------------+
            |       TE Topology      |
            |         Model          |
            |                       |
            +-----------------------+
```
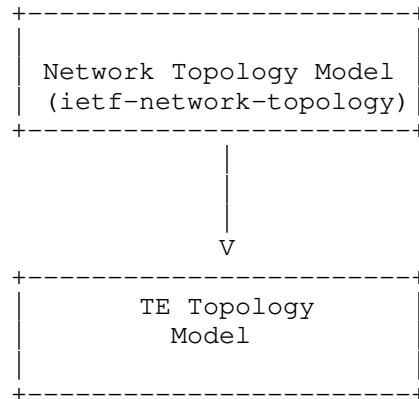
              Figure 12: Augmenting the Network Topology Model

5.2. Technology agnostic TE Topology model

   The TE Topology model proposed in this document is meant to be
   network technology agnostic. Other technology specific TE Topology
   models can augment and use the building blocks provided by the
   proposed model.

```
              +---------------------------+
              │        TE Topology Model  │
              │ (Defined in This Document) │
              +---------------------------+
                           │
          +------------+------------+------------+
          │            │            │            │
          V            V            V            V
    +-----------+                        +-----------+
    │ Technology │                       │ Technology │
    │  Specific  │  .....................  │  Specific  │
    │ TE Topology│                       │ TE Topology│
    │  Model 1   │                       │  Model n   │
    +-----------+                        +-----------+
```
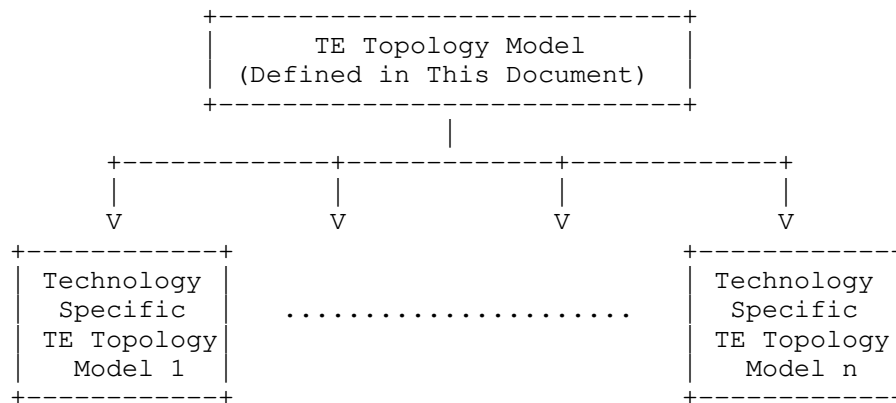
       Figure 13: Augmenting the Technology agnostic TE Topology model

5.3. Model Structure

   The high-level model structure proposed by this document is as shown
   below:

```
   module: ietf-te-topology
   augment /nw:networks/nw:network/nw:network-types:
      +--rw te-topology!

   augment /nw:networks:
      +--rw te!
         +--rw templates
            +--rw node-template* [name] {template}?
            │  ...........
            +--rw link-template* [name] {template}?
               ...........

   augment /nw:networks/nw:network:
      +--rw te-topology-identifier
      │  +--rw provider-id?   te-global-id
      │  +--rw client-id?     te-global-id
      │  +--rw topology-id?   te-topology-id
      +--rw te!
         │  ...........

   augment /nw:networks/nw:network/nw:node:
      +--rw te-node-id?   te-types:te-node-id
      +--rw te!
         │  ...........
         +--rw tunnel-termination-point* [tunnel-tp-id]
```

```
        +--rw tunnel-tp-id    binary
        | .............
        +--rw supporting-tunnel-termination-point* [node-ref tunnel-
tp-ref]
            | ............
```

```
augment /nw:networks/nw:network/nt:link:
   +--rw te!
   | ..........
```

```
augment /nw:networks/nw:network/nw:node/nt:termination-point:
   +--rw te-tp-id?   te-types:te-tp-id
   +--rw te!
   | ............
```

5.4. Topology Identifiers

The TE-Topology is uniquely identified by a key that has 3
constituents - topology-id, provider-id and client-id. The
combination of provider-id and topology-id uniquely identifies a
native TE Topology on a given provider. The client-id is used only
when Customized TE Topologies come into play; a value of "0" is used
as the client-id for native TE Topologies.

```
augment /nw:networks/nw:network:
   +--rw te-topology-identifier
   |  +--rw provider-id?   te-global-id
   |  +--rw client-id?     te-global-id
   |  +--rw topology-id?   te-topology-id
   +--rw te!
       | ...........
```

5.5. Generic TE Link Attributes

The model covers the definitions for generic TE Link attributes -
bandwidth, admin groups, SRLGs, switching capabilities, TE metric
extensions etc.

```
  +--rw te-link-attributes
     ....................
     +--rw admin-status?                   te-admin-status
     | ....................
     +--rw link-index?                     uint64
     +--rw administrative-group?           te-types:admin-groups
     +--rw link-protection-type?           enumeration
     +--rw max-link-bandwidth?             te-bandwidth
```

```
      +--rw max-resv-link-bandwidth?          te-bandwidth
      +--rw unreserved-bandwidth* [priority]
      | ....................
      +--rw te-default-metric?                uint32
      | ....................
      +--rw te-srlgs
      +--rw te-nsrlgs {nsrlg}?        ....................
```

5.6. Generic TE Node Attributes

The model covers the definitions for generic TE Node attributes.

The definition of a generic connectivity matrix is shown below:

```
   +--rw te-node-attributes
      ...........
      +--rw connectivity-matrices
      ...........
      |  +--rw connectivity-matrix* [id]
      |  |  +--rw id           uint32
      |  |  +--rw from
      |  |  |  +--rw tp-ref?        leafref
      |  |  |  +--rw label-restrictions
      |  |  +--rw to
      |  |  |  +--rw tp-ref?        leafref
      |  |  |  +--rw label-restrictions
      |  |  +--rw is-allowed?   boolean
      ...........
      |  |  +--rw underlay! {te-topology-hierarchy}?
      ...........
      |  |  +--rw path-constraints
      ...........
      |  |  +--rw optimizations
      ...........
      |  |  +--ro path-properties
      ...........
```

The definition of a TTP Local Link Connectivity List is shown below:

```
   +--rw tunnel-termination-point* [tunnel-tp-id]
      +--rw tunnel-tp-id                binary
      +--rw admin-status?               te-types:te-admin-status
      +--rw name?                       string
      +--rw switching-capability?       identityref
      +--rw encoding?                   identityref
      +--rw inter-layer-lock-id*        uint32
```

```
     +--rw protection-type?          Identityref
     +--rw client-layer-adaptation
     ...........
     +--rw local-link-connectivities
     ...........
     |  +--rw local-link-connectivity* [link-tp-ref]
     |     +--rw link-tp-ref               leafref
     |     +--rw label-restrictions
     ...........
     |     +--rw is-allowed?               boolean
     |     +--rw underlay {te-topology-hierarchy}?
     ...........
     |     +--rw path-constraints
     ...........
     |     +--rw optimizations
     ...........
     |     +--ro path-properties
     ...........
     +--rw supporting-tunnel-termination-point* [node-ref tunnel-tp-
  ref]
        +--rw node-ref         inet:uri
        +--rw tunnel-tp-ref    binary
```

   The attributes directly under container connectivity-matrices are the
   default attributes for all connectivity-matrix entries when the per
   entry corresponding attribute is not specified. When a per entry
   attribute is specified, it overrides the corresponding attribute
   directly under the container connectivity-matrices. The same rule
   applies to the attributes directly under container local-link-
   connectivities.

   Each TTP (Tunnel Termination Point) MAY be supported by one or more
   supporting TTPs. If the TE node hosting the TTP in question refers to
   a supporting TE node, then the supporting TTPs are hosted by the
   supporting TE node. If the TE node refers to an underlay TE topology,
   the supporting TTPs are hosted by one or more specified TE nodes of
   the underlay TE topology.

5.7. TED Information Sources

   The model allows each TE topological element to have multiple TE
   information sources (OSPF-TE, ISIS-TE, BGP-LS, User-Configured,
   System-Processed, Other). Each information source is associated with
   a credibility preference to indicate precedence. In scenarios where a
   customized TE Topology is merged into a Client's native TE Topology,
   the merged topological elements would point to the corresponding
   customized TE Topology as its information source.

```
   augment /nw:networks/nw:network/nw:node:
     +--rw te!
        ...........
        +--ro information-source?              te-info-source
        +--ro information-source-instance?    string
        +--ro information-source-state
        │  +--ro credibility-preference?    uint16
        │  +--ro logical-network-element?   string
        │  +--ro network-instance?          string
        │  +--ro topology
        │     +--ro node-ref?       leafref
        │     +--ro network-ref?    leafref
        +--ro information-source-entry*
        │      [information-source information-source-instance]
        │  +--ro information-source               te-info-source
        │  +--ro information-source-instance    string
               ...........

   augment /nw:networks/nw:network/nt:link:
     +--rw te!
        ...........
        +--ro information-source?              te-info-source
        +--ro information-source-instance?    string
        +--ro information-source-state
        │  +--ro credibility-preference?    uint16
        │  +--ro logical-network-element?   string
        │  +--ro network-instance?          string
        │  +--ro topology
        │     +--ro link-ref?       leafref
        │     +--ro network-ref?    leafref
        +--ro information-source-entry*
        │      [information-source information-source-instance]
        │  +--ro information-source               te-info-source
        │  +--ro information-source-instance    string
               ...........
```

5.8. Overlay/Underlay Relationship

   The model captures overlay and underlay relationship for TE
   nodes/links. For example - in networks where multiple TE Topologies
   are built hierarchically, this model allows the user to start from a
   specific topological element in the top most topology and traverse
   all the way down to the supporting topological elements in the bottom
   most topology.

   This relationship is captured via the "underlay-topology" field for
   the node and via the "underlay" field for the link. The use of these

fields is optional and this functionality is tagged as a "feature"
("te-topology-hierarchy").

