TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2019

T. Saad, Ed.
R. Gandhi
Cisco Systems Inc
X. Liu
Jabil
V. Beeram
Juniper Networks
H. Shah
Ciena
I. Bryskin
Huawei Technologies

July 01, 2018

A YANG Data Model for Traffic Engineering Tunnels and Interfaces
draft-ietf-teas-yang-te-16

Abstract

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

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

Status of This Memo

Table of Contents

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes the YANG data models for TE Tunnels, Label Switched Paths (LSPs) and TE interfaces that cover data applicable to generic or device-independent, device-specific, Multiprotocol Label Switching (MPLS) technology specific, and Segment Routing (SR) TE technology. It also describes helper modules that define TE grouping(s) and data types that can be imported by other modules.

The document defines the high-level relationship between the modules defined in this document, as well as other external protocol modules. It is expected other data plane technology model(s) will augment the TE generic model. Also, the TE generic model does not include any data specific to a signaling protocol. It is expected YANG models for TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) will augment the TE generic module.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

1.2. Tree Diagram

A simplified graphical representation of the data model is presented in each section of the model. The following notations are used for the YANG model data tree representation.

```
   <status> <flags> <name> <opts> <type>

    <status> is one of:
      +  for current
      x  for deprecated
      o  for obsolete

    <flags> is one of:
      rw  for read-write configuration data
      ro  for read-only non-configuration data
      -x  for execution rpcs
      -n  for notifications

    <name> is the name of the node

   If the node is augmented into the tree from another module, its name
   is printed as <prefix>:<name>

    <opts> is one of:
      ? for an optional leaf or node
      ! for a presence container
      * for a leaf-list or list
      Brackets [<keys>] for a list's keys
      Curly braces {<condition>} for optional feature that make node
   conditional
      Colon : for marking case nodes
      Ellipses ("...") subtree contents not shown

      Parentheses enclose choice and case nodes, and case nodes are also
      marked with a colon (":").

    <type> is the name of the type for leafs and leaf-lists.
```

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]
te	ietf-te	this document
te-types	ietf-te-types	this document
te-mpls-types	ietf-te-mpls-types	this document
te-dev	ietf-te-device	this document
te-mpls	ietf-te-mpls	this document
te-sr-mpls	ietf-te-sr-mpls	this document

Table 1: Prefixes and corresponding YANG modules

1.4. TE Technology Models

This document describes the generic TE YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technologies models to reuse the generic TE data model and possibly augment it with technology specific data model(s). There are multiple options being considered to achieve this:

o The generic TE model, including the lists of TE tunnels, LSPs, and interfaces can be defined and rooted at the top of the YANG tree. Specific leaf(s) under the TE tunnel, LSP, or interface, in this case, can identify the specific technology layer that it belongs to. This approach implies a single list for each of TE tunnel(s), LSP(s), and interface(s) in the model carries elements of different technology layers.

o An instance of the generic TE YANG model can be mounted in the YANG tree once for each TE technology layer(s). This approach provides separation of elements belonging to different technology layers into separate lists per layer in the data model.

o The generic TE data node(s) and TE list(s) for tunnels, LSPs, and interfaces are defined as grouping(s) in a separate module. The specific technology layer imports the generic TE groupings and uses them their respective technology specific module.

This revision of the model leverages the LSP encoding type of a tunnel (and interfaces) to identify the specific technology associated with the a TE interfaces, tunnel(s) and the LSP(s). For example, for an MPLS TE LSP, the LSP encoding type is assumed to be "lsp-encoding-packet".

Finally, the TE generic model does not include any signaling protocol data. It is expected that TE signaling protocol module(s) will be defined in other document(s) that will cover the RSVP-TE ([RFC3209], [RFC3473]), and Segment-Routing TE (SR-TE) model and that augment the TE generic model.

1.5. State Data Organization

Pure state data (for example, ephemeral or protocol derived state objects) can be modeled using one of the options below:

o Contained inside a read-write container, in a "state" sub-container, as shown in Figure 3

o Contained inside a separate read-only container, for example a lsps-state container

The Network Management Datastore Architecture (NMDA) addresses the "OpState" that was discussed in the IETF. As per NMDA guidelines for new models and models that are not concerned with the operational state of configuration information, this revision of the draft adopts the NMDA proposal for configuration and state data of this model.

2. Model Overview

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

2.1. Module(s) Relationship

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

The TE data relevant to a specific instantiations of data plane technology exists in a separate YANG module(s) that augment the TE generic model. For example, the MPLS-TE module "ietf-te-mpls.yang" is defined in Figure 10 and augments the TE generic model as shown in Figure 1. Similarly, the module "ietf-te-sr-mpls.yang" models the Segment Routing (SR) TE specific data and augments the TE generic and MPLS-TE model(s).

The TE data relevant to a TE specific signaling protocol instantiation is outside the scope and is covered in other documents. For example, the RSVP-TE [RFC3209] YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp], and other signaling protocol model(s) (e.g. for Segment-Routing TE) are expected to also augment the TE generic model.

```
                                        ^: import
     TE generic     +---------+         o: augment
     module         | ietf-te |o-------------+
                    +---------+               \
                       |   o  \                \
                       |   |\  \                \
                       |   | \  V                \
                       |   |  +----------------+  \
                       |   |  | ietf-te-device | TE device module
                       |   |  +----------------+    \
                       |   |       o        o        \
                       |   |     /           \        \
                       v   |   /              V        V
                    +--------------+         +---------------+
     RSVP-TE module | ietf-rsvp-te |o .       | ietf-te-mpls  |
                    +--------------+   \     +---------------+
                       ^                \     o
                       |                 \    +-----------------+
                       |                  \   | ietf-te-sr-mpls |
                       |                   \  +-----------------+
                       |                    \
                       o                +-------------------+
                    +-----------+       | ietf-rsvp-otn-te  |
     RSVP module    | ietf-rsvp |       +-------------------+
                    +-----------+          RSVP-TE with OTN
                                           extensions
                                          (shown for illustration
                                           not in this document)
```

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

```
             +---------+
             | ietf-te |       ^: import
             +---------+       o: augment
           import ^
                  |
                  |
            +---------------+
            | ietf-te-types |
            +---------------+
               o          o
               |           \
               |            \
       +-------------------+  +-------------------+
       | ietf-te-mpls-types |  | ietf-te-otn-types |
       +-------------------+  +-------------------+
                               (shown for illustration
                                not in this document)
```

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

2.2. Design Considerations

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

- o reusable data elements are grouped into separate TE types module(s) that can be readily imported by other modules whenever needed

- o reusable TE data types that are data plane independent are grouped in the TE generic types module "ietf-te-types.yang"

- o reusable TE data elements that are data plane specific (e.g. packet MPLS or switching technologies as defined in [RFC3473]) are expected to be grouped in a technology- specific types module, e.g. "ietf-te-mpls-types.yang". It is expected that technology specific types will augment TE generic types as shown in Figure 2

- o The TE generic model contains device independent data and can be used to model data off a device (e.g. on a controller). The TE data that is device-specific are grouped in a separate module as shown in Figure 1.

- o In general, little information in the model is designated as "mandatory", to allow freedom to vendors to adapt the data model to their specific product implementation.

2.3. Optional Features

Optional features that are beyond the base TE model are left to the specific vendor to decide support using vendor model augmentation and/or using feature checks.

This model declares a number of TE functions as features (such as P2MP-TE, soft-preemption etc.).

2.4. Configuration Inheritance

The defined data model supports configuration inheritance for tunnels, paths, and interfaces. Data elements defined in the main container (e.g. that encompasses the list of tunnels, interfaces, or paths) are assumed to apply equally to all elements of the list, unless overridden explicitly for a certain element of a list (e.g. a tunnel, interface or path).

3. TE Generic Model Organization

The TE generic model covers configuration, state, RPCs, and notifications data pertaining to TE global parameters, interfaces, tunnels and LSPs parameters that are device independent.

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

The model follows the guidelines in for modeling the intended, applied and derived state.

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

         +--rw tunnels
            .
            .

         +-- lsps-state

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

Figure 3: TE generic highlevel model view

3.1. Global Configuration and State Data

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

- Table of named SRLG mappings
- Table of named (extended) administrative groups mappings
- Table of named explicit paths to be referenced by TE tunnels
- Table of named path-constraints sets
- Auto-bandwidth global parameters
- TE diff-serve TE-class maps
- System-wide capabilities for LSP reoptimization (included in the TE device model)
  * Reoptimization timers (periodic interval, LSP installation and cleanup)
- System-wide capabilities for TE state flooding (included in the TE device model)

   * Periodic flooding interval

- Global capabilities that affect the originating, traversing and terminating LSPs. For example:

   * Path selection parameters (e.g. metric to optimize, etc.)

   * Path or segment protection parameters

The global state data is represented under the global "state" sub-container as shown in Figure 3.

Examples of such states are:

- Global statistics (signaling, admission, preemption, flooding)

- Global counters (number of tunnels/LSPs/interfaces)

3.2. Interfaces Configuration and State Data

This branch of the model covers configuration and state data items corresponding to TE interfaces that are present on a specific device. A new module is introduced that holds the TE device specific properties.

Examples of TE interface properties are:

- Maximum reservable bandwidth, bandwidth constraints (BC)

- Flooding parameters

   * Flooding intervals and threshold values

- Fast reroute backup tunnel properties (such as static, auto-tunnel)

- interface attributes

   * (Extended) administrative groups

   * SRLG values

   * TE metric value

The state corresponding to the TE interfaces applied configuration, protocol derived state, and stats and counters all fall under the interface "state" sub-container as shown in Figure 4 below:

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

Figure 4: TE interface state

This covers state data for TE interfaces such as:

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

3.3. Tunnels Configuration and State Data

This branch of the model covers intended, and corresponding applied configuration for tunnels. As well, it holds possible derived state pertaining to TE tunnels.