```
augment /nw:networks/nw:network/nw:node:
   +--rw te-node-id?   te-types:te-node-id
   +--rw te!
      +--rw te-node-template*           leafref {template}?
      +--rw te-node-attributes
      |  +--rw admin-status?               te-types:te-admin-status
      |  |  ....................
      |  +--rw underlay-topology {te-topology-hierarchy}?
      |     +--rw network-ref?   leafref

augment /nw:networks/nw:network/nt:link:
   +--rw te!
      +--rw te-link-attributes
      |  ...................
      |  +--rw underlay {te-topology-hierarchy}?
      |  |  +--rw enabled?                    boolean
      |  |  +--rw primary-path
      |  |  |  +--rw network-ref?   leafref
      |  |  |     ....................
      |  |  +--rw backup-path* [index]
      |  |  |  +--rw index          uint32
      |  |  |  +--rw network-ref?   leafref
      |  |  |     ....................
      |  |  +--rw protection-type?            identityref
      |  |  +--rw tunnel-termination-points
      |  |  |  +--rw source?        binary
      |  |  |  +--rw destination?   binary
      |  |  +--rw tunnels
      |  |  |  ...................
```

5.9. Templates

   The data model provides the users with the ability to define
   templates and apply them to link and node configurations. The use of
   "template" configuration is optional and this functionality is tagged
   as a "feature" ("template").

```
augment /nw:networks/nw:network/nw:node:
   +--rw te-node-id?   te-types:te-node-id
   +--rw te!
      +--rw te-node-template*
      |        -> ../../../../te/templates/node-template/name
      |        {template}?
```

```
   augment /nw:networks/nw:network/nt:link:
      +--rw te!
         +--rw te-link-template*
         │         -> ../../../../te/templates/link-template/name
         │      {template}?

   augment /nw:networks:
      +--rw te!
         +--rw templates
            +--rw node-template* [name] {template}?
            │  +--rw name
            │  │      te-types:te-template-name
            │  +--rw priority?                 uint16
            │  +--rw reference-change-policy?  enumeration
            │  +--rw te-node-attributes
            │  ..........
            +--rw link-template* [name] {template}?
               +--rw name
               │      te-types:te-template-name
               +--rw priority?                 uint16
               +--rw reference-change-policy?  enumeration
               +--rw te-link-attributes
               ..........
```

   Multiple templates can be specified to a configuration element. When
   two or more templates specify values for the same configuration
   field, the value from the template with the highest priority is used.
   The range of the priority is from 0 to 65535, with a lower number
   indicating a higher priority. The reference-change-policy specifies
   the action that needs to be taken when the template changes on a
   configuration element that has a reference to this template. The
   choices of action include taking no action, rejecting the change to
   the template and applying the change to the corresponding
   configuration.

5.10. Scheduling Parameters

   The model allows time scheduling parameters to be specified for each
   topological element or for the topology as a whole. These parameters
   allow the provider to present different topological views to the
   client at different time slots. The use of "scheduling parameters" is
   optional.

   The YANG data model for configuration scheduling is defined in
   [I-D.liu-netmod-yang-schedule], which allows specifying configuration
   schedules without altering this data model.

5.11. Notifications

   Notifications are a key component of any topology data model.

   [I-D.ietf-netconf-subscribed-notifications] and
   [I-D.ietf-netconf-yang-push] define a subscription and push mechanism
   for YANG datastores. This mechanism currently allows the user to:

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

6. Guidance for Writing Technology Specific TE Topology Augmentations

   The TE topology model defined in this document is technology agnostic
   as it defines concepts, abstractions and attributes that are common
   across multiple network technologies. It is envisioned that this base
   model will be widely used when defining technology specific TE
   topology models for various layer networks.
   [I-D.ietf-ccamp-wson-yang], [I-D.ietf-ccamp-otn-topo-yang], and
   [I-D.ietf-teas-yang-l3-te-topo] are some examples of technology
   specific TE Topology models. Writers of such models are encouraged to
   augment the basic TE topology model's containers, such as TE
   Topology, TE Node, TE Link, Link Termination Point (LTP), Tunnel
   Termination Point (TTP), Bandwidth and Label with the layer specific
   attributes instead of defining new containers.

   Consider the following technology specific example-topology model:

```
module: example-topology
  augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
    +--rw example-topology!
  augment /nw:networks/nw:network/tet:te:
    +--rw attributes
       +--rw attribute-1?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
          /tet:te-node-attributes:
    +--rw attributes
       +--rw attribute-2?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
          /tet:te-node-attributes/tet:connectivity-matrices:
    +--rw attributes
       +--rw attribute-3?   uint8
  augment /nw:networks/nw:network/nw:node/tet:te
```

```
                 /tet:te-node-attributes/tet:connectivity-matrices
                 /tet:connectivity-matrix:
        +--rw attributes
           +--rw attribute-3?   uint8
      augment /nw:networks/nw:network/nw:node/tet:te
                 /tet:tunnel-termination-point:
        +--rw attributes
           +--rw attribute-4?   uint8
      augment /nw:networks/nw:network/nw:node/nt:termination-point
                 /tet:te:
        +--rw attributes
           +--rw attribute-5?   uint8
      augment /nw:networks/nw:network/nt:link/tet:te
                 /tet:te-link-attributes:
        +--rw attributes
           +--rw attribute-6?   uint8
```

   The technology specific TE bandwidth for this example topology can be
   specified using the following augment statements:

```
   augment /nw:networks/tet:te/tet:templates/tet:link-template
             /tet:te-link-attributes
             /tet:interface-switching-capability/tet:max-lsp-bandwidth
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/tet:te/tet:templates/tet:link-template
             /tet:te-link-attributes/tet:max-link-bandwidth
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/tet:te/tet:templates/tet:link-template
             /tet:te-link-attributes/tet:max-resv-link-bandwidth
             /tet:te-bandwidth/tet:technology:
     +--:(example)
        +--rw example
           +--rw bandwidth-1?   uint32
   augment /nw:networks/tet:te/tet:templates/tet:link-template
```

```
                /tet:te-link-attributes/tet:unreserved-bandwidth
                /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:te-node-attributes/tet:connectivity-matrices
                /tet:path-constraints/tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:te-node-attributes/tet:connectivity-matrices
                /tet:connectivity-matrix/tet:path-constraints
                /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:information-source-entry/tet:connectivity-matrices
                /tet:path-constraints/tet:te-bandwidth/tet:technology:
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:information-source-entry/tet:connectivity-matrices
                /tet:connectivity-matrix/tet:path-constraints
                /tet:te-bandwidth/tet:technology:
       +--:(example)
          +--ro example
             +--ro bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:tunnel-termination-point/tet:client-layer-adaptation
                /tet:switching-capability/tet:te-bandwidth
                /tet:technology:
       +--:(example)
          +--rw example
             +--rw bandwidth-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
                /tet:tunnel-termination-point
                /tet:local-link-connectivities/tet:path-constraints
```

```
                /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--rw example
            +--rw bandwidth-1?   uint32
      augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point
              /tet:local-link-connectivities
              /tet:local-link-connectivity/tet:path-constraints
              /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--rw example
            +--rw bandwidth-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:te-link-attributes
              /tet:interface-switching-capability/tet:max-lsp-bandwidth
              /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--rw example
            +--rw bandwidth-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:te-link-attributes/tet:max-link-bandwidth
              /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--rw example
            +--rw bandwidth-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:te-link-attributes/tet:max-resv-link-bandwidth
              /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--rw example
            +--rw bandwidth-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:information-source-entry
              /tet:interface-switching-capability/tet:max-lsp-bandwidth
              /tet:te-bandwidth/tet:technology:
      +--:(example)
         +--ro example
            +--ro bandwidth-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:information-source-entry/tet:max-link-bandwidth
              /tet:te-bandwidth/tet:technology:
```

```
          +--:(example)
             +--ro example
                +--ro bandwidth-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:information-source-entry/tet:max-resv-link-bandwidth
                /tet:te-bandwidth/tet:technology:
          +--:(example)
             +--ro example
                +--ro bandwidth-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:information-source-entry/tet:unreserved-bandwidth
                /tet:te-bandwidth/tet:technology:
          +--:(example)
             +--ro example
                +--ro bandwidth-1?   uint32
       augment /nw:networks/nw:network/nw:node/nt:termination-point/tet:te
                /tet:interface-switching-capability/tet:max-lsp-bandwidth
                /tet:te-bandwidth/tet:technology:
          +--:(example)
             +--rw example
                +--rw bandwidth-1?   uint32
```

   The technology specific TE label for this example topology can be
   specified using the following augment statements:

```
       augment /nw:networks/tet:te/tet:templates/tet:link-template
                /tet:te-link-attributes/tet:underlay/tet:primary-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
          +--:(example)
             +--rw example
                +--rw label-1?   uint32
       augment /nw:networks/tet:te/tet:templates/tet:link-template
                /tet:te-link-attributes/tet:underlay/tet:backup-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
          +--:(example)
             +--rw example
                +--rw label-1?   uint32
       augment /nw:networks/tet:te/tet:templates/tet:link-template
```

```
             /tet:te-link-attributes/tet:label-restrictions
             /tet:label-restriction/tet:label-start/tet:te-label
             /tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
   augment /nw:networks/tet:te/tet:templates/tet:link-template
             /tet:te-link-attributes/tet:label-restrictions
             /tet:label-restriction/tet:label-end/tet:te-label
             /tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:label-restrictions/tet:label-restriction
             /tet:label-start/tet:te-label/tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:label-restrictions/tet:label-restriction
             /tet:label-end/tet:te-label/tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:underlay/tet:primary-path/tet:path-element/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
   augment /nw:networks/nw:network/nw:node/tet:te
             /tet:te-node-attributes/tet:connectivity-matrices
             /tet:underlay/tet:backup-path/tet:path-element/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
     +--:(example)
        +--rw example
           +--rw label-1?   uint32
```

```
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:te-node-attributes/tet:connectivity-matrices
               /tet:path-properties/tet:path-route-objects
               /tet:path-route-object/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:te-node-attributes/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:te-node-attributes/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:te-node-attributes/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:to/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
               /tet:te-node-attributes/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:to/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
```

```
                /tet:te-node-attributes/tet:connectivity-matrices
                /tet:connectivity-matrix/tet:underlay/tet:primary-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
        +--:(example)
           +--rw example
              +--rw label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
                /tet:te-node-attributes/tet:connectivity-matrices
                /tet:connectivity-matrix/tet:underlay/tet:backup-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
        +--:(example)
           +--rw example
              +--rw label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
                /tet:te-node-attributes/tet:connectivity-matrices
                /tet:connectivity-matrix/tet:path-properties
                /tet:path-route-objects/tet:path-route-object/tet:type
                /tet:label/tet:label-hop/tet:te-label/tet:technology:
        +--:(example)
           +--ro example
              +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
                /tet:information-source-entry/tet:connectivity-matrices
                /tet:label-restrictions/tet:label-restriction
                /tet:label-start/tet:te-label/tet:technology:
        +--:(example)
           +--ro example
              +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
                /tet:information-source-entry/tet:connectivity-matrices
                /tet:label-restrictions/tet:label-restriction
                /tet:label-end/tet:te-label/tet:technology:
        +--:(example)
           +--ro example
              +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
                /tet:information-source-entry/tet:connectivity-matrices
                /tet:underlay/tet:primary-path/tet:path-element/tet:type
                /tet:label/tet:label-hop/tet:te-label/tet:technology:
```