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

Figure 5: TE interface state tree

Examples of tunnel configuration date for TE tunnels:

- o Name and type (e.g. P2P, P2MP) of the TE tunnel
- o Admin-state
- o Set of primary and corresponding secondary paths
- o Routing usage (auto-route announce, forwarding adjacency)
- o Policy based routing (PBR) parameters

3.3.1. Tunnel Compute-Only Mode

By default, a configured TE tunnel is provisioned so it can carry traffic as soon as a valid path is computed and an LSP instantiated in the network. In other cases, a TE tunnel may be provisioned for computed path reporting purposes without the need to instantiate an LSP or commit resources in the network. In such a case, a tunnel configuration in "compute-only" mode to distinguish it from default tunnel behavior.

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

3.3.2. Tunnel Hierarchical Link Endpoint

TE LSPs can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used to form links to carry traffic in in other (client) networks [RFC6107]. In this case, the model introduces the TE tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the TE tunnel is associated with.

3.4. TE LSPs State Data

TE LSPs are derived state data that is usually instantiated via signaling protocols. TE LSPs exists on routers as ingress (starting point of LSP), transit (mid-point of LSP), or egress (termination point of the LSP). TE LSPs are distinguished by the 5 tuple, and LSP type (P2P or P2MP). In the model, the nodes holding LSPs data exist in the read-only lsps-state list as show in Figure 6.

3.5. Global RPC Data

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

- o Clear global TE statistics of various features

3.6. Interface RPC Data

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

- o Clear TE statistics for all or for individual TE interfaces
- o Trigger immediate flooding for one or all TE interfaces

3.7. Tunnel RPC Data

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

- o Clear statistics for all or for individual tunnels
- o Trigger the tear and setup of existing tunnels or LSPs.

3.8. Global Notifications Data

This branch of the model covers system-wide notifications data. The node notifies the registered events to the server using the defined notification messages.

3.9. Interfaces Notifications Data

This branch of the model covers TE interfaces related notifications data. The TE interface configuration is used for specific events registration. Notifications are sent for registered events to the server. Example events for TE interfaces are:

- o Interface creation and deletion
- o Interface state transitions
- o (Soft) preemption triggers
- o Fast reroute activation

3.10. Tunnel Notification Data

This branch of the model covers TE tunnels related notifications data. The TE tunnels configuration is used for specific events registration. Notifications are sent for registered events to the server. Example events for TE tunnels are:

o Tunnel creation and deletion events

o Tunnel state up/down changes

o Tunnel state reoptimization changes

Figure Figure 6 below shows the tree diagram of the YANG model defined in modules: ietf-te.yang, ietf-te-device.yang, ietf-te-mpls.yang, and ietf-te-sr.yang.

```
module: ietf-te
  +--rw te!
     +--rw globals
     |  +--rw named-admin-groups
     |  |  +--rw named-admin-group* [name]
     {te-types:extended-admin-groups,te-types:
     named-extended-admin-groups}?
     |  |     +--rw name            string
     |  |     +--rw bit-position?   uint32
     |  +--rw named-srlgs
     |  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
     |  |     +--rw name     string
     |  |     +--rw group?   te-types:srlg
     |  |     +--rw cost?    uint32
     |  +--rw named-path-constraints
     |  |  +--rw named-path-constraint* [name]
     {te-types:named-path-constraints}?
     |  |     +--rw name                             string
     |  |     +--rw te-bandwidth
     |  |     |  +--rw (technology)?
     |  |     |     +--:(generic)
     |  |     |        +--rw generic?   te-bandwidth
     |  |     +--rw 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
     |  |     |  +--rw constraints* [usage]
     |  |     |     +--rw usage                  identityref
     |  |     |     +--rw (style)?
     |  |     |        +--:(value)
     |  |     |        |  +--rw value?           te-types:admin-groups
     |  |     |        +--:(named)
     |  |     |           +--rw affinity-names* [name]
     |  |     |              +--rw name    string
     |  |     +--rw path-srlgs
     |  |     |  +--rw (style)?
     |  |     |     +--:(values)
     |  |     |     |  +--rw usage?         identityref
     |  |     |     |  +--rw values*        te-types:srlg
     |  |     |     +--:(named)
     |  |     |        +--rw constraints
     |  |     |           +--rw constraint* [usage]
     |  |     |              +--rw usage         identityref
     |  |     |              +--rw constraint
     |  |     |                 +--rw srlg-names* [name]
     |  |     |                    +--rw name    string
     |  |     +--rw explicit-route-objects
     |  |     |  +--rw route-object-exclude-always* [index]
     |  |     |  |  +--rw index                 uint32
     |  |     |  |  +--rw (type)?
     |  |     |  |     +--:(num-unnum-hop)
     |  |     |  |     |  +--rw num-unnum-hop
     |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |     |  |     |     +--rw direction?    te-link-direction
     |  |     |  |     +--:(as-number)
     |  |     |  |     |  +--rw as-number-hop
     |  |     |  |     |     +--rw as-number?   binary
     |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |     |  |     +--:(label)
     |  |     |  |        +--rw label-hop
     |  |     |  |           +--rw te-label
     |  |     |  |              +--rw (technology)?
     |  |     |  |              |  +--:(generic)
     |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |     |  |              +--rw direction?
     te-label-direction
     |  |     |  +--rw route-object-include-exclude* [index]
     |  |     |     +--rw explicit-route-usage?   identityref
     |  |     |     +--rw index                   uint32
     |  |     |     +--rw (type)?
     |  |     |        +--:(num-unnum-hop)
     |  |     |        |  +--rw num-unnum-hop
     |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |     |        |     +--rw hop-type?     te-hop-type
     |  |     |        |     +--rw direction?    te-link-direction
     |  |     |        +--:(as-number)
     |  |     |        |  +--rw as-number-hop
     |  |     |        |     +--rw as-number?   binary
     |  |     |        |     +--rw hop-type?    te-hop-type
     |  |     |        +--:(label)
     |  |     |        |  +--rw label-hop
     |  |     |        |     +--rw te-label
     |  |     |        |        +--rw (technology)?
     |  |     |        |        |  +--:(generic)
     |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |     |        |        +--rw direction?
     te-label-direction
     |  |     |        +--:(srlg)
     |  |     |           +--rw srlg
     |  |     |              +--rw srlg?   uint32
     |  |     +--rw shared-resources-tunnels
     |  |     |  +--rw lsp-shared-resources-tunnel*   te:tunnel-ref
     |  |     +--rw path-in-segment!
     |  |     |  +--rw forward
     |  |     |  |  +--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 range-bitmap?   binary
     |  |     |  +--rw reverse
     |  |     |     +--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 range-bitmap?   binary
     |  |     +--rw path-out-segment!
     |  |     |  +--rw forward
     |  |     |  |  +--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 range-bitmap?   binary
     |  |     |  +--rw reverse
     |  |     |     +--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 range-bitmap?   binary
     |  |     +--ro state
     |  |     |  +--ro bandwidth-generic_state?   te-types:te-bandwidth
     |  |     |  +--ro disjointness_state?
     te-types:te-path-disjointness
     |  |     +--rw te-mpls:bandwidth
     |  |     |  +--rw te-mpls:specification-type?
     te-mpls-types:te-bandwidth-type
     |  |     |  +--rw te-mpls:set-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     |  +--rw te-mpls:class-type?
     te-types:te-ds-class
     |  |     |  +--ro te-mpls:state
     |  |     |     +--ro te-mpls:signaled-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     +--rw te-sr-mpls:sid-selection-mode?
     te-sid-selection-mode
     |  |     +--rw te-sr-mpls:sid-protection?       identityref
     |  +--rw te-dev:lsp-install-interval?        uint32
     |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  +--rw te-dev:lsp-invalidation-interval?   uint32
     +--rw tunnels
     |  +--rw tunnel* [name]
     |  |  +--rw name                                 string
     |  |  +--rw identifier?                          uint16
     |  |  +--rw description?                         string
     |  |  +--rw encoding?                            identityref
     |  |  +--rw switching-type?                      identityref
     |  |  +--rw provisioning-state?                  identityref
     |  |  +--rw preference?                          uint8
     |  |  +--rw reoptimize-timer?                    uint16
     |  |  +--rw source?                              inet:ip-address
     |  |  +--rw destination?                         inet:ip-address
     |  |  +--rw src-tp-id?                            binary
     |  |  +--rw dst-tp-id?                            binary
     |  |  +--rw bidirectional?                        boolean
     |  |  +--rw association-objects
     |  |  |  +--rw association-object* [type ID source global-source]
     |  |  |  |  +--rw type             identityref
     |  |  |  |  +--rw ID               uint16
     |  |  |  |  +--rw source           inet:ip-address
     |  |  |  |  +--rw global-source    inet:ip-address
     |  |  |  +--rw association-object-extended* [type ID source
     global-source extended-ID]
     |  |  |     +--rw type             identityref
     |  |  |     +--rw ID               uint16
     |  |  |     +--rw source           inet:ip-address
     |  |  |     +--rw global-source    inet:ip-address
     |  |  |     +--rw extended-ID      binary
     |  |  +--rw protection
     |  |  |  +--rw enable?                           boolean
     |  |  |  +--rw protection-type?                  identityref
     |  |  |  +--rw protection-reversion-disable?     boolean
     |  |  |  +--rw hold-off-time?                    uint32
     |  |  |  +--rw wait-to-revert?                   uint16
     |  |  |  +--rw aps-signal-id?                    uint8
     |  |  +--rw restoration
     |  |  |  +--rw enable?                            boolean
     |  |  |  +--rw restoration-type?                  identityref
     |  |  |  +--rw restoration-scheme?                identityref
     |  |  |  +--rw restoration-reversion-disable?     boolean
     |  |  |  +--rw hold-off-time?                     uint32
     |  |  |  +--rw wait-to-restore?                   uint16
     |  |  |  +--rw wait-to-revert?                    uint16
     |  |  +--rw te-topology-identifier
     |  |  |  +--rw provider-id?   te-types:te-global-id
     |  |  |  +--rw client-id?     te-types:te-global-id
     |  |  |  +--rw topology-id?   te-types:te-topology-id
     |  |  +--rw te-bandwidth
     |  |  |  +--rw (technology)?
     |  |  |     +--:(generic)
     |  |  |        +--rw generic?   te-bandwidth
     |  |  +--rw setup-priority?                       uint8
     |  |  +--rw hold-priority?                        uint8
     |  |  +--rw signaling-type?                       identityref
     |  |  +--rw dependency-tunnels
     |  |  |  +--rw dependency-tunnel* [name]
     |  |  |     +--rw name              ->
     ../../../../../tunnels/tunnel/name
     |  |  |     +--rw encoding?         identityref
     |  |  |     +--rw switching-type?   identityref
     |  |  +--rw hierarchical-link
     |  |  |  +--rw local-te-node-id?         te-types:te-node-id
     |  |  |  +--rw local-te-link-tp-id?      te-types:te-tp-id
     |  |  |  +--rw remote-te-node-id?        te-types:te-node-id
     |  |  |  +--rw te-topology-identifier
     |  |  |     +--rw provider-id?   te-types:te-global-id
     |  |  |     +--rw client-id?     te-types:te-global-id
     |  |  |     +--rw topology-id?   te-types:te-topology-id
     |  |  +--ro state
     |  |  |  +--ro operational-state?                     identityref
     |  |  |  +--ro te-dev:lsp-install-interval?        uint32
     |  |  |  +--ro te-dev:lsp-cleanup-interval?        uint32
     |  |  |  +--ro te-dev:lsp-invalidation-interval?   uint32
     |  |  +--rw p2p-primary-paths
     |  |  |  +--rw p2p-primary-path* [name]
     |  |  |     +--rw name                               string
     |  |  |     +--rw path-setup-protocol?               identityref
     |  |  |     +--rw path-computation-method?           identityref
     |  |  |     +--rw path-computation-server?           inet:ip-address
     |  |  |     +--rw compute-only?                      empty
     |  |  |     +--rw use-path-computation?              boolean
     |  |  |     +--rw lockdown?                          empty
     |  |  |     +--rw path-scope?                        identityref
     |  |  |     +--rw optimizations
     |  |  |     |  +--rw (algorithm)?
     |  |  |     |     +--:(metric) {path-optimization-metric}?
     |  |  |     |     |  +--rw optimization-metric* [metric-type]
     |  |  |     |     |  |  +--rw metric-type
     identityref
     |  |  |     |     |  |  +--rw weight?
     uint8
     |  |  |     |     |  |  +--rw explicit-route-exclude-objects
     |  |  |     |     |  |  |  +--rw route-object-exclude-object*
     [index]
     |  |  |     |     |  |  |     +--rw index                   uint32
     |  |  |     |     |  |  |     +--rw (type)?
     |  |  |     |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |  |        +--:(as-number)
     |  |  |     |     |  |  |        |  +--rw as-number-hop
     |  |  |     |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |     |  |  |        |     +--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)?
     |  |  |     |     |  |           +--:(num-unnum-hop)
     |  |  |     |     |  |           |  +--rw num-unnum-hop
     |  |  |     |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |           +--:(as-number)
     |  |  |     |     |  |           |  +--rw as-number-hop
     |  |  |     |     |  |           |     +--rw as-number?   binary
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           +--:(label)
     |  |  |     |     |  |              +--rw label-hop
     |  |  |     |     |  |                 +--rw te-label
     |  |  |     |     |  |                    +--rw (technology)?
     |  |  |     |     |  |                    |  +--:(generic)
     |  |  |     |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |     |  +--rw tiebreakers
     |  |  |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |     |        +--rw tiebreaker-type    identityref
     |  |  |     |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |        +--rw objective-function
     |  |  |     |           +--rw objective-function-type?
     identityref
     |  |  |     +--rw preference?                     uint8
     |  |  |     +--rw named-path-constraint?          ->
     ../../../../../globals/named-path-constraints/
     named-path-constraint/name
     {te-types:named-path-constraints}?
     |  |  |     +--rw te-bandwidth
     |  |  |     |  +--rw (technology)?
     |  |  |     |     +--:(generic)
     |  |  |     |        +--rw generic?   te-bandwidth
     |  |  |     +--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
     |  |  |     |  +--rw constraints* [usage]
     |  |  |     |     +--rw usage                  identityref
     |  |  |     |     +--rw (style)?
     |  |  |     |        +--:(value)
     |  |  |     |        |  +--rw value?
     te-types:admin-groups
     |  |  |     |        +--:(named)
     |  |  |     |           +--rw affinity-names* [name]
     |  |  |     |              +--rw name    string
     |  |  |     +--rw path-srlgs
     |  |  |     |  +--rw (style)?
     |  |  |     |     +--:(values)
     |  |  |     |     |  +--rw usage?         identityref
     |  |  |     |     |  +--rw values*        te-types:srlg
     |  |  |     |     +--:(named)
     |  |  |     |        +--rw constraints
     |  |  |     |           +--rw constraint* [usage]
     |  |  |     |              +--rw usage         identityref