```
      +--:(example)
         +--ro example
            +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:underlay/tet:backup-path/tet:path-element/tet:type
               /tet:label/tet:label-hop/tet:te-label/tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:path-properties/tet:path-route-objects
               /tet:path-route-object/tet:type/tet:label/tet:label-hop
               /tet:te-label/tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:from/tet:label-restrictions
               /tet:label-restriction/tet:label-end/tet:te-label
               /tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?    uint32
      augment /nw:networks/nw:network/nw:node/tet:te
               /tet:information-source-entry/tet:connectivity-matrices
               /tet:connectivity-matrix/tet:to/tet:label-restrictions
               /tet:label-restriction/tet:label-start/tet:te-label
               /tet:technology:
      +--:(example)
         +--ro example
```

```
            +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:to/tet:label-restrictions
             /tet:label-restriction/tet:label-end/tet:te-label
             /tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:underlay/tet:primary-path
             /tet:path-element/tet:type/tet:label/tet:label-hop
             /tet:te-label/tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:underlay/tet:backup-path
             /tet:path-element/tet:type/tet:label/tet:label-hop
             /tet:te-label/tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:information-source-entry/tet:connectivity-matrices
             /tet:connectivity-matrix/tet:path-properties
             /tet:path-route-objects/tet:path-route-object/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
      +--:(example)
         +--ro example
            +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:label-restrictions
             /tet:label-restriction/tet:label-start/tet:te-label
             /tet:technology:
      +--:(example)
         +--rw example
            +--rw label-1?   uint32
```

```
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:label-restrictions
             /tet:label-restriction/tet:label-end/tet:te-label
             /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:underlay
             /tet:primary-path/tet:path-element/tet:type/tet:label
             /tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:underlay
             /tet:backup-path/tet:path-element/tet:type/tet:label
             /tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities/tet:path-properties
             /tet:path-route-objects/tet:path-route-object/tet:type
             /tet:label/tet:label-hop/tet:te-label/tet:technology:
       +--:(example)
          +--ro example
             +--ro label-1?   uint32
     augment /nw:networks/nw:network/nw:node/tet:te
             /tet:tunnel-termination-point
             /tet:local-link-connectivities
             /tet:local-link-connectivity/tet:label-restrictions
             /tet:label-restriction/tet:label-start/tet:te-label
             /tet:technology:
       +--:(example)
          +--rw example
             +--rw label-1?   uint32
```

```
      augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point
              /tet:local-link-connectivities
              /tet:local-link-connectivity/tet:label-restrictions
              /tet:label-restriction/tet:label-end/tet:te-label
              /tet:technology:
        +--:(example)
           +--rw example
              +--rw label-1?   uint32
      augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point
              /tet:local-link-connectivities
              /tet:local-link-connectivity/tet:underlay
              /tet:primary-path/tet:path-element/tet:type/tet:label
              /tet:label-hop/tet:te-label/tet:technology:
        +--:(example)
           +--rw example
              +--rw label-1?   uint32
      augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point
              /tet:local-link-connectivities
              /tet:local-link-connectivity/tet:underlay/tet:backup-path
              /tet:path-element/tet:type/tet:label/tet:label-hop
              /tet:te-label/tet:technology:
        +--:(example)
           +--rw example
              +--rw label-1?   uint32
      augment /nw:networks/nw:network/nw:node/tet:te
              /tet:tunnel-termination-point
              /tet:local-link-connectivities
              /tet:local-link-connectivity/tet:path-properties
              /tet:path-route-objects/tet:path-route-object/tet:type
              /tet:label/tet:label-hop/tet:te-label/tet:technology:
        +--:(example)
           +--ro example
              +--ro label-1?   uint32
      augment /nw:networks/nw:network/nt:link/tet:te
              /tet:te-link-attributes/tet:label-restrictions
              /tet:label-restriction/tet:label-start/tet:te-label
              /tet:technology:
        +--:(example)
```

```
           +--rw example
              +--rw label-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:te-link-attributes/tet:label-restrictions
                /tet:label-restriction/tet:label-end/tet:te-label
                /tet:technology:
         +--:(example)
           +--rw example
              +--rw label-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:te-link-attributes/tet:underlay/tet:primary-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
         +--:(example)
           +--rw example
              +--rw label-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:te-link-attributes/tet:underlay/tet:backup-path
                /tet:path-element/tet:type/tet:label/tet:label-hop
                /tet:te-label/tet:technology:
         +--:(example)
           +--rw example
              +--rw label-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:information-source-entry/tet:label-restrictions
                /tet:label-restriction/tet:label-start/tet:te-label
                /tet:technology:
         +--:(example)
           +--ro example
              +--ro label-1?   uint32
       augment /nw:networks/nw:network/nt:link/tet:te
                /tet:information-source-entry/tet:label-restrictions
                /tet:label-restriction/tet:label-end/tet:te-label
                /tet:technology:
         +--:(example)
           +--ro example
              +--ro label-1?   uint32
```

   The YANG module to implement the above example topology can be seen
   in Appendix C.

7. TE Topology YANG Module

   This module references [RFC1195], [RFC3209], [RFC3272], [RFC3471],
   [RFC3630], [RFC3785], [RFC4201], [RFC4202], [RFC4203], [RFC4206],
   [RFC4872], [RFC5152], [RFC5212], [RFC5305], [RFC5316], [RFC5329],
   [RFC5392], [RFC6001], [RFC6241], [RFC6991], [RFC7308], [RFC7471],
   [RFC7579], [RFC7752], [RFC8345], and [I-D.ietf-teas-yang-te-types].

   <CODE BEGINS> file "ietf-te-topology@2019-02-07.yang"
   module ietf-te-topology {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology";

     prefix "tet";

     import ietf-yang-types {
       prefix "yang";
       reference "RFC 6991: Common YANG Data Types";
     }

     import ietf-inet-types {
       prefix "inet";
       reference "RFC 6991: Common YANG Data Types";
     }

     import ietf-te-types {
       prefix "te-types";
       reference
         "I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
          Types";
     }

     import ietf-network {
       prefix "nw";
       reference "RFC 8345: A YANG Data Model for Network Topologies";
     }

     import ietf-network-topology {
       prefix "nt";
       reference "RFC 8345: A YANG Data Model for Network Topologies";
     }

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

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

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

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

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

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

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

          Editor: Oscar Gonzalez De Dios
                    <mailto:oscar.gonzalezdedios@telefonica.com>";

       description
         "TE topology model for representing and manipulating technology
          agnostic TE Topologies.

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

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

          This version of this YANG module is part of RFC XXXX; see the

```
      RFC itself for full legal notices.";

    revision "2019-02-07" {
      description "Initial revision";
      reference "RFC XXXX: YANG Data Model for TE Topologies";
    // RFC Ed.: replace XXXX with actual RFC number and remove
    // this note
    }

    /*
     * Features
     */
    feature nsrlg {
      description
        "This feature indicates that the system supports NSRLG
         (Not Sharing Risk Link Group).";
    }

    feature te-topology-hierarchy {
      description
        "This feature indicates that the system allows underlay
         and/or overlay TE topology hierarchy.";
    }

    feature template {
      description
        "This feature indicates that the system supports
         template configuration.";
    }

    /*
     * Typedefs
     */
    typedef geographic-coordinate-degree {
        type decimal64 {
          fraction-digits 8;
        }
        description
          "Decimal degree (DD) used to express latitude and longitude
           geographic coordinates.";
    } // geographic-coordinate-degree
```

```
      typedef te-info-source {
        type enumeration {
          enum "unknown" {
            description "The source is unknown.";
          }
          enum "locally-configured" {
            description "Configured entity.";
          }
          enum "ospfv2" {
            description "OSPFv2.";
          }
          enum "ospfv3" {
            description "OSPFv3.";
          }
          enum "isis" {
            description "ISIS.";
          }
          enum "bgp-ls" {
            description "BGP-LS.";
            reference
              "RFC 7752: North-Bound Distribution of Link-State and
               Traffic Engineering (TE) Information Using BGP";
          }
          enum "system-processed" {
            description "System processed entity.";
          }
          enum "other" {
            description "Other source.";
          }
        }
        description
          "Describining the type of source that has provided the
           related information, and the source credibility.";
      } // te-info-source

      /*
       * Groupings
       */
      grouping connectivity-matrix-entry-path-attributes {
        description
```

```
            "Attributes of connectivity matrix entry.";
          leaf is-allowed {
            type boolean;
            description
              "true  - switching is allowed,
               false - switching is disallowed.";
          }
          container underlay {
            if-feature te-topology-hierarchy;
            description "Attributes of the te-link underlay.";
            reference
              "RFC 4206: Label Switched Paths (LSP) Hierarchy with
               Generalized Multi-Protocol Label Switching (GMPLS)
               Traffic Engineering (TE)";

            uses te-link-underlay-attributes;
          } // underlay

          uses te-types:generic-path-constraints;
          uses te-types:generic-path-optimization;
          uses te-types:generic-path-properties;
        } // connectivity-matrix-entry-path-attributes

        grouping geolocation-container {
          description
            "A container containing a GPS location.";
          container geolocation{
            config false;
            description
              "A container containing a GPS location.";
            leaf altitude {
              type int64;
              units millimeter;
              description
                "Distance above the sea level.";
            }
            leaf latitude {
              type geographic-coordinate-degree {
                range "-90..90";
              }
              description
```

```
            "Relative position north or south on the Earth's surface.";
          }
          leaf longitude {
            type geographic-coordinate-degree {
              range "-180..180";
            }
            description
              "Angular distance east or west on the Earth's surface.";
          }
        } // gps-location
      } // geolocation-container

      grouping information-source-state-attributes {
        description
          "The attributes identifying source that has provided the
           related information, and the source credibility.";
        leaf credibility-preference {
          type uint16;
          description
            "The preference value to calculate the traffic
             engineering database credibility value used for
             tie-break selection between different
             information-source values.
             Higher value is more preferable.";
        }
        leaf logical-network-element {
          type string;
          description
            "When applicable, this is the name of a logical network
             element from which the information is learned.";
        } // logical-network-element
        leaf network-instance {
          type string;
          description
            "When applicable, this is the name of a network-instance
             from which the information is learned.";
        } // network-instance
      } // information-source-state-attributes

      grouping information-source-per-link-attributes {
        description
```

```
          "Per node container of the attributes identifying source that
           has provided the related information, and the source
           credibility.";
        leaf information-source {
          type te-info-source;
          config false;
          description
            "Indicates the type of the information source.";
        }
        leaf information-source-instance {
          type string;
          config false;
          description
            "The name indicating the instance of the information
             source.";
        }
        container information-source-state {
          config false;
          description
            "The container contains state attributes related to
             the information source.";
          uses information-source-state-attributes;
          container topology {
            description
              "When the information is processed by the system,
               the attributes in this container indicate which topology
               is used to process to generate the result information.";
            uses nt:link-ref;
          } // topology
        } // information-source-state
      } // information-source-per-link-attributes

      grouping information-source-per-node-attributes {
        description
          "Per node container of the attributes identifying source that
           has provided the related information, and the source
           credibility.";
        leaf information-source {
          type te-info-source;
          config false;
          description
```