     |  |  |     |              +--rw constraint
     |  |  |     |                 +--rw srlg-names* [name]
     |  |  |     |                    +--rw name    string
     |  |  |     +--rw explicit-route-objects
     |  |  |     |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  +--rw index                  uint32
     |  |  |     |  |  +--rw (type)?
     |  |  |     |  |     +--:(num-unnum-hop)
     |  |  |     |  |     |  +--rw num-unnum-hop
     |  |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |     +--:(as-number)
     |  |  |     |  |     |  +--rw as-number-hop
     |  |  |     |  |     |     +--rw as-number?   binary
     |  |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |     +--:(label)
     |  |  |     |  |        +--rw label-hop
     |  |  |     |  |           +--rw te-label
     |  |  |     |  |              +--rw (technology)?
     |  |  |     |  |              |  +--:(generic)
     |  |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |              +--rw direction?
     te-label-direction
     |  |  |     |  +--rw route-object-include-exclude* [index]
     |  |  |     |     +--rw explicit-route-usage?   identityref
     |  |  |     |     +--rw index                   uint32
     |  |  |     |     +--rw (type)?
     |  |  |     |        +--:(num-unnum-hop)
     |  |  |     |        |  +--rw num-unnum-hop
     |  |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |        |     +--rw hop-type?     te-hop-type
     |  |  |     |        |     +--rw direction?    te-link-direction
     |  |  |     |        +--:(as-number)
     |  |  |     |        |  +--rw as-number-hop
     |  |  |     |        |     +--rw as-number?   binary
     |  |  |     |        |     +--rw hop-type?    te-hop-type
     |  |  |     |        +--:(label)
     |  |  |     |        |  +--rw label-hop
     |  |  |     |        |     +--rw te-label
     |  |  |     |        |        +--rw (technology)?
     |  |  |     |        |        |  +--:(generic)
     |  |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |        +--rw direction?
     te-label-direction
     |  |  |     |        +--:(srlg)
     |  |  |     |           +--rw srlg
     |  |  |     |              +--rw srlg?   uint32
     |  |  |     +--rw shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   te:tunnel-ref
     |  |  |     +--rw path-in-segment!
     |  |  |     |  +--rw forward
     |  |  |     |  |  +--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 range-bitmap?   binary
     |  |  |     |  +--rw reverse
     |  |  |     |     +--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 range-bitmap?   binary
     |  |  |     +--rw path-out-segment!
     |  |  |     |  +--rw forward
     |  |  |     |  |  +--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 range-bitmap?   binary
     |  |  |     |  +--rw reverse
     |  |  |     |     +--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 range-bitmap?   binary
     |  |  |     +--ro state
     |  |  |     |  +--ro path-properties
     |  |  |     |  |  +--ro path-metric* [metric-type]
     |  |  |     |  |  |  +--ro metric-type    -> ../state/metric-type
     |  |  |     |  |  |  +--ro state
     |  |  |     |  |  |     +--ro metric-type?          identityref
     |  |  |     |  |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |  +--ro path-affinities
     |  |  |     |  |  |  +--ro constraints* [usage]
     |  |  |     |  |  |     +--ro usage                   identityref
     |  |  |     |  |  |     +--ro (style)?
     |  |  |     |  |  |        +--:(value)
     |  |  |     |  |  |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |  |        +--:(named)
     |  |  |     |  |  |           +--ro affinity-names* [name]
     |  |  |     |  |  |              +--ro name    string
     |  |  |     |  |  +--ro path-srlgs
     |  |  |     |  |  |  +--ro (style)?
     |  |  |     |  |  |     +--:(values)
     |  |  |     |  |  |     |  +--ro usage?         identityref
     |  |  |     |  |  |     |  +--ro values*        te-types:srlg
     |  |  |     |  |  |     +--:(named)
     |  |  |     |  |  |        +--ro constraints
     |  |  |     |  |  |           +--ro constraint* [usage]
     |  |  |     |  |  |              +--ro usage         identityref
     |  |  |     |  |  |              +--ro constraint
     |  |  |     |  |  |                 +--ro srlg-names* [name]
     |  |  |     |  |  |                    +--ro name    string
     |  |  |     |  |  +--ro path-route-objects
     |  |  |     |  |  |  +--ro path-computed-route-object* [index]
     |  |  |     |  |  |     +--ro index    -> ../state/index
     |  |  |     |  |  |     +--ro state
     |  |  |     |  |  |        +--ro index?                uint32
     |  |  |     |  |  |        +--ro (type)?
     |  |  |     |  |  |           +--:(num-unnum-hop)
     |  |  |     |  |  |           |  +--ro num-unnum-hop
     |  |  |     |  |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |  |           |     +--ro hop-type?     te-hop-type
     |  |  |     |  |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |  |           +--:(as-number)
     |  |  |     |  |  |           |  +--ro as-number-hop
     |  |  |     |  |  |           |     +--ro as-number?   binary
     |  |  |     |  |  |           |     +--ro hop-type?    te-hop-type
     |  |  |     |  |  |           +--:(label)
     |  |  |     |  |  |              +--ro label-hop
     |  |  |     |  |  |                 +--ro te-label
     |  |  |     |  |  |                    +--ro (technology)?
     |  |  |     |  |  |                    |  +--:(generic)
     |  |  |     |  |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |  +--ro shared-resources-tunnels
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  +--ro lsps
     |  |  |     |  |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |  |  |     |  |     +--ro source
     inet:ip-address
     |  |  |     |  |     +--ro destination
     inet:ip-address
     |  |  |     |  |     +--ro tunnel-id
     uint16
     |  |  |     |  |     +--ro lsp-id
     uint16
     |  |  |     |  |     +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |     +--ro operational-state?
     identityref
     |  |  |     |  |     +--ro path-setup-protocol?
     identityref
     |  |  |     |  |     +--ro origin-type?
     enumeration
     |  |  |     |  |     +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |     +--ro lockout-of-normal?
     boolean
     |  |  |     |  |     +--ro freeze?
     boolean
     |  |  |     |  |     +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |     +--ro lsp-protection-state?
     identityref
     |  |  |     |  |     +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel?
     te:tunnel-ref
     |  |  |     |  |     +--ro lsp-record-route-subobjects
     |  |  |     |  |     |  +--ro record-route-subobject* [index]
     |  |  |     |  |     |     +--ro index               uint32
     |  |  |     |  |     |     +--ro (type)?
     |  |  |     |  |     |        +--:(numbered)
     |  |  |     |  |     |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |     |        |  +--ro ip-flags?     binary
     |  |  |     |  |     |        +--:(unnumbered)
     |  |  |     |  |     |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |        +--:(label)
     |  |  |     |  |     |           +--ro label-hop
     |  |  |     |  |     |              +--ro te-label
     |  |  |     |  |     |              |  +--ro (technology)?
     |  |  |     |  |     |              |  |  +--:(generic)
     |  |  |     |  |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |     |              +--ro label-flags?   binary
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |     |  +--ro path-metric* [metric-type]
     |  |  |     |  |     |  |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |     |  |  +--ro state
     |  |  |     |  |     |  |     +--ro metric-type?
     identityref
     |  |  |     |  |     |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |     |  +--ro path-affinities
     |  |  |     |  |     |  |  +--ro constraints* [usage]
     |  |  |     |  |     |  |     +--ro usage
     identityref
     |  |  |     |  |     |  |     +--ro (style)?
     |  |  |     |  |     |  |        +--:(value)
     |  |  |     |  |     |  |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |     |  |        +--:(named)
     |  |  |     |  |     |  |           +--ro affinity-names* [name]
     |  |  |     |  |     |  |              +--ro name    string
     |  |  |     |  |     |  +--ro path-srlgs
     |  |  |     |  |     |  |  +--ro (style)?
     |  |  |     |  |     |  |     +--:(values)
     |  |  |     |  |     |  |     |  +--ro usage?         identityref
     |  |  |     |  |     |  |     |  +--ro values*        te-types:srlg
     |  |  |     |  |     |  |     +--:(named)
     |  |  |     |  |     |  |        +--ro constraints
     |  |  |     |  |     |  |           +--ro constraint* [usage]
     |  |  |     |  |     |  |              +--ro usage
     identityref
     |  |  |     |  |     |  |              +--ro constraint
     |  |  |     |  |     |  |                 +--ro srlg-names* [name]
     |  |  |     |  |     |  |                    +--ro name    string
     |  |  |     |  |     |  +--ro path-route-objects
     |  |  |     |  |     |  |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |     |  |     +--ro index    -> ../state/index
     |  |  |     |  |     |  |     +--ro state
     |  |  |     |  |     |  |        +--ro index?
     uint32
     |  |  |     |  |     |  |        +--ro (type)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--ro num-unnum-hop
     |  |  |     |  |     |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--ro as-number-hop
     |  |  |     |  |     |  |           |     +--ro as-number?   binary
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--ro label-hop
     |  |  |     |  |     |  |                 +--ro te-label
     |  |  |     |  |     |  |                    +--ro (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |     |  +--ro shared-resources-tunnels
     |  |  |     |  |     |     +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  |     +--ro te-dev:lsp-timers
     |  |  |     |  |     |  +--ro te-dev:life-time?         uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-install?   uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-destroy?   uint32
     |  |  |     |  |     +--ro te-dev:downstream-info
     |  |  |     |  |     |  +--ro te-dev:nhop?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:outgoing-interface?
     if:interface-ref
     |  |  |     |  |     |  +--ro te-dev:neighbor?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  |     +--ro te-dev:upstream-info
     |  |  |     |  |        +--ro te-dev:phop?       inet:ip-address
     |  |  |     |  |        +--ro te-dev:neighbor?   inet:ip-address
     |  |  |     |  |        +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  +--ro te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  |     +--rw p2p-reverse-primary-path
     |  |  |     |  +--rw name?                          string
     |  |  |     |  +--rw path-setup-protocol?           identityref
     |  |  |     |  +--rw path-computation-method?       identityref
     |  |  |     |  +--rw path-computation-server?       inet:ip-address
     |  |  |     |  +--rw compute-only?                  empty
     |  |  |     |  +--rw use-path-computation?          boolean
     |  |  |     |  +--rw lockdown?                      empty
     |  |  |     |  +--rw path-scope?                    identityref
     |  |  |     |  +--rw optimizations
     |  |  |     |  |  +--rw (algorithm)?
     |  |  |     |  |     +--:(metric) {path-optimization-metric}?
     |  |  |     |  |     |  +--rw optimization-metric* [metric-type]
     |  |  |     |  |     |  |  +--rw metric-type
     identityref
     |  |  |     |  |     |  |  +--rw weight?
     uint8
     |  |  |     |  |     |  |  +--rw explicit-route-exclude-objects
     |  |  |     |  |     |  |  |  +--rw route-object-exclude-object*
     [index]
     |  |  |     |  |     |  |  |     +--rw index
     uint32
     |  |  |     |  |     |  |  |     +--rw (type)?
     |  |  |     |  |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |  |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |  |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |  |     |  |  |        +--:(as-number)
     |  |  |     |  |     |  |  |        |  +--rw as-number-hop
     |  |  |     |  |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |  |     |  |  |        |     +--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)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--rw num-unnum-hop
     |  |  |     |  |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--rw as-number-hop
     |  |  |     |  |     |  |           |     +--rw as-number?   binary
     |  |  |     |  |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--rw label-hop
     |  |  |     |  |     |  |                 +--rw te-label
     |  |  |     |  |     |  |                    +--rw (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |  |     |  +--rw tiebreakers
     |  |  |     |  |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |  |     |        +--rw tiebreaker-type    identityref
     |  |  |     |  |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |  |        +--rw objective-function
     |  |  |     |  |           +--rw objective-function-type?
     identityref
     |  |  |     |  +--rw named-path-constraint?        ->
     ../../../../../../globals/named-path-constraints/