```
          "Indicates the type of the information source.";
      }
      leaf information-source-instance {
        type string;
        config false;
        description
          "The name indicating the instance of the information
           source.";
      }
      container information-source-state {
        config false;
        description
          "The container contains state attributes related to
           the information source.";
        uses information-source-state-attributes;
        container topology {
          description
            "When the information is processed by the system,
             the attributes in this container indicate which topology
             is used to process to generate the result information.";
          uses nw:node-ref;
        } // topology
      } // information-source-state
    } // information-source-per-node-attributes

    grouping interface-switching-capability-list {
      description
        "List of Interface Switching Capabilities Descriptors (ISCD)";
      list interface-switching-capability {
        key "switching-capability encoding";
        description
          "List of Interface Switching Capabilities Descriptors (ISCD)
           for this link.";
        reference
          "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
           Signaling Functional Description.
           RFC 4203: OSPF Extensions in Support of Generalized
           Multi-Protocol Label Switching (GMPLS).";
        leaf switching-capability {
          type identityref {
            base te-types:switching-capabilities;
```

```
        }
        description
          "Switching Capability for this interface.";
      }
      leaf encoding {
        type identityref {
          base te-types:lsp-encoding-types;
        }
        description
          "Encoding supported by this interface.";
      }
      uses te-link-iscd-attributes;
    } // interface-switching-capability
  } // interface-switching-capability-list

  grouping statistics-per-link {
    description
      "Statistics attributes per TE link.";
    leaf discontinuity-time {
      type yang:date-and-time;
      description
        "The time on the most recent occasion at which any one or
         more of this interface's counters suffered a
         discontinuity.  If no such discontinuities have occurred
         since the last re-initialization of the local management
         subsystem, then this node contains the time the local
         management subsystem re-initialized itself.";
    }
    /* Administrative attributes */
    leaf disables {
      type yang:counter32;
      description
        "Number of times that link was disabled.";
    }
    leaf enables {
      type yang:counter32;
      description
        "Number of times that link was enabled.";
    }
    leaf maintenance-clears {
      type yang:counter32;
```

```
          description
            "Number of times that link was put out of maintenance.";
        }
        leaf maintenance-sets {
          type yang:counter32;
          description
            "Number of times that link was put in maintenance.";
        }
        leaf modifies {
          type yang:counter32;
          description
            "Number of times that link was modified.";
        }
        /* Operational attributes */
        leaf downs {
          type yang:counter32;
          description
            "Number of times that link was set to operational down.";
        }
        leaf ups {
          type yang:counter32;
          description
            "Number of times that link was set to operational up.";
        }
        /* Recovery attributes */
        leaf fault-clears {
          type yang:counter32;
          description
            "Number of times that link experienced fault clear event.";
        }
        leaf fault-detects {
          type yang:counter32;
          description
            "Number of times that link experienced fault detection.";
        }
        leaf protection-switches {
          type yang:counter32;
          description
            "Number of times that link experienced protection
             switchover.";
        }
```

```
       leaf protection-reverts {
         type yang:counter32;
         description
           "Number of times that link experienced protection
            reversion.";
       }
       leaf restoration-failures {
         type yang:counter32;
         description
           "Number of times that link experienced restoration
            failure.";
       }
       leaf restoration-starts {
         type yang:counter32;
         description
           "Number of times that link experienced restoration
            start.";
       }
       leaf restoration-successes {
         type yang:counter32;
         description
           "Number of times that link experienced restoration
            success.";
       }
       leaf restoration-reversion-failures {
         type yang:counter32;
         description
           "Number of times that link experienced restoration reversion
            failure.";
       }
       leaf restoration-reversion-starts {
         type yang:counter32;
         description
           "Number of times that link experienced restoration reversion
            start.";
       }
       leaf restoration-reversion-successes {
         type yang:counter32;
         description
           "Number of times that link experienced restoration reversion
            success.";
```

```
      }
    } // statistics-per-link

    grouping statistics-per-node {
      description
        "Statistics attributes per TE node.";
      leaf discontinuity-time {
        type yang:date-and-time;
        description
          "The time on the most recent occasion at which any one or
           more of this interface's counters suffered a
           discontinuity.  If no such discontinuities have occurred
           since the last re-initialization of the local management
           subsystem, then this node contains the time the local
           management subsystem re-initialized itself.";
      }
      container node {
        description
          "Containing TE node level statistics attributes.";
        leaf disables {
          type yang:counter32;
          description
            "Number of times that node was disabled.";
        }
        leaf enables {
          type yang:counter32;
          description
            "Number of times that node was enabled.";
        }
        leaf maintenance-sets {
          type yang:counter32;
          description
            "Number of times that node was put in maintenance.";
        }
        leaf maintenance-clears {
          type yang:counter32;
          description
            "Number of times that node was put out of maintenance.";
        }
        leaf modifies {
          type yang:counter32;
```

```
          description
            "Number of times that node was modified.";
        }
      } // node
      container connectivity-matrix-entry {
        description
          "Containing connectivity matrix entry level statistics
           attributes.";
        leaf creates {
          type yang:counter32;
          description
            "Number of times that a connectivity matrix entry was
             created.";
          reference
            "RFC 6241. Section 7.2 for 'create' operation. ";
        }
        leaf deletes {
          type yang:counter32;
          description
            "Number of times that a connectivity matrix entry was
             deleted.";
          reference
            "RFC 6241. Section 7.2 for 'delete' operation. ";
        }
        leaf disables {
          type yang:counter32;
          description
            "Number of times that a connectivity matrix entry was
             disabled.";
        }
        leaf enables {
          type yang:counter32;
          description
            "Number of times that a connectivity matrix entry was
             enabled.";
        }
        leaf modifies {
          type yang:counter32;
          description
            "Number of times that a connectivity matrix entry was
             modified.";
```

```
        }
      } // connectivity-matrix-entry
    } // statistics-per-node

    grouping statistics-per-ttp {
      description
        "Statistics attributes per TE TTP (Tunnel Termination Point).";
      leaf discontinuity-time {
        type yang:date-and-time;
        description
          "The time on the most recent occasion at which any one or
           more of this interface's counters suffered a
           discontinuity.  If no such discontinuities have occurred
           since the last re-initialization of the local management
           subsystem, then this node contains the time the local
           management subsystem re-initialized itself.";
      }
      container tunnel-termination-point {
        description
          "Containing TE TTP (Tunnel Termination Point) level
           statistics attributes.";
        /* Administrative attributes */
        leaf disables {
          type yang:counter32;
          description
            "Number of times that TTP was disabled.";
        }
        leaf enables {
          type yang:counter32;
          description
            "Number of times that TTP was enabled.";
        }
        leaf maintenance-clears {
          type yang:counter32;
          description
            "Number of times that TTP was put out of maintenance.";
        }
        leaf maintenance-sets {
          type yang:counter32;
          description
            "Number of times that TTP was put in maintenance.";
```

```
        }
        leaf modifies {
          type yang:counter32;
          description
            "Number of times that TTP was modified.";
        }
        /* Operational attributes */
        leaf downs {
          type yang:counter32;
          description
            "Number of times that TTP was set to operational down.";
        }
        leaf ups {
          type yang:counter32;
          description
            "Number of times that TTP was set to operational up.";
        }
        leaf in-service-clears {
          type yang:counter32;
          description
            "Number of times that TTP was taken out of service
             (TE tunnel was released).";
        }
        leaf in-service-sets {
          type yang:counter32;
          description
            "Number of times that TTP was put in service by a TE
             tunnel (TE tunnel was set up).";
        }
      } // tunnel-termination-point

      container local-link-connectivity {
        description
          "Containing TE LLCL (Local Link Connectivity List) level
           statistics attributes.";
        leaf creates {
          type yang:counter32;
          description
            "Number of times that an LLCL entry was created.";
          reference
            "RFC 6241. Section 7.2 for 'create' operation.";
```

```
        }
        leaf deletes {
          type yang:counter32;
          description
            "Number of times that an LLCL entry was deleted.";
          reference
            "RFC 6241. Section 7.2 for 'delete' operation.";
        }
        leaf disables {
          type yang:counter32;
          description
            "Number of times that an LLCL  entry was disabled.";
        }
        leaf enables {
          type yang:counter32;
          description
            "Number of times that an LLCL  entry was enabled.";
        }
        leaf modifies {
          type yang:counter32;
          description
            "Number of times that an LLCL  entry was modified.";
        }
      } // local-link-connectivity
    } // statistics-per-ttp

    grouping te-link-augment {
      description
        "Augmentation for TE link.";
      uses te-link-config;
      uses te-link-state-derived;
      container statistics {
        config false;
        description
          "Statistics data.";
        uses statistics-per-link;
      } // statistics
    } // te-link-augment

    grouping te-link-config {
      description
```

```
        "TE link configuration grouping.";
      choice bundle-stack-level {
        description
          "The TE link can be partitioned into bundled
           links, or component links.";
        case bundle {
          container bundled-links {
            description
              "A set of bundled links.";
            reference
              "RFC 4201: Link Bundling in MPLS Traffic Engineering
              (TE).";
            list bundled-link {
              key "sequence";
              description
                "Specify a bundled interface that is
                 further partitioned.";
              leaf sequence {
                type uint32;
                description
                  "Identify the sequence in the bundle.";
              }
            } // list bundled-link
          }
        }
        case component {
          container component-links {
            description
              "A set of component links";
            list component-link {
              key "sequence";
              description
                "Specify a component interface that is
                 sufficient to unambiguously identify the
                 appropriate resources";

              leaf sequence {
                type uint32;
                description
                  "Identify the sequence in the bundle.";
              }
```