     named-path-constraint/name
     {te-types:named-path-constraints}?
     |  |  |     |  +--rw te-bandwidth
     |  |  |     |  |  +--rw (technology)?
     |  |  |     |  |     +--:(generic)
     |  |  |     |  |        +--rw generic?   te-bandwidth
     |  |  |     |  +--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
     |  |  |     |  |  +--rw constraints* [usage]
     |  |  |     |  |     +--rw usage                   identityref
     |  |  |     |  |     +--rw (style)?
     |  |  |     |  |        +--:(value)
     |  |  |     |  |        |  +--rw value?
     te-types:admin-groups
     |  |  |     |  |        +--:(named)
     |  |  |     |  |           +--rw affinity-names* [name]
     |  |  |     |  |              +--rw name    string
     |  |  |     |  +--rw path-srlgs
     |  |  |     |  |  +--rw (style)?
     |  |  |     |  |     +--:(values)
     |  |  |     |  |     |  +--rw usage?         identityref
     |  |  |     |  |     |  +--rw values*        te-types:srlg
     |  |  |     |  |     +--:(named)
     |  |  |     |  |        +--rw constraints
     |  |  |     |  |           +--rw constraint* [usage]
     |  |  |     |  |              +--rw usage         identityref
     |  |  |     |  |              +--rw constraint
     |  |  |     |  |                 +--rw srlg-names* [name]
     |  |  |     |  |                    +--rw name    string
     |  |  |     |  +--rw explicit-route-objects
     |  |  |     |  |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  |  +--rw index                  uint32
     |  |  |     |  |  |  +--rw (type)?
     |  |  |     |  |  |     +--:(num-unnum-hop)
     |  |  |     |  |  |     |  +--rw num-unnum-hop
     |  |  |     |  |  |     |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |  |     +--:(as-number)
     |  |  |     |  |  |     |  +--rw as-number-hop
     |  |  |     |  |  |     |     +--rw as-number?   binary
     |  |  |     |  |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |  |     +--:(label)
     |  |  |     |  |  |        +--rw label-hop
     |  |  |     |  |  |           +--rw te-label
     |  |  |     |  |  |              +--rw (technology)?
     |  |  |     |  |  |              |  +--:(generic)
     |  |  |     |  |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |  |              +--rw direction?
     te-label-direction
     |  |  |     |  |  +--rw route-object-include-exclude* [index]
     |  |  |     |  |     +--rw explicit-route-usage?   identityref
     |  |  |     |  |     +--rw index                   uint32
     |  |  |     |  |     +--rw (type)?
     |  |  |     |  |        +--:(num-unnum-hop)
     |  |  |     |  |        |  +--rw num-unnum-hop
     |  |  |     |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |  |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |        |     +--rw hop-type?     te-hop-type
     |  |  |     |  |        |     +--rw direction?    te-link-direction
     |  |  |     |  |        +--:(as-number)
     |  |  |     |  |        |  +--rw as-number-hop
     |  |  |     |  |        |     +--rw as-number?   binary
     |  |  |     |  |        |     +--rw hop-type?    te-hop-type
     |  |  |     |  |        +--:(label)
     |  |  |     |  |        |  +--rw label-hop
     |  |  |     |  |        |     +--rw te-label
     |  |  |     |  |        |        +--rw (technology)?
     |  |  |     |  |        |        |  +--:(generic)
     |  |  |     |  |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |        |        +--rw direction?
     te-label-direction
     |  |  |     |  |        +--:(srlg)
     |  |  |     |  |           +--rw srlg
     |  |  |     |  |              +--rw srlg?   uint32
     |  |  |     |  +--rw shared-resources-tunnels
     |  |  |     |  |  +--rw lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  +--rw path-in-segment!
     |  |  |     |  |  +--rw forward
     |  |  |     |  |  |  +--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 range-bitmap?   binary
     |  |  |     |  |  +--rw reverse
     |  |  |     |  |     +--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 range-bitmap?   binary
     |  |  |     |  +--rw path-out-segment!
     |  |  |     |  |  +--rw forward
     |  |  |     |  |  |  +--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 range-bitmap?   binary
     |  |  |     |  |  +--rw reverse
     |  |  |     |  |     +--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 range-bitmap?   binary
     |  |  |     |  +--ro state
     |  |  |     |  |  +--ro path-properties
     |  |  |     |  |  |  +--ro path-metric* [metric-type]
     |  |  |     |  |  |  |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |  |  |  +--ro state
     |  |  |     |  |  |  |     +--ro metric-type?          identityref
     |  |  |     |  |  |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |  |  +--ro path-affinities
     |  |  |     |  |  |  |  +--ro constraints* [usage]
     |  |  |     |  |  |  |     +--ro usage
     identityref
     |  |  |     |  |  |  |     +--ro (style)?
     |  |  |     |  |  |  |        +--:(value)
     |  |  |     |  |  |  |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |  |  |        +--:(named)
     |  |  |     |  |  |  |           +--ro affinity-names* [name]
     |  |  |     |  |  |  |              +--ro name    string
     |  |  |     |  |  |  +--ro path-srlgs
     |  |  |     |  |  |  |  +--ro (style)?
     |  |  |     |  |  |  |     +--:(values)
     |  |  |     |  |  |  |     |  +--ro usage?         identityref
     |  |  |     |  |  |  |     |  +--ro values*        te-types:srlg
     |  |  |     |  |  |  |     +--:(named)
     |  |  |     |  |  |  |        +--ro constraints
     |  |  |     |  |  |  |           +--ro constraint* [usage]
     |  |  |     |  |  |  |              +--ro usage         identityref
     |  |  |     |  |  |  |              +--ro constraint
     |  |  |     |  |  |  |                 +--ro srlg-names* [name]
     |  |  |     |  |  |  |                    +--ro name    string
     |  |  |     |  |  |  +--ro path-route-objects
     |  |  |     |  |  |  |  +--ro path-computed-route-object* [index]
     |  |  |     |  |  |  |     +--ro index    -> ../state/index
     |  |  |     |  |  |  |     +--ro state
     |  |  |     |  |  |  |        +--ro index?                uint32
     |  |  |     |  |  |  |        +--ro (type)?
     |  |  |     |  |  |  |           +--:(num-unnum-hop)
     |  |  |     |  |  |  |           |  +--ro num-unnum-hop
     |  |  |     |  |  |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |  |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |  |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |  |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |  |  |           +--:(as-number)
     |  |  |     |  |  |  |           |  +--ro as-number-hop
     |  |  |     |  |  |  |           |     +--ro as-number?   binary
     |  |  |     |  |  |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |  |  |           +--:(label)
     |  |  |     |  |  |  |              +--ro label-hop
     |  |  |     |  |  |  |                 +--ro te-label
     |  |  |     |  |  |  |                    +--ro (technology)?
     |  |  |     |  |  |  |                    |  +--:(generic)
     |  |  |     |  |  |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |  |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |  |  +--ro shared-resources-tunnels
     |  |  |     |  |  |     +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  |  +--ro lsps
     |  |  |     |  |     +--ro lsp* [source destination tunnel-id
     lsp-id extended-tunnel-id]
     |  |  |     |  |        +--ro source
     inet:ip-address
     |  |  |     |  |        +--ro destination
     inet:ip-address
     |  |  |     |  |        +--ro tunnel-id
     uint16
     |  |  |     |  |        +--ro lsp-id
     uint16
     |  |  |     |  |        +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |        +--ro operational-state?
     identityref
     |  |  |     |  |        +--ro path-setup-protocol?
     identityref
     |  |  |     |  |        +--ro origin-type?
     enumeration
     |  |  |     |  |        +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |        +--ro lockout-of-normal?
     boolean
     |  |  |     |  |        +--ro freeze?
     boolean
     |  |  |     |  |        +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |        +--ro lsp-protection-state?
     identityref
     |  |  |     |  |        +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |        +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |        +--ro lsp-shared-resources-tunnel?
     te:tunnel-ref
     |  |  |     |  |        +--ro lsp-record-route-subobjects
     |  |  |     |  |        |  +--ro record-route-subobject* [index]
     |  |  |     |  |        |     +--ro index               uint32
     |  |  |     |  |        |     +--ro (type)?
     |  |  |     |  |        |        +--:(numbered)
     |  |  |     |  |        |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |        |        |  +--ro ip-flags?     binary
     |  |  |     |  |        |        +--:(unnumbered)
     |  |  |     |  |        |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |        |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |        |        +--:(label)
     |  |  |     |  |        |           +--ro label-hop
     |  |  |     |  |        |              +--ro te-label
     |  |  |     |  |        |              |  +--ro (technology)?
     |  |  |     |  |        |              |  |  +--:(generic)
     |  |  |     |  |        |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |        |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |        |              +--ro label-flags?   binary
     |  |  |     |  |       +--ro path-properties
     |  |  |     |  |          +--ro path-metric* [metric-type]
     |  |  |     |  |          |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |          |  +--ro state
     |  |  |     |  |          |     +--ro metric-type?
     identityref
     |  |  |     |  |          |     +--ro accumulative-value?   uint64
     |  |  |     |  |          +--ro path-affinities
     |  |  |     |  |          |  +--ro constraints* [usage]
     |  |  |     |  |          |     +--ro usage
     identityref
     |  |  |     |  |          |     +--ro (style)?
     |  |  |     |  |          |        +--:(value)
     |  |  |     |  |          |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |          |        +--:(named)
     |  |  |     |  |          |           +--ro affinity-names* [name]
     |  |  |     |  |          |              +--ro name    string
     |  |  |     |  |          +--ro path-srlgs
     |  |  |     |  |          |  +--ro (style)?
     |  |  |     |  |          |     +--:(values)
     |  |  |     |  |          |     |  +--ro usage?
     identityref
     |  |  |     |  |          |     |  +--ro values*
     te-types:srlg
     |  |  |     |  |          |     +--:(named)
     |  |  |     |  |          |        +--ro constraints
     |  |  |     |  |          |           +--ro constraint* [usage]
     |  |  |     |  |          |              +--ro usage
     identityref
     |  |  |     |  |          |              +--ro constraint
     |  |  |     |  |          |                 +--ro srlg-names*
     [name]
     |  |  |     |  |          |                    +--ro name
     string
     |  |  |     |  |          +--ro path-route-objects
     |  |  |     |  |           |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |           |     +--ro index    -> ../state/index
     |  |  |     |  |           |     +--ro state
     |  |  |     |  |           |        +--ro index?
     uint32
     |  |  |     |  |           |        +--ro (type)?
     |  |  |     |  |           |           +--:(num-unnum-hop)
     |  |  |     |  |           |           |  +--ro num-unnum-hop
     |  |  |     |  |           |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |           |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |           |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |           |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |           |           +--:(as-number)
     |  |  |     |  |           |           |  +--ro as-number-hop
     |  |  |     |  |           |           |     +--ro as-number?
     binary
     |  |  |     |  |           |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |           |           +--:(label)
     |  |  |     |  |           |              +--ro label-hop
     |  |  |     |  |           |                 +--ro te-label
     |  |  |     |  |           |                    +--ro (technology)?
     |  |  |     |  |           |                    |  +--:(generic)
     |  |  |     |  |           |                    |     +--ro
     generic?   rt-types:generalized-label
     |  |  |     |  |           |                    +--ro direction?
     te-label-direction
     |  |  |     |  |           +--ro shared-resources-tunnels
     |  |  |     |  |              +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  +--rw p2p-reverse-secondary-path
     |  |  |     |     +--rw secondary-path?        ->
     ../../../../../p2p-secondary-paths/p2p-secondary-path/name
     |  |  |     |     +--rw path-setup-protocol?   identityref
     |  |  |     +--rw candidate-p2p-secondary-paths
     |  |  |     |  +--rw candidate-p2p-secondary-path* [secondary-path]
     |  |  |     |     +--rw secondary-path         ->
     ../../../../../p2p-secondary-paths/p2p-secondary-path/name