```
            leaf src-interface-ref {
              type string;
              description
                "Reference to component link interface on the
                 source node.";
            }
            leaf des-interface-ref {
              type string;
              description
                "Reference to component link interface on the
                 destinatioin node.";
            }
          }
        }
      }
    } // bundle-stack-level

    leaf-list te-link-template {
      if-feature template;
      type leafref {
        path "../../../../te/templates/link-template/name";
      }
      description
        "The reference to a TE link template.";
    }
    uses te-link-config-attributes;
  } // te-link-config

  grouping te-link-config-attributes {
    description
      "Link configuration attributes in a TE topology.";
    container te-link-attributes {
      description "Link attributes in a TE topology.";
      leaf access-type {
        type te-types:te-link-access-type;
        description
          "Link access type, which can be point-to-point or
           multi-access.";
      }
      container external-domain {
        description
```

```
              "For an inter-domain link, specify the attributes of
               the remote end of link, to facilitate the signalling at
               local end.";
            uses nw:network-ref;
            leaf remote-te-node-id {
              type te-types:te-node-id;
              description
                "Remote TE node identifier, used together with
                 remote-te-link-id to identify the remote link
                 termination point in a different domain.";
            }
            leaf remote-te-link-tp-id {
              type te-types:te-tp-id;
              description
                "Remote TE link termination point identifier, used
                 together with remote-te-node-id to identify the remote
                 link termination point in a different domain.";
            }
          }
          leaf is-abstract {
            type empty;
            description "Present if the link is abstract.";
          }
          leaf name {
            type string;
            description "Link Name.";
          }
          container underlay {
            if-feature te-topology-hierarchy;
            description "Attributes of the te-link underlay.";
            reference
              "RFC 4206: Label Switched Paths (LSP) Hierarchy with
               Generalized Multi-Protocol Label Switching (GMPLS)
               Traffic Engineering (TE)";

            uses te-link-underlay-attributes;
          } // underlay
          leaf admin-status {
            type te-types:te-admin-status;
            description
              "The administrative state of the link.";
```

```
          }

        uses te-link-info-attributes;
      } // te-link-attributes
    } // te-link-config-attributes

    grouping te-link-info-attributes {
      description
        "Advertised TE information attributes.";
      leaf link-index {
        type uint64;
        description
          "The link identifier.  If OSPF is used, this represents an
           ospfLsdbID.  If IS-IS is used, this represents an isisLSPID.
           If a locally configured link is used, this object represents
           a unique value, which is locally defined in a router.";
      }
      leaf administrative-group {
        type te-types:admin-groups;
        description
          "Administrative group or color of the link.
           This attribute covers both administrative group (defined in
           RFC 3630, RFC 5305 and RFC 5329), and extended
           administrative group (defined in RFC 7308).";
      }

      uses interface-switching-capability-list;
      uses te-types:label-set-info;

      leaf link-protection-type {
        type identityref {
          base te-types:link-protection-type;
        }
        description
          "Link Protection Type desired for this link.";
        reference
          "RFC 4202: Routing Extensions in Support of
           Generalized Multi-Protocol Label Switching (GMPLS).";
      }

      container max-link-bandwidth {
```

```
        uses te-types:te-bandwidth;
        description
          "Maximum bandwidth that can be seen on this link in this
           direction. Units in bytes per second.";
        reference
          "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
           Version 2.
           RFC 5305: IS-IS Extensions for Traffic Engineering.";
      }
      container max-resv-link-bandwidth {
        uses te-types:te-bandwidth;
        description
          "Maximum amount of bandwidth that can be reserved in this
           direction in this link. Units in bytes per second.";
        reference
          "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
           Version 2.
           RFC 5305: IS-IS Extensions for Traffic Engineering.";
      }
      list unreserved-bandwidth {
        key "priority";
        max-elements "8";
        description
          "Unreserved bandwidth for 0-7 priority levels. Units in
           bytes per second.";
        reference
          "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
           Version 2.
           RFC 5305: IS-IS Extensions for Traffic Engineering.";
        leaf priority {
          type uint8 {
            range "0..7";
          }
          description "Priority.";
        }
        uses te-types:te-bandwidth;
      }
      leaf te-default-metric {
        type uint32;
        description
          "Traffic engineering metric.";
```

```
          reference
            "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
             Version 2.
             RFC 5305: IS-IS Extensions for Traffic Engineering.";
        }
        leaf te-delay-metric {
          type uint32;
          description
            "Traffic engineering delay metric.";
          reference
            "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.";
        }
        leaf te-igp-metric {
          type uint32;
          description
            "IGP metric used for traffic engineering.";
          reference
            "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
             Second MPLS Traffic Engineering (TE) Metric.";
        }
        container te-srlgs {
          description
            "Containing a list of SLRGs.";
          leaf-list value {
            type te-types:srlg;
            description "SRLG value.";
            reference
              "RFC 4202: Routing Extensions in Support of
               Generalized Multi-Protocol Label Switching (GMPLS).";
          }
        }
        container te-nsrlgs {
          if-feature nsrlg;
          description
            "Containing a list of NSRLGs (Not Sharing Risk Link
             Groups).
             When an abstract TE link is configured, this list specifies
             the request that underlay TE paths need to be mutually
             disjoint with other TE links in the same groups.";
          leaf-list id {
            type uint32;
```

```
          description
            "NSRLG ID, uniquely configured within a topology.";
          reference
            "RFC 4872: RSVP-TE Extensions in Support of End-to-End
             Generalized Multi-Protocol Label Switching (GMPLS)
             Recovery";
        }
      }
    } // te-link-info-attributes

    grouping te-link-iscd-attributes {
      description
        "TE link ISCD (Interface Switching Capability Descriptor)
         attributes.";
      reference
        "Sec 1.4, RFC 4203: OSPF Extensions in Support of Generalized
         Multi-Protocol Label Switching (GMPLS). Section 1.4.";
      list max-lsp-bandwidth {
        key "priority";
        max-elements "8";
        description
          "Maximum LSP Bandwidth at priorities 0-7.";
        leaf priority {
          type uint8 {
            range "0..7";
          }
          description "Priority.";
        }
        uses te-types:te-bandwidth;
      }
    } // te-link-iscd-attributes

    grouping te-link-state-derived {
      description
        "Link state attributes in a TE topology.";
      leaf oper-status {
        type te-types:te-oper-status;
        config false;
        description
          "The current operational state of the link.";
      }
```

```
      leaf is-transitional {
        type empty;
        config false;
        description
          "Present if the link is transitional, used as an
           alternative approach in lieu of inter-layer-lock-id
           for path computation in a TE topology covering multiple
           layers or multiple regions.";
        reference
          "RFC 5212: Requirements for GMPLS-Based Multi-Region and
           Multi-Layer Networks (MRN/MLN).
           RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
           for Multi-Layer and Multi-Region Networks (MLN/MRN).";
      }
      uses information-source-per-link-attributes;
      list information-source-entry {
        key "information-source information-source-instance";
        config false;
        description
          "A list of information sources learned, including the one
           used.";
        uses information-source-per-link-attributes;
        uses te-link-info-attributes;
      }
      container recovery {
        config false;
        description
          "Status of the recovery process.";
        leaf restoration-status {
          type te-types:te-recovery-status;
          description
            "Restoration status.";
        }
        leaf protection-status {
          type te-types:te-recovery-status;
          description
            "Protection status.";
        }
      }
      container underlay {
        if-feature te-topology-hierarchy;
```

```
        config false;
        description "State attributes for te-link underlay.";
        leaf dynamic {
          type boolean;
          description
            "true if the underlay is dynamically created.";
        }
        leaf committed {
          type boolean;
          description
            "true if the underlay is committed.";
        }
      }
    }
  } // te-link-state-derived

  grouping te-link-underlay-attributes {
    description "Attributes for  te-link underlay.";
    reference
      "RFC 4206: Label Switched Paths (LSP) Hierarchy with
       Generalized Multi-Protocol Label Switching (GMPLS)
       Traffic Engineering (TE)";
    leaf enabled {
      type boolean;
      description
        "'true' if the underlay is enabled.
         'false' if the underlay is disabled.";
    }
    container primary-path {
      description
        "The service path on the underlay topology that
         supports this link.";
      uses nw:network-ref;
      list path-element {
        key "path-element-id";
        description
          "A list of path elements describing the service path.";
        leaf path-element-id {
          type uint32;
          description "To identify the element in a path.";
        }
        uses te-path-element;
```

```
          }
        } // primary-path
        list backup-path {
          key "index";
          description
            "A list of backup service paths on the underlay topology that
             protect the underlay primary path. If the primary path is
             not protected, the list contains zero elements. If the
             primary path is protected, the list contains one or more
             elements.";
          leaf index {
            type uint32;
            description
              "A sequence number to identify a backup path.";
          }
          uses nw:network-ref;
          list path-element {
            key "path-element-id";
            description
              "A list of path elements describing the backup service
               path";
            leaf path-element-id {
              type uint32;
              description "To identify the element in a path.";
            }
            uses te-path-element;
          }
        } // underlay-backup-path
        leaf protection-type {
          type identityref {
            base te-types:lsp-protection-type;
          }
          description
            "Underlay protection type desired for this link.";
        }
        container tunnel-termination-points {
          description
            "Underlay TTP(Tunnel Termination Points) desired for this
             link.";
          leaf source {
            type binary;
```

```
            description
              "Source tunnel termination point identifier.";
          }
          leaf destination {
            type binary;
            description
              "Destination tunnel termination point identifier.";
          }
        }
        container tunnels {
          description
            "Underlay TE tunnels supporting this TE link.";
          leaf sharing {
            type boolean;
            default true;
            description
              "'true' if the underlay tunnel can be shared with other
               TE links;
               'false' if the underlay tunnel is dedicated to this
               TE link.
               This leaf is the default option for all TE tunnels,
               and may be overridden by the per TE tunnel value.";
          }
          list tunnel {
            key "tunnel-name";
            description
              "Zero, one or more underlay TE tunnels that support this TE
               link.";
            leaf tunnel-name {
              type string;
              description
                "A tunnel name uniquely identifies an underlay TE tunnel,
                 used together with the source-node of this link.
                 The detailed information of this tunnel can be retrieved
                 from the ietf-te model.";
              reference "RFC 3209";
            }
            leaf sharing {
              type boolean;
              description
                "'true' if the underlay tunnel can be shared with other
```