     |  |  |     |     +--rw path-setup-protocol?   identityref
     |  |  |     |     +--ro state
     |  |  |     |        +--ro active?   boolean
     |  |  |     +--rw te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  +--rw p2p-secondary-paths
     |  |  |  +--rw p2p-secondary-path* [name]
     |  |  |     +--rw name                          string
     |  |  |     +--rw path-setup-protocol?          identityref
     |  |  |     +--rw path-computation-method?      identityref
     |  |  |     +--rw path-computation-server?      inet:ip-address
     |  |  |     +--rw compute-only?                 empty
     |  |  |     +--rw use-path-computation?         boolean
     |  |  |     +--rw lockdown?                     empty
     |  |  |     +--rw path-scope?                   identityref
     |  |  |     +--rw optimizations
     |  |  |     |  +--rw (algorithm)?
     |  |  |     |     +--:(metric) {path-optimization-metric}?
     |  |  |     |     |  +--rw optimization-metric* [metric-type]
     |  |  |     |     |  |  +--rw metric-type
     identityref
     |  |  |     |     |  |  +--rw weight?
     uint8
     |  |  |     |     |  |  +--rw explicit-route-exclude-objects
     |  |  |     |     |  |  |  +--rw route-object-exclude-object*
     [index]
     |  |  |     |     |  |  |     +--rw index                 uint32
     |  |  |     |     |  |  |     +--rw (type)?
     |  |  |     |     |  |  |        +--:(num-unnum-hop)
     |  |  |     |     |  |  |        |  +--rw num-unnum-hop
     |  |  |     |     |  |  |        |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |  |        |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |  |        |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |  |        |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |  |        +--:(as-number)
     |  |  |     |     |  |  |        |  +--rw as-number-hop
     |  |  |     |     |  |  |        |     +--rw as-number?   binary
     |  |  |     |     |  |  |        |     +--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)?
     |  |  |     |     |  |           +--:(num-unnum-hop)
     |  |  |     |     |  |           |  +--rw num-unnum-hop
     |  |  |     |     |  |           |     +--rw node-id?
     te-types:te-node-id
     |  |  |     |     |  |           |     +--rw link-tp-id?
     te-types:te-tp-id
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           |     +--rw direction?
     te-link-direction
     |  |  |     |     |  |           +--:(as-number)
     |  |  |     |     |  |           |  +--rw as-number-hop
     |  |  |     |     |  |           |     +--rw as-number?   binary
     |  |  |     |     |  |           |     +--rw hop-type?
     te-hop-type
     |  |  |     |     |  |           +--:(label)
     |  |  |     |     |  |              +--rw label-hop
     |  |  |     |     |  |                 +--rw te-label
     |  |  |     |     |  |                    +--rw (technology)?
     |  |  |     |     |  |                    |  +--:(generic)
     |  |  |     |     |  |                    |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |     |  |                    +--rw direction?
     te-label-direction
     |  |  |     |     |  +--rw tiebreakers
     |  |  |     |     |     +--rw tiebreaker* [tiebreaker-type]
     |  |  |     |     |        +--rw tiebreaker-type    identityref
     |  |  |     |     +--:(objective-function)
     {path-optimization-objective-function}?
     |  |  |     |        +--rw objective-function
     |  |  |     |           +--rw objective-function-type?
     identityref
     |  |  |     +--rw preference?                 uint8
     |  |  |     +--rw named-path-constraint?      ->
     ../../../../../globals/named-path-constraints/
     named-path-constraint/name
     {te-types:named-path-constraints}?
     |  |  |     +--rw te-bandwidth
     |  |  |     |  +--rw (technology)?
     |  |  |     |     +--:(generic)
     |  |  |     |        +--rw generic?   te-bandwidth
     |  |  |     +--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
     |  |  |     |  +--rw constraints* [usage]
     |  |  |     |     +--rw usage                    identityref
     |  |  |     |     +--rw (style)?
     |  |  |     |        +--:(value)
     |  |  |     |        |  +--rw value?
     te-types:admin-groups
     |  |  |     |        +--:(named)
     |  |  |     |           +--rw affinity-names* [name]
     |  |  |     |              +--rw name    string
     |  |  |     +--rw path-srlgs
     |  |  |     |  +--rw (style)?
     |  |  |     |     +--:(values)
     |  |  |     |     |  +--rw usage?         identityref
     |  |  |     |     |  +--rw values*        te-types:srlg
     |  |  |     |     +--:(named)
     |  |  |     |        +--rw constraints
     |  |  |     |           +--rw constraint* [usage]
     |  |  |     |              +--rw usage         identityref
     |  |  |     |              +--rw constraint
     |  |  |     |                 +--rw srlg-names* [name]
     |  |  |     |                    +--rw name    string
     |  |  |     +--rw explicit-route-objects
     |  |  |     |  +--rw route-object-exclude-always* [index]
     |  |  |     |  |  +--rw index                    uint32
     |  |  |     |  |  +--rw (type)?
     |  |  |     |  |     +--:(num-unnum-hop)
     |  |  |     |  |     |  +--rw num-unnum-hop
     |  |  |     |  |     |     +--rw node-id?      te-types:te-node-id
     |  |  |     |  |     |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |  |     |     +--rw hop-type?     te-hop-type
     |  |  |     |  |     |     +--rw direction?    te-link-direction
     |  |  |     |  |     +--:(as-number)
     |  |  |     |  |     |  +--rw as-number-hop
     |  |  |     |  |     |     +--rw as-number?   binary
     |  |  |     |  |     |     +--rw hop-type?    te-hop-type
     |  |  |     |  |     +--:(label)
     |  |  |     |  |        +--rw label-hop
     |  |  |     |  |           +--rw te-label
     |  |  |     |  |              +--rw (technology)?
     |  |  |     |  |              |  +--:(generic)
     |  |  |     |  |              |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |  |              +--rw direction?
     te-label-direction
     |  |  |     |  +--rw route-object-include-exclude* [index]
     |  |  |     |     +--rw explicit-route-usage?   identityref
     |  |  |     |     +--rw index                   uint32
     |  |  |     |     +--rw (type)?
     |  |  |     |        +--:(num-unnum-hop)
     |  |  |     |        |  +--rw num-unnum-hop
     |  |  |     |        |     +--rw node-id?      te-types:te-node-id
     |  |  |     |        |     +--rw link-tp-id?   te-types:te-tp-id
     |  |  |     |        |     +--rw hop-type?     te-hop-type
     |  |  |     |        |     +--rw direction?    te-link-direction
     |  |  |     |        +--:(as-number)
     |  |  |     |        |  +--rw as-number-hop
     |  |  |     |        |     +--rw as-number?   binary
     |  |  |     |        |     +--rw hop-type?    te-hop-type
     |  |  |     |        +--:(label)
     |  |  |     |        |  +--rw label-hop
     |  |  |     |        |     +--rw te-label
     |  |  |     |        |        +--rw (technology)?
     |  |  |     |        |        |  +--:(generic)
     |  |  |     |        |        |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |        +--rw direction?
     te-label-direction
     |  |  |     |        +--:(srlg)
     |  |  |     |           +--rw srlg
     |  |  |     |              +--rw srlg?   uint32
     |  |  |     +--rw shared-resources-tunnels
     |  |  |     |  +--rw lsp-shared-resources-tunnel*   te:tunnel-ref
     |  |  |     +--rw path-in-segment!
     |  |  |     |  +--rw forward
     |  |  |     |  |  +--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 range-bitmap?   binary
     |  |  |     |  +--rw reverse
     |  |  |     |     +--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 range-bitmap?   binary
     |  |  |     +--rw path-out-segment!
     |  |  |     |  +--rw forward
     |  |  |     |  |  +--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 range-bitmap?   binary
     |  |  |     |  +--rw reverse
     |  |  |     |     +--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 range-bitmap?   binary
     |  |  |     +--rw disjointness?
     te-types:te-path-disjointness
     |  |  |     +--rw protection
     |  |  |     |  +--rw enable?                        boolean
     |  |  |     |  +--rw protection-type?               identityref
     |  |  |     |  +--rw protection-reversion-disable?   boolean
     |  |  |     |  +--rw hold-off-time?                 uint32
     |  |  |     |  +--rw wait-to-revert?                uint16
     |  |  |     |  +--rw aps-signal-id?                 uint8
     |  |  |     +--rw restoration
     |  |  |     |  +--rw enable?                         boolean
     |  |  |     |  +--rw restoration-type?               identityref
     |  |  |     |  +--rw restoration-scheme?             identityref
     |  |  |     |  +--rw restoration-reversion-disable?   boolean
     |  |  |     |  +--rw hold-off-time?                  uint32
     |  |  |     |  +--rw wait-to-restore?                uint16
     |  |  |     |  +--rw wait-to-revert?                 uint16
     |  |  |     +--ro state
     |  |  |     |  +--ro path-properties
     |  |  |     |  |  +--ro path-metric* [metric-type]
     |  |  |     |  |  |  +--ro metric-type    -> ../state/metric-type
     |  |  |     |  |  |  +--ro state
     |  |  |     |  |  |     +--ro metric-type?          identityref
     |  |  |     |  |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |  +--ro path-affinities
     |  |  |     |  |  |  +--ro constraints* [usage]
     |  |  |     |  |  |     +--ro usage                   identityref
     |  |  |     |  |  |     +--ro (style)?
     |  |  |     |  |  |        +--:(value)
     |  |  |     |  |  |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |  |        +--:(named)
     |  |  |     |  |  |           +--ro affinity-names* [name]
     |  |  |     |  |  |              +--ro name    string
     |  |  |     |  |  +--ro path-srlgs
     |  |  |     |  |  |  +--ro (style)?
     |  |  |     |  |  |     +--:(values)
     |  |  |     |  |  |     |  +--ro usage?         identityref
     |  |  |     |  |  |     |  +--ro values*        te-types:srlg
     |  |  |     |  |  |     +--:(named)
     |  |  |     |  |  |        +--ro constraints
     |  |  |     |  |  |           +--ro constraint* [usage]
     |  |  |     |  |  |              +--ro usage         identityref
     |  |  |     |  |  |              +--ro constraint
     |  |  |     |  |  |                 +--ro srlg-names* [name]
     |  |  |     |  |  |                    +--ro name    string
     |  |  |     |  |  +--ro path-route-objects
     |  |  |     |  |  |  +--ro path-computed-route-object* [index]
     |  |  |     |  |  |     +--ro index    -> ../state/index
     |  |  |     |  |  |     +--ro state
     |  |  |     |  |  |        +--ro index?                uint32
     |  |  |     |  |  |        +--ro (type)?
     |  |  |     |  |  |           +--:(num-unnum-hop)
     |  |  |     |  |  |           |  +--ro num-unnum-hop
     |  |  |     |  |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |  |           |     +--ro hop-type?     te-hop-type
     |  |  |     |  |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |  |           +--:(as-number)
     |  |  |     |  |  |           |  +--ro as-number-hop
     |  |  |     |  |  |           |     +--ro as-number?   binary
     |  |  |     |  |  |           |     +--ro hop-type?    te-hop-type
     |  |  |     |  |  |           +--:(label)
     |  |  |     |  |  |              +--ro label-hop
     |  |  |     |  |  |                 +--ro te-label
     |  |  |     |  |  |                    +--ro (technology)?
     |  |  |     |  |  |                    |  +--:(generic)
     |  |  |     |  |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |  +--ro shared-resources-tunnels
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  +--ro lsps
     |  |  |     |  |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |  |  |     |  |     +--ro source
     inet:ip-address
     |  |  |     |  |     +--ro destination
     inet:ip-address
     |  |  |     |  |     +--ro tunnel-id
     uint16
     |  |  |     |  |     +--ro lsp-id
     uint16
     |  |  |     |  |     +--ro extended-tunnel-id
     inet:ip-address
     |  |  |     |  |     +--ro operational-state?
     identityref
     |  |  |     |  |     +--ro path-setup-protocol?
     identityref
     |  |  |     |  |     +--ro origin-type?
     enumeration
     |  |  |     |  |     +--ro lsp-resource-status?
     enumeration
     |  |  |     |  |     +--ro lockout-of-normal?
     boolean
     |  |  |     |  |     +--ro freeze?
     boolean
     |  |  |     |  |     +--ro lsp-protection-role?
     enumeration
     |  |  |     |  |     +--ro lsp-protection-state?
     identityref
     |  |  |     |  |     +--ro protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     |  |     +--ro lsp-shared-resources-tunnel?
     te:tunnel-ref
     |  |  |     |  |     +--ro lsp-record-route-subobjects
     |  |  |     |  |     |  +--ro record-route-subobject* [index]
     |  |  |     |  |     |     +--ro index               uint32
     |  |  |     |  |     |     +--ro (type)?
     |  |  |     |  |     |        +--:(numbered)
     |  |  |     |  |     |        |  +--ro address?
     te-types:te-tp-id
     |  |  |     |  |     |        |  +--ro ip-flags?     binary
     |  |  |     |  |     |        +--:(unnumbered)