```
                 TE links;
                 'false' if the underlay tunnel is dedicated to this
                 TE link.";
           }
         } // tunnel
       } // tunnels
     } // te-link-underlay-attributes

     grouping te-node-augment {
       description
         "Augmentation for TE node.";
       uses te-node-config;
       uses te-node-state-derived;
       container statistics {
         config false;
         description
           "Statistics data.";
         uses statistics-per-node;
       } // statistics

       list tunnel-termination-point {
         key "tunnel-tp-id";
         description
           "A termination point can terminate a tunnel.";
         leaf tunnel-tp-id {
           type binary;
           description
             "Tunnel termination point identifier.";
         }

         uses te-node-tunnel-termination-point-config;
         leaf oper-status {
           type te-types:te-oper-status;
           config false;
           description
             "The current operational state of the tunnel
              termination point.";
         }
         uses geolocation-container;
         container statistics {
           config false;
```

```
              description
                "Statistics data.";
              uses statistics-per-ttp;
            } // statistics

            // Relations to other tunnel termination points
            list supporting-tunnel-termination-point {
              key "node-ref tunnel-tp-ref";
              description
                "Identifies the tunnel termination points, that this
                 tunnel termination point is depending on.";
              leaf node-ref {
                type inet:uri;
                description
                  "This leaf identifies the node in which the supporting
                   tunnel termination point is present.
                   This node is either the supporting node or a node in
                   an underlay topology.";
              }
              leaf tunnel-tp-ref {
                type binary;
                description
                  "Reference to a tunnel terminiation point, which is
                   either in the supporting node or a node in an
                   underlay topology.";
              }
            } // supporting-tunnel-termination-point
          } // tunnel-termination-point
        } // te-node-augment

      grouping te-node-config {
        description "TE node configuration grouping.";
        leaf-list te-node-template {
          if-feature template;
          type leafref {
            path "../../../../te/templates/node-template/name";
          }
          description
            "The reference to a TE node template.";
        }
        uses te-node-config-attributes;
```

```
      } // te-node-config

      grouping te-node-config-attributes {
        description "Configuration node attributes in a TE topology.";
        container te-node-attributes {
          description "Containing node attributes in a TE topology.";
          leaf admin-status {
            type te-types:te-admin-status;
            description
              "The administrative state of the link.";
          }
          uses te-node-connectivity-matrices;
          uses te-node-info-attributes;
        } // te-node-attributes
      } // te-node-config-attributes

      grouping te-node-config-attributes-template {
        description
          "Configuration node attributes for template in a TE topology.";
        container te-node-attributes {
          description "Containing node attributes in a TE topology.";
          leaf admin-status {
            type te-types:te-admin-status;
            description
              "The administrative state of the link.";
          }
          uses te-node-info-attributes;
        } // te-node-attributes
      } // te-node-config-attributes-template

      grouping te-node-connectivity-matrices {
        description "Connectivity matrix on a TE node.";
        container connectivity-matrices {
          description
            "Containing connectivity matrix on a TE node.";
          leaf number-of-entries {
            type uint16;
            description
              "The number of connectivity matrix entries.
               If this number is specified in the configuration request,
               the number is requested number of entries, which may not
```

```
              all be listed in the list;
              if this number is reported in the state data,
              the number is the current number of operational entries.";
        }
        uses te-types:label-set-info;
        uses connectivity-matrix-entry-path-attributes;
        list connectivity-matrix {
          key "id";
          description
            "Represents node's switching limitations, i.e. limitations
             in interconnecting network TE links across the node.";
          reference
            "RFC 7579: General Network Element Constraint Encoding
             for GMPLS-Controlled Networks.";
          leaf id {
            type uint32;
            description "Identifies the connectivity-matrix entry.";
          }
        } // connectivity-matrix
      } // connectivity-matrices
    } // te-node-connectivity-matrices

    grouping te-node-connectivity-matrix-attributes {
      description
        "Termination point references of a connectivity matrix entry.";
      container from {
        description
          "Reference to source link termination point.";
        leaf tp-ref {
          type leafref {
            path "../../../../../../nt:termination-point/nt:tp-id";
          }
          description
            "Relative reference to a termination point.";
        }
        uses te-types:label-set-info;
      }
      container to {
        description
          "Reference to destination link termination point.";
        leaf tp-ref {
```

```
        type leafref {
          path "../../../../../../nt:termination-point/nt:tp-id";
        }
        description
          "Relative reference to a termination point.";
      }
      uses te-types:label-set-info;
    }
    uses connectivity-matrix-entry-path-attributes;
  } // te-node-connectivity-matrix-attributes

  grouping te-node-info-attributes {
    description
      "Advertised TE information attributes.";
    leaf domain-id {
      type uint32;
      description
        "Identifies the domain that this node belongs.
         This attribute is used to support inter-domain links.";
      reference
        "RFC 5152: A Per-Domain Path Computation Method for
         Establishing Inter-Domain Traffic Engineering (TE)
         Label Switched Paths (LSPs).
         RFC 5392: OSPF Extensions in Support of Inter-Autonomous
         System (AS) MPLS and GMPLS Traffic Engineering.
         RFC 5316: ISIS Extensions in Support of Inter-Autonomous
         System (AS) MPLS and GMPLS Traffic Engineering.";
    }
    leaf is-abstract {
      type empty;
      description
        "Present if the node is abstract, not present if the node
         is actual.";
    }
    leaf name {
      type string;
      description "Node name.";
    }
    leaf-list signaling-address {
      type inet:ip-address;
      description "Node signaling address.";
```

```
      }
      container underlay-topology {
        if-feature te-topology-hierarchy;
        description
          "When an abstract node encapsulates a topology,
           the attributes in this container point to said topology.";
        uses nw:network-ref;
      }
    } // te-node-info-attributes

    grouping te-node-state-derived {
      description "Node state attributes in a TE topology.";
      leaf oper-status {
        type te-types:te-oper-status;
        config false;
        description
          "The current operational state of the node.";
      }
      uses geolocation-container;
      leaf is-multi-access-dr {
        type empty;
        config false;
        description
          "The presence of this attribute indicates that this TE node
           is a pseudonode elected as a designated router.";
        reference
          "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
           Version 2.
           RFC 1195: Use of OSI IS-IS for Routing in TCP/IP and Dual
           Environments.";
      }
      uses information-source-per-node-attributes;
      list information-source-entry {
        key "information-source information-source-instance";
        config false;
        description
          "A list of information sources learned, including the one
           used.";
        uses information-source-per-node-attributes;
        uses te-node-connectivity-matrices;
        uses te-node-info-attributes;
```

```
      }
    } // te-node-state-derived

    grouping te-node-tunnel-termination-point-config {
      description
        "Termination capability of a tunnel termination point on a
         TE node.";
      uses te-node-tunnel-termination-point-config-attributes;
      container local-link-connectivities {
        description
          "Containing local link connectivity list for
           a tunnel termination point on a TE node.";
        leaf number-of-entries {
          type uint16;
          description
            "The number of local link connectivity list entries.
             If this number is specified in the configuration request,
             the number is requested number of entries, which may not
             all be listed in the list;
             if this number is reported in the state data,
             the number is the current number of operational entries.";
        }
        uses te-types:label-set-info;
        uses connectivity-matrix-entry-path-attributes;
      } // local-link-connectivities
    } // te-node-tunnel-termination-point-config

    grouping te-node-tunnel-termination-point-config-attributes {
      description
        "Configuration attributes of a tunnel termination point on a
         TE node.";
      leaf admin-status {
        type te-types:te-admin-status;
        description
          "The administrative state of the tunnel termination point.";
      }
      leaf name {
        type string;
        description
          "A descriptive name for the tunnel termination point.";
      }
```

```
        leaf switching-capability {
          type identityref {
            base te-types:switching-capabilities;
          }
          description
            "Switching Capability for this interface.";
        }
        leaf encoding {
          type identityref {
            base te-types:lsp-encoding-types;
          }
          description
            "Encoding supported by this interface.";
        }
        leaf-list inter-layer-lock-id {
          type uint32;
          description
            "Inter layer lock ID, used for path computation in a TE
             topology covering multiple layers or multiple regions.";
          reference
            "RFC 5212: Requirements for GMPLS-Based Multi-Region and
             Multi-Layer Networks (MRN/MLN).
             RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
             for Multi-Layer and Multi-Region Networks (MLN/MRN).";
        }
        leaf protection-type {
          type identityref {
            base te-types:lsp-protection-type;
          }
          description
            "The protection type that this tunnel termination point
             is capable of.";
        }

        container client-layer-adaptation {
          description
            "Containing capability information to support a client layer
             adaption in multi-layer topology.";
          list switching-capability {
            key "switching-capability encoding";
            description
```

```
            "List of supported switching capabilities";
          reference
            "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
             for Multi-Layer and Multi-Region Networks (MLN/MRN).
             RFC 4202: Routing Extensions in Support of
             Generalized Multi-Protocol Label Switching (GMPLS).";
          leaf switching-capability {
            type identityref {
              base te-types:switching-capabilities;
            }
            description
              "Switching Capability for the client layer adaption.";
          }
          leaf encoding {
            type identityref {
              base te-types:lsp-encoding-types;
            }
            description
              "Encoding supported by the client layer adaption.";
          }
          uses te-types:te-bandwidth;
        }
      }
    } // te-node-tunnel-termination-point-config-attributes

    grouping te-node-tunnel-termination-point-llc-list {
      description
        "Local link connectivity list of a tunnel termination
         point on a TE node.";
      list local-link-connectivity {
        key "link-tp-ref";
        description
          "The termination capabilities between
           tunnel-termination-point and link termination-point.
           The capability information can be used to compute
           the tunnel path.
           The Interface Adjustment Capability Descriptors (IACD)
           (defined in RFC 6001) on each link-tp can be derived from
           this local-link-connectivity list.";
        reference
          "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
```

```
            for Multi-Layer and Multi-Region Networks (MLN/MRN).";

        leaf link-tp-ref {
          type leafref {
            path "../../../../../nt:termination-point/nt:tp-id";
          }
          description
            "Link termination point.";
        }
        uses te-types:label-set-info;
        uses connectivity-matrix-entry-path-attributes;
      } // local-link-connectivity
    } // te-node-tunnel-termination-point-config

    grouping te-path-element {
      description
        "A group of attributes defining an element in a TE path
         such as TE node, TE link, TE atomic resource or label.";
      uses te-types:explicit-route-hop;
    } // te-path-element

    grouping te-termination-point-augment {
      description
        "Augmentation for TE termination point.";
      leaf te-tp-id {
        type te-types:te-tp-id;
        description
          "An identifier to uniquely identify a TE termination
           point.";
      }
      container te {
        must "../te-tp-id";
        presence "TE support.";
        description
          "Indicates TE support.";

        uses te-termination-point-config;
        leaf oper-status {
          type te-types:te-oper-status;
          config false;
          description
```

```
             "The current operational state of the link termination
              point.";
          }
          uses geolocation-container;
        } // te
      } // te-termination-point-augment

      grouping te-termination-point-config {
        description
          "TE termination point configuration grouping.";
        leaf admin-status {
          type te-types:te-admin-status;
          description
            "The administrative state of the link termination point.";
        }
        leaf name {
          type string;
          description
            "A descriptive name for the link termination point.";
        }
        uses interface-switching-capability-list;
        leaf inter-domain-plug-id {
          type binary;
          description
            "A topology-wide unique number that identifies on the
             network a connectivity supporting a given inter-domain
             TE link. This is more flexible alternative to specifying
             remote-te-node-id and remote-te-link-tp-id on a TE link,
             when the provider does not know remote-te-node-id and
             remote-te-link-tp-id or need to give client the
             flexibility to mix-n-match multiple topologies.";
        }
        leaf-list inter-layer-lock-id {
          type uint32;
          description
            "Inter layer lock ID, used for path computation in a TE
             topology covering multiple layers or multiple regions.";
          reference
            "RFC 5212: Requirements for GMPLS-Based Multi-Region and
             Multi-Layer Networks (MRN/MLN).
             RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions
```

```
            for Multi-Layer and Multi-Region Networks (MLN/MRN).";
      }
    } // te-termination-point-config

    grouping te-topologies-augment {
      description
        "Augmentation for TE topologies.";
      container te {
        presence "TE support.";
        description
          "Indicates TE support.";

        container templates {
          description
            "Configuration parameters for templates used for TE
             topology.";

          list node-template {
            if-feature template;
            key "name";
            leaf name {
              type te-types:te-template-name;
              description
                "The name to identify a TE node template.";
            }
            description
              "The list of TE node templates used to define sharable
               and reusable TE node attributes.";
            uses template-attributes;
            uses te-node-config-attributes-template;
          } // node-template

          list link-template {
            if-feature template;
            key "name";
            leaf name {
              type te-types:te-template-name;
              description
                "The name to identify a TE link template.";
            }
            description
```

```
              "The list of TE link templates used to define sharable
               and reusable TE link attributes.";
            uses template-attributes;
            uses te-link-config-attributes;
          } // link-template
        } // templates
      } // te
    } // te-topologies-augment

    grouping te-topology-augment {
      description
        "Augmentation for TE topology.";
      uses te-types:te-topology-identifier;

      container te {
        must "../te-topology-identifier/provider-id"
          + " and ../te-topology-identifier/client-id"
          + " and ../te-topology-identifier/topology-id";
        presence "TE support.";
        description
          "Indicates TE support.";

        uses te-topology-config;
        uses geolocation-container;
      } // te
    } // te-topology-augment

    grouping te-topology-config {
      description
        "TE topology configuration grouping.";
      leaf name {
        type string;
        description
          "Name of the TE topology. This attribute is optional and can
           be specified by the operator to describe the TE topology,
           which can be useful when network-id is not descriptive
           and not modifiable because of being generated by the
           system.";
      }
      leaf preference {
        type uint8 {
```

```
        range "1..255";
      }
      description
        "Specifies a preference for this topology. A lower number
         indicates a higher preference.";
    }
    leaf optimization-criterion {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Optimization criterion applied to this topology.";
      reference
        "RFC 3272: Overview and Principles of Internet Traffic
         Engineering.";
    }
    list nsrlg {
      if-feature nsrlg;
      key "id";
      description
        "List of NSRLGs (Not Sharing Risk Link Groups).";
      reference
        "RFC 4872: RSVP-TE Extensions in Support of End-to-End
         Generalized Multi-Protocol Label Switching (GMPLS)
         Recovery";
      leaf id {
        type uint32;
        description
          "Identify the NSRLG entry.";
      }
      leaf disjointness {
        type te-types:te-path-disjointness;
        description
          "The type of resource disjointness.";
      }
    } // nsrlg
  } // te-topology-config

  grouping template-attributes {
    description
      "Common attributes for all templates.";
```

```
      leaf priority {
        type uint16;
        description
          "The preference value to resolve conflicts between different
           templates. When two or more templates specify values for
           one configuration attribute, the value from the template
           with the highest priority is used.
           A lower number indicates a higher priority. The highest
           priority is 0.";
      }
      leaf reference-change-policy {
        type enumeration {
          enum no-action {
            description
              "When an attribute changes in this template, the
               configuration node referring to this template does
               not take any action.";
          }
          enum not-allowed {
            description
              "When any configuration object has a reference to this
               template, changing this template is not allowed.";
          }
          enum cascade {
            description
              "When an attribute changes in this template, the
               configuration object referring to this template applies
               the new attribute value to the corresponding
               configuration.";
          }
        }
        description
          "This attribute specifies the action taken to a configuration
           node that has a reference to this template.";
      }
    } // template-attributes

    /*
     * Data nodes
     */
    augment "/nw:networks/nw:network/nw:network-types" {
```

```
        description
          "Introduce new network type for TE topology.";
        container te-topology {
          presence "Indicates TE topology.";
          description
            "Its presence identifies the TE topology type.";
        }
      }

    augment "/nw:networks" {
      description
        "Augmentation parameters for TE topologies.";
      uses te-topologies-augment;
    }

    augment "/nw:networks/nw:network" {
      when "nw:network-types/tet:te-topology" {
        description
          "Augmentation parameters apply only for networks with
           TE topology type.";
      }
      description
        "Configuration parameters for TE topology.";
      uses te-topology-augment;
    }

    augment "/nw:networks/nw:network/nw:node" {
      when "../nw:network-types/tet:te-topology" {
        description
          "Augmentation parameters apply only for networks with
           TE topology type.";
      }
      description
        "Configuration parameters for TE at node level.";
      leaf te-node-id {
        type te-types:te-node-id;
        description
          "The identifier of a node in the TE topology.
           A node is specific to a topology to which it belongs.";
      }
      container te {
```

```
        must "../te-node-id" {
          description
            "te-node-id is mandatory.";
        }
        must "count(../nw:supporting-node)<=1" {
          description
            "For a node in a TE topology, there cannot be more
             than 1 supporting node. If multiple nodes are abstracted,
             the underlay-topology is used.";
        }
        presence "TE support.";
        description
          "Indicates TE support.";
        uses te-node-augment;
      } // te
    }

    augment "/nw:networks/nw:network/nt:link" {
      when "../nw:network-types/tet:te-topology" {
        description
          "Augmentation parameters apply only for networks with
           TE topology type.";
      }
      description
        "Configuration parameters for TE at link level.";
      container te {
        must "count(../nt:supporting-link)<=1" {
          description
            "For a link in a TE topology, there cannot be more
             than 1 supporting link. If one or more link paths are
             abstracted, the underlay is used.";
        }
        presence "TE support.";
        description
          "Indicates TE support.";
        uses te-link-augment;
      } // te
    }

    augment "/nw:networks/nw:network/nw:node/"
          + "nt:termination-point" {
```

```
        when "../../nw:network-types/tet:te-topology" {
          description
            "Augmentation parameters apply only for networks with
             TE topology type.";
        }
        description
          "Configuration parameters for TE at termination point level.";
        uses te-termination-point-augment;
      }

      augment
        "/nw:networks/nw:network/nt:link/te/bundle-stack-level/"
        + "bundle/bundled-links/bundled-link" {
        when "../../../../nw:network-types/tet:te-topology" {
          description
            "Augmentation parameters apply only for networks with
             TE topology type.";
        }
        description
          "Augment TE link bundled link.";
        leaf src-tp-ref {
          type leafref {
            path "../../../../../nw:node[nw:node-id = "
              + "current()/../../../../nt:source/"
              + "nt:source-node]/"
              + "nt:termination-point/nt:tp-id";
            require-instance true;
          }
          description
            "Reference to another TE termination point on the
             same source node.";
        }
        leaf des-tp-ref {
          type leafref {
            path "../../../../../nw:node[nw:node-id = "
              + "current()/../../../../nt:destination/"
              + "nt:dest-node]/"
              + "nt:termination-point/nt:tp-id";
            require-instance true;
          }
          description
```

```
      "Reference to another TE termination point on the
       same destination node.";
  }
}

augment
  "/nw:networks/nw:network/nw:node/te/"
  + "information-source-entry/connectivity-matrices/"
  + "connectivity-matrix" {
  when "../../../../../nw:network-types/tet:te-topology" {
    description
      "Augmentation parameters apply only for networks with
       TE topology type.";
  }
  description
    "Augment TE node connectivity-matrix.";
  uses te-node-connectivity-matrix-attributes;
}

augment
  "/nw:networks/nw:network/nw:node/te/te-node-attributes/"
  + "connectivity-matrices/connectivity-matrix" {
  when "../../../../../nw:network-types/tet:te-topology" {
    description
      "Augmentation parameters apply only for networks with
       TE topology type.";
  }
  description
    "Augment TE node connectivity-matrix.";
  uses te-node-connectivity-matrix-attributes;
}

augment
  "/nw:networks/nw:network/nw:node/te/"
  + "tunnel-termination-point/local-link-connectivities" {
  when "../../../../nw:network-types/tet:te-topology" {
    description
      "Augmentation parameters apply only for networks with
       TE topology type.";
  }
  description
```