     |  |  |     |  |     |        |  +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |        |  +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |        +--:(label)
     |  |  |     |  |     |           +--ro label-hop
     |  |  |     |  |     |              +--ro te-label
     |  |  |     |  |     |              |  +--ro (technology)?
     |  |  |     |  |     |              |  |  +--:(generic)
     |  |  |     |  |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |              |  +--ro direction?
     te-label-direction
     |  |  |     |  |     |              +--ro label-flags?   binary
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |     |  +--ro path-metric* [metric-type]
     |  |  |     |  |     |  |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |     |  |  +--ro state
     |  |  |     |  |     |  |     +--ro metric-type?
     identityref
     |  |  |     |  |     |  |     +--ro accumulative-value?   uint64
     |  |  |     |  |     |  +--ro path-affinities
     |  |  |     |  |     |  |  +--ro constraints* [usage]
     |  |  |     |  |     |  |     +--ro usage
     identityref
     |  |  |     |  |     |  |     +--ro (style)?
     |  |  |     |  |     |  |        +--:(value)
     |  |  |     |  |     |  |        |  +--ro value?
     te-types:admin-groups
     |  |  |     |  |     |  |        +--:(named)
     |  |  |     |  |     |  |           +--ro affinity-names* [name]
     |  |  |     |  |     |  |              +--ro name    string
     |  |  |     |  |     |  +--ro path-srlgs
     |  |  |     |  |     |  |  +--ro (style)?
     |  |  |     |  |     |  |     +--:(values)
     |  |  |     |  |     |  |     |  +--ro usage?         identityref
     |  |  |     |  |     |  |     |  +--ro values*        te-types:srlg
     |  |  |     |  |     |  |     +--:(named)
     |  |  |     |  |     |  |        +--ro constraints
     |  |  |     |  |     |  |           +--ro constraint* [usage]
     |  |  |     |  |     |  |              +--ro usage
     identityref
     |  |  |     |  |     |  |              +--ro constraint
     |  |  |     |  |     |  |                 +--ro srlg-names* [name]
     |  |  |     |  |     |  |                    +--ro name    string
     |  |  |     |  |     |  +--ro path-route-objects
     |  |  |     |  |     |  |  +--ro path-computed-route-object*
     [index]
     |  |  |     |  |     |  |     +--ro index    -> ../state/index
     |  |  |     |  |     |  |     +--ro state
     |  |  |     |  |     |  |        +--ro index?
     uint32
     |  |  |     |  |     |  |        +--ro (type)?
     |  |  |     |  |     |  |           +--:(num-unnum-hop)
     |  |  |     |  |     |  |           |  +--ro num-unnum-hop
     |  |  |     |  |     |  |           |     +--ro node-id?
     te-types:te-node-id
     |  |  |     |  |     |  |           |     +--ro link-tp-id?
     te-types:te-tp-id
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           |     +--ro direction?
     te-link-direction
     |  |  |     |  |     |  |           +--:(as-number)
     |  |  |     |  |     |  |           |  +--ro as-number-hop
     |  |  |     |  |     |  |           |     +--ro as-number?   binary
     |  |  |     |  |     |  |           |     +--ro hop-type?
     te-hop-type
     |  |  |     |  |     |  |           +--:(label)
     |  |  |     |  |     |  |              +--ro label-hop
     |  |  |     |  |     |  |                 +--ro te-label
     |  |  |     |  |     |  |                    +--ro (technology)?
     |  |  |     |  |     |  |                    |  +--:(generic)
     |  |  |     |  |     |  |                    |     +--ro generic?
     rt-types:generalized-label
     |  |  |     |  |     |  |                    +--ro direction?
     te-label-direction
     |  |  |     |  |     |  +--ro shared-resources-tunnels
     |  |  |     |  |     |     +--ro lsp-shared-resources-tunnel*
     te:tunnel-ref
     |  |  |     |  |     +--ro te-dev:lsp-timers
     |  |  |     |  |     |  +--ro te-dev:life-time?         uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-install?   uint32
     |  |  |     |  |     |  +--ro te-dev:time-to-destroy?   uint32
     |  |  |     |  |     +--ro te-dev:downstream-info
     |  |  |     |  |     |  +--ro te-dev:nhop?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:outgoing-interface?
     if:interface-ref
     |  |  |     |  |     |  +--ro te-dev:neighbor?
     inet:ip-address
     |  |  |     |  |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  |     +--ro te-dev:upstream-info
     |  |  |     |  |        +--ro te-dev:phop?       inet:ip-address
     |  |  |     |  |        +--ro te-dev:neighbor?   inet:ip-address
     |  |  |     |  |        +--ro te-dev:label?
     rt-types:generalized-label
     |  |  |     |  +--ro te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  |     +--rw te-mpls:static-lsp-name?
     mpls-static:static-lsp-ref
     |  |  +---x tunnel-action
     |  |  |  +---w input
     |  |  |  |  +---w action-type?   identityref
     |  |  |  +--ro output
     |  |  |     +--ro action-result?   identityref
     |  |  +---x protection-external-commands
     |  |  |  +---w input
     |  |  |     +---w protection-external-command?        identityref
     |  |  |     +---w protection-group-ingress-node-id?
     te-types:te-node-id
     |  |  |     +---w protection-group-egress-node-id?
     te-types:te-node-id
     |  |  |     +---w path-ref?                           path-ref
     |  |  |     +---w traffic-type?                       enumeration
     |  |  |     +---w extra-traffic-tunnel-ref?           te:tunnel-ref
     |  |  +--rw te-dev:lsp-install-interval?        uint32
     |  |  +--rw te-dev:lsp-cleanup-interval?        uint32
     |  |  +--rw te-dev:lsp-invalidation-interval?   uint32
     |  |  +--rw te-mpls:tunnel-igp-shortcut
     |  |  |  +--rw te-mpls:shortcut-eligible?   boolean
     |  |  |  +--rw te-mpls:metric-type?         identityref
     |  |  |  +--rw te-mpls:metric?              int32
     |  |  |  +--rw te-mpls:routing-afs*         inet:ip-version
     |  |  +--rw te-mpls:forwarding
     |  |  |  +--rw te-mpls:binding-label?   rt-types:mpls-label
     |  |  |  +--rw te-mpls:load-share?      uint32
     |  |  |  +--rw te-mpls:policy-class?    uint8
     |  |  +--rw te-mpls:bandwidth-mpls
     |  |     +--rw te-mpls:specification-type?
     te-mpls-types:te-bandwidth-type
     |  |     +--rw te-mpls:set-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     +--rw te-mpls:class-type?           te-types:te-ds-class
     |  |     +--ro te-mpls:state
     |  |     |  +--ro te-mpls:signaled-bandwidth?
     te-mpls-types:bandwidth-kbps
     |  |     +--rw te-mpls:auto-bandwidth
     |  |        +--rw te-mpls:enabled?            boolean
     |  |        +--rw te-mpls:min-bw?
     te-mpls-types:bandwidth-kbps
     |  |        +--rw te-mpls:max-bw?
     te-mpls-types:bandwidth-kbps
     |  |        +--rw te-mpls:adjust-interval?    uint32
     |  |        +--rw te-mpls:adjust-threshold?   te-types:percentage
     |  |        +--rw te-mpls:overflow
     |  |        |  +--rw te-mpls:enabled?               boolean
     |  |        |  +--rw te-mpls:overflow-threshold?
     te-types:percentage
     |  |        |  +--rw te-mpls:trigger-event-count?   uint16
     |  |        +--rw te-mpls:underflow
     |  |           +--rw te-mpls:enabled?               boolean
     |  |           +--rw te-mpls:underflow-threshold?
     te-types:percentage
     |  |           +--rw te-mpls:trigger-event-count?   uint16
     |  +--rw tunnel-p2mp* [name]
     |     +--rw name           string
     |     +--rw identifier?    uint16
     |     +--rw description?   string
     |     +--ro state
     |        +--ro operational-state?   identityref
     +--ro lsps-state
     |  +--ro lsp* [source destination tunnel-id lsp-id
     extended-tunnel-id]
     |     +--ro source                             inet:ip-address
     |     +--ro destination                        inet:ip-address
     |     +--ro tunnel-id                          uint16
     |     +--ro lsp-id                             uint16
     |     +--ro extended-tunnel-id                 inet:ip-address
     |     +--ro operational-state?                 identityref
     |     +--ro path-setup-protocol?               identityref
     |     +--ro origin-type?                       enumeration
     |     +--ro lsp-resource-status?               enumeration
     |     +--ro lockout-of-normal?                 boolean
     |     +--ro freeze?                            boolean
     |     +--ro lsp-protection-role?               enumeration
     |     +--ro lsp-protection-state?              identityref
     |     +--ro protection-group-ingress-node-id?   te-types:te-node-id
     |     +--ro protection-group-egress-node-id?    te-types:te-node-id
     |     +--ro lsp-record-route-subobjects
     |     |  +--ro record-route-subobject* [index]
     |     |     +--ro index              uint32
     |     |     +--ro (type)?
     |     |        +--:(numbered)
     |     |        |  +--ro address?      te-types:te-tp-id
     |     |        |  +--ro ip-flags?     binary
     |     |        +--:(unnumbered)
     |     |        |  +--ro node-id?      te-types:te-node-id
     |     |        |  +--ro link-tp-id?   te-types:te-tp-id
     |     |        +--:(label)
     |     |           +--ro label-hop
     |     |              +--ro te-label
     |     |              |  +--ro (technology)?
     |     |              |  |  +--:(generic)
     |     |              |  |     +--ro generic?
     rt-types:generalized-label
     |     |              |  +--ro direction?       te-label-direction
     |     |              +--ro label-flags?   binary
     |     +--ro te-dev:lsp-timers
     |     |  +--ro te-dev:life-time?         uint32
     |     |  +--ro te-dev:time-to-install?   uint32
     |     |  +--ro te-dev:time-to-destroy?   uint32
     |     +--ro te-dev:downstream-info
     |     |  +--ro te-dev:nhop?                 inet:ip-address
     |     |  +--ro te-dev:outgoing-interface?   if:interface-ref
     |     |  +--ro te-dev:neighbor?             inet:ip-address
     |     |  +--ro te-dev:label?
     rt-types:generalized-label
     |     +--ro te-dev:upstream-info
     |        +--ro te-dev:phop?       inet:ip-address
     |        +--ro te-dev:neighbor?   inet:ip-address
     |        +--ro te-dev:label?      rt-types:generalized-label
     +--rw te-dev:interfaces
        +--rw te-dev:threshold-type?           enumeration
        +--rw te-dev:delta-percentage?         te-types:percentage
        +--rw te-dev:threshold-specification?   enumeration
        +--rw te-dev:up-thresholds*            te-types:percentage
        +--rw te-dev:down-thresholds*          te-types:percentage
        +--rw te-dev:up-down-thresholds*       te-types:percentage
        +--rw te-dev:interface* [interface]
           +--rw te-dev:interface
           if:interface-ref
           +--rw te-dev:te-metric?
           te-types:te-metric
           +--rw (te-dev:admin-group-type)?
           |  +--:(te-dev:value-admin-groups)
           |  |  +--rw (te-dev:value-admin-group-type)?
           |  |     +--:(te-dev:admin-groups)
           |  |     |  +--rw te-dev:admin-group?
           te-types:admin-group
           |  |     +--:(te-dev:extended-admin-groups)
           {te-types:extended-admin-groups}?
           |  |        +--rw te-dev:extended-admin-group?
           te-types:extended-admin-group
           |  +--:(te-dev:named-admin-groups)
           |     +--rw te-dev:named-admin-groups* [named-admin-group]
           {te-types:extended-admin-groups,te-types:
           named-extended-admin-groups}?
           |        +--rw te-dev:named-admin-group    ->
           ../../../../te:globals/named-admin-groups/named-admin-group/
           name
           +--rw (te-dev:srlg-type)?
           |  +--:(te-dev:value-srlgs)
           |  |  +--rw te-dev:values* [value]
           |  |     +--rw te-dev:value    uint32
           |  +--:(te-dev:named-srlgs)
           |     +--rw te-dev:named-srlgs* [named-srlg]
           {te-types:named-srlg-groups}?
           |        +--rw te-dev:named-srlg    ->
           ../../../../te:globals/named-srlgs/named-srlg/name
           +--rw te-dev:threshold-type?                     enumeration
           +--rw te-dev:delta-percentage?
           te-types:percentage
           +--rw te-dev:threshold-specification?            enumeration
           +--rw te-dev:up-thresholds*
           te-types:percentage
           +--rw te-dev:down-thresholds*
           te-types:percentage
           +--rw te-dev:up-down-thresholds*
           te-types:percentage
           +--rw te-dev:switching-capabilities* [switching-capability]
           |  +--rw te-dev:switching-capability    identityref
           |  +--rw te-dev:encoding?               identityref
           +--ro te-dev:state
              +--ro te-dev:te-advertisements_state
                 +--ro te-dev:flood-interval?           uint32
                 +--ro te-dev:last-flooded-time?        uint32
                 +--ro te-dev:next-flooded-time?        uint32
                 +--ro te-dev:last-flooded-trigger?     enumeration
                 +--ro te-dev:advertized-level-areas* [level-area]
                    +--ro te-dev:level-area    uint32

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

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

  rpcs:
    +---x interfaces-rpc

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

          Figure 6: TE generic model configuration and state tree

4. TE Generic and Helper YANG Modules