```
          "Augment TE node tunnel termination point LLCs
          (Local Link Connectivities).";
        uses te-node-tunnel-termination-point-llc-list;
    }
  }
  <CODE ENDS>
```

8. Security Considerations

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

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

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

   o /nw:networks/nw:network/nw:network-types/tet:te-topology
      This subtree specifies the TE topology type. Modifying the
      configurations can make TE topology type invalid. By such
      modifications, a malicious attacker may disable the TE
      capabilities on the related networks and cause traffic disrupted
      or misrouted.

   o /nw:networks/tet:te
      This subtree specifies the TE node templates and TE link
      templates. Modifying the configurations in this subtree will
      change the related future TE configurations. By such
      modifications, a malicious attacker may change the TE capabilities
      scheduled at a future time, to cause traffic disrupted or
      misrouted.

o /nw:networks/nw:network
   This subtree specifies the topology-wide configurations, including
   the TE topology ID and topology-wide policies. Modifying the
   configurations in this subtree can add, remove, or modify TE
   topologies. By adding a TE topology, a malicious attacker may
   create an unauthorized traffic network. By removing or modifying a
   TE topology, a malicious attacker may cause traffic disabled or
   misrouted in the specified TE topology. Such traffic changes may
   also affect the traffic in the connected TE topologies.

o /nw:networks/nw:network/nw:node
   This subtree specifies the configurations for TE nodes. Modifying
   the configurations in this subtree can add, remove, or modify TE
   nodes. By adding a TE node, a malicious attacker may create an
   unauthorized traffic path. By removing or modifying a TE node, a
   malicious attacker may cause traffic disabled or misrouted in the
   specified TE node. Such traffic changes may also affect the
   traffic on the surrounding TE nodes and TE links in this TE
   topology and the connected TE topologies.

o /nw:networks/nw:network/nt:link/tet:te
   This subtree specifies the configurations for TE links. Modifying
   the configurations in this subtree can add, remove, or modify TE
   links. By adding a TE link, a malicious attacker may create an
   unauthorized traffic path. By removing or modifying a TE link, a
   malicious attacker may cause traffic disabled or misrouted on the
   specified TE link. Such traffic changes may also affect the
   traffic on the surrounding TE nodes and TE links in this TE
   topology and the connected TE topologies.

o /nw:networks/nw:network/nw:node/nt:termination-point
   This subtree specifies the configurations of TE link termination
   points. Modifying the configurations in this subtree can add,
   remove, or modify TE link termination points. By adding a TE link
   termination point, a malicious attacker may create an unauthorized
   traffic path. By removing or modifying a TE link termination
   point, a malicious attacker may cause traffic disabled or
   misrouted on the specified TE link termination point. Such traffic
   changes may also affect the traffic on the surrounding TE nodes
   and TE links in this TE topology and the connected TE topologies.

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

   o /nw:networks/nw:network/nw:network-types/tet:te-topology
      Unauthorized access to this subtree can disclose the TE topology
      type.

   o /nw:networks/tet:te
      Unauthorized access to this subtree can disclose the TE node
      templates and TE link templates.

   o /nw:networks/nw:network
      Unauthorized access to this subtree can disclose the topology-wide
      configurations, including the TE topology ID, the topology-wide
      policies, and the topology geolocation.

   o /nw:networks/nw:network/nw:node
      Unauthorized access to this subtree can disclose the operational
      state information of TE nodes.

   o /nw:networks/nw:network/nt:link/tet:te
      Unauthorized access to this subtree can disclose the operational
      state information of TE links.

   o /nw:networks/nw:network/nw:node/nt:termination-point
      Unauthorized access to this subtree can disclose the operational
      state information of TE link termination points.

9. IANA Considerations

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

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

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

   This document registers a YANG module in the YANG Module Names
   registry [RFC7950].

   name: ietf-te-topology
   namespace: urn:ietf:params:xml:ns:yang:ietf-te-topology
   prefix: tet
   reference: RFC XXXX

```
name:          ietf-te-topology-state
namespace:     urn:ietf:params:xml:ns:yang:ietf-te-topology-state
prefix:        tet-s
reference:     RFC XXXX
```

10. References

10.1. Normative References

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

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

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label
           Switching (GMPLS) Architecture", RFC 3945,
           DOI 10.17487/RFC3945, October 2004,
           <https://www.rfc-editor.org/info/rfc3945>.

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

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

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

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G.,
           Ceccarelli, D., and X. Zhang, "Problem Statement and
           Architecture for Information Exchange between
           Interconnected Traffic-Engineered Networks", BCP 206,
           RFC 7926, DOI 10.17487/RFC7926, July 2016,
           <https://www.rfc-editor.org/info/rfc7926>.

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

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

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

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

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

   [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N.,
              Ananthakrishnan, H., and X. Liu, "A YANG Data Model for
              Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March
              2018, <https://www.rfc-editor.org/info/rfc8345>.

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

   [I-D.ietf-teas-yang-te-types]
              Saad, T., Gandhi, R., Liu, X., Beeram, V., and
              I. Bryskin, "Traffic Engineering Common YANG Types",
              draft-ietf-teas-yang-te-types-08 (work in progress),
              April 2019.

10.2. Informative References

   [G.709]   ITU-T, "Interfaces for the optical transport network",
               ITU-T Recommendation G.709, June 2016.

   [G.805]   ITU-T, "Generic functional architecture of transport
               networks", ITU-T Recommendation G.805, March 2000.

   [G.872]   ITU-T, "Architecture of optical transport networks", ITU-T
              Recommendation G.872, January 2017.

   [G.8080]   ITU-T, "Architecture for the automatically switched
              optical network", ITU-T Recommendation G.8080, February
              2012.

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

   [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J.
              McManus, "Requirements for Traffic Engineering Over MPLS",
              RFC 2702, DOI 10.17487/RFC2702, September 1999,
              <https://www.rfc-editor.org/info/rfc2702>.

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

   [RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X.
              Xiao, "Overview and Principles of Internet Traffic
              Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002,
              <https://www.rfc-editor.org/info/rfc3272>.

   [RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label
              Switching (GMPLS) Signaling Functional Description",
              RFC 3471, DOI 10.17487/RFC3471, January 2003,
              <https://www.rfc-editor.org/info/rfc3471>.

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

   [RFC3785] Le Faucheur, F., Uppili, R., Vedrenne, A., Merckx, P., and
              T. Telkamp, "Use of Interior Gateway Protocol (IGP) Metric
              as a second MPLS Traffic Engineering (TE) Metric", BCP 87,
              RFC 3785, DOI 10.17487/RFC3785, May 2004,
              <https://www.rfc-editor.org/info/rfc3785>.

   [RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling
              in MPLS Traffic Engineering (TE)", RFC 4201,
              DOI 10.17487/RFC4201, October 2005,
              <https://www.rfc-editor.org/info/rfc4201>.

   [RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions
              in Support of Generalized Multi-Protocol Label Switching
              (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005,
              <https://www.rfc-editor.org/info/rfc4202>.

   [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in
              Support of Generalized Multi-Protocol Label Switching

              (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005,
              <https://www.rfc-editor.org/info/rfc4203>.

   [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP)
              Hierarchy with Generalized Multi-Protocol Label Switching
              (GMPLS) Traffic Engineering (TE)", RFC 4206,
              DOI 10.17487/RFC4206, October 2005,
              <https://www.rfc-editor.org/info/rfc4206>.

   [RFC4872] Lang, J., Ed., Rekhter, Y., Ed., and D. Papadimitriou,
              Ed., "RSVP-TE Extensions in Support of End-to-End
              Generalized Multi-Protocol Label Switching (GMPLS)
              Recovery", RFC 4872, DOI 10.17487/RFC4872, May 2007,
              <https://www.rfc-editor.org/info/rfc4872>.

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

   [RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux,
              M., and D. Brungard, "Requirements for GMPLS-Based Multi-
              Region and Multi-Layer Networks (MRN/MLN)", RFC 5212,
              DOI 10.17487/RFC5212, July 2008,
              <https://www.rfc-editor.org/info/rfc5212>.

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

   [RFC5316] Chen, M., Zhang, R., and X. Duan, "ISIS Extensions in
              Support of Inter-Autonomous System (AS) MPLS and GMPLS
              Traffic Engineering", RFC 5316, DOI 10.17487/RFC5316,
              December 2008, <https://www.rfc-editor.org/info/rfc5316>.

   [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, Ed.,
              "Traffic Engineering Extensions to OSPF Version 3",
              RFC 5329, DOI 10.17487/RFC5329, September 2008,
              <https://www.rfc-editor.org/info/rfc5329>.

   [RFC5392] Chen, M., Zhang, R., and X. Duan, "OSPF Extensions in
              Support of Inter-Autonomous System (AS) MPLS and GMPLS
              Traffic Engineering", RFC 5392, DOI 10.17487/RFC5392,
              January 2009, <https://www.rfc-editor.org/info/rfc5392>.

   [RFC6001] Papadimitriou, D., Vigoureux, M., Shiomoto, K., Brungard,

                D., and JL. Le Roux, "Generalized MPLS (GMPLS) Protocol
                Extensions for Multi-Layer and Multi-Region Networks (MLN/
                MRN)", RFC 6001, DOI 10.17487/RFC6001, October 2010,
                <https://www.rfc-editor.org/info/rfc6001>.

     [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS
                Traffic Engineering (MPLS-TE)", RFC 7308,
                DOI 10.17487/RFC7308, July 2014,
                <https://www.rfc-editor.org/info/rfc7308>.

     [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S.
                Previdi, "OSPF Traffic Engineering (TE) Metric
                Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015,
                <https://www.rfc-editor.org/info/rfc7471>.

     [RFC7579] Bernstein, G., Ed., Lee, Y., Ed., Li, D., Imajuku, W., and
                J. Han, "General Network Element Constraint Encoding for
                GMPLS-Controlled Networks", RFC 7579,
                DOI 10.17487/RFC7579, June 2015,
                <https://www.rfc-editor.org/info/rfc7579>.

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

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

     [I-D.ietf-netconf-subscribed-notifications]
                Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and
                A. Tripathy, "Customized Subscriptions to a Publisher's
                Event Streams", draft-ietf-netconf-subscribed-
                notifications-23 (work in progress), February 2019.

     [I-D.ietf-netconf-yang-push]
                Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-
                Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore
                Subscription", draft-ietf-netconf-yang-push-22 (work in
                progress), February 2019.

     [I-D.liu-netmod-yang-schedule]
                Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and
                O. Dios, "A YANG Data Model for Configuration Scheduling",
                draft-liu-netmod-yang-schedule-05 (work in progress),

                March 2018.

   [I-D.ietf-ccamp-wson-yang]
                Lee, Y., Dhody, D., Zhang, X., Guo, A., Lopezalvarez, V.,
                King, D., Yoon, B., and R. Vilata, "A Yang Data Model for
                WSON Optical Networks", draft-ietf-ccamp-wson-yang-20
                (work in progress), March 2019.

   [I-D.ietf-ccamp-otn-topo-yang]
                zhenghaomian@huawei.com, z., Guo, A., Busi, I., Sharma,
                A., Liu, X., Belotti, S., Xu, Y., Wang, L., and O. Dios,
                "A YANG Data Model for Optical Transport Network
                Topology", draft-ietf-ccamp-otn-topo-yang-06 (work in
                progress), February 2019.

   [I-D.ietf-teas-yang-l3-te-topo]
                Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and
                O. Dios, "YANG Data Model for Layer 3 TE Topologies",
                draft-ietf-teas-yang-l3-te-topo-04 (work in progress),
                March 2019.

   [I-D.ietf-teas-te-topo-and-tunnel-modeling]
                Bryskin, I., Beeram, V., Saad, T., and X. Liu, "TE
                Topology and Tunnel Modeling for Transport Networks",
                draft-ietf-teas-te-topo-and-tunnel-modeling-03 (work in
                progress), October 2018.

11. Acknowledgments

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.
```

```
      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