```
<CODE BEGINS> file "ietf-te-types@2018-07-01.yang"
module ietf-te-types {

  namespace "urn:ietf:params:xml:ns:yang:ietf-te-types";

  /* Replace with IANA when assigned */
  prefix "te-types";

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

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

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

     WG Chair: Lou Berger
```

```
               <mailto:lberger@labn.net>

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

     Editor:   Tarek Saad
               <mailto:tsaad@cisco.com>

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

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

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

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

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

  description
    "This module contains a collection of generally
    useful TE specific YANG data type defintions.";

  revision "2018-07-01" {
    description "Latest revision of TE types";
    reference "RFC3209";
  }

  /*
   * Identities
   */
  identity association-type {
    description "Base identity for tunnel association";
    reference "RFC6780, RFC4872, RFC4873";
  }
  identity association-type-recovery {
    base association-type;
    description
      "Association Type Recovery used to association LSPs of
       same tunnel for recovery";
    reference "RFC4872";
  }
  identity association-type-resource-sharing {
    base association-type;
```

```
      description
        "Association Type Resource Sharing used to enable resource
         sharing during make-before-break.";
      reference "RFC4873";
    }
    identity association-type-double-sided-bidir {
      base association-type;
      description
        "Association Type Double Sided bidirectional used to associate
         two LSPs of two tunnels that are independently configured on
         either endpoint";
      reference "RFC7551";
    }
    identity association-type-single-sided-bidir {
      base association-type;
      description
        "Association Type Single Sided bidirectional used to associate
         two LSPs of two tunnels, where a tunnel is configured on one
         side/endpoint, and the other tunnel is dynamically created on
         the other endpoint";
      reference "RFC7551";
    }

    identity objective-function-type {
      description "Base objective function type";
      reference "RFC4657";
    }
    identity of-minimize-cost-path {
      base objective-function-type;
      description
          "Mimimuze cost of path objective function";
    }
    identity of-minimize-load-path {
      base objective-function-type;
      description
          "Minimize the load on path(s) objective
           function";
    }
    identity of-maximize-residual-bandwidth {
      base objective-function-type;
      description
          "Maximize the residual bandwidth objective
           function";
    }
    identity of-minimize-agg-bandwidth-consumption {
      base objective-function-type;
      description
          "minimize the aggregate bandwidth consumption
```

```
           objective function";
   }
   identity of-minimize-load-most-loaded-link {
     base objective-function-type;
     description
         "Minimize the load on the most loaded link
          objective function";
   }
   identity of-minimize-cost-path-set {
     base objective-function-type;
     description
         "Minimize the cost on a path set objective
          function";
   }

   identity path-computation-method {
     description
      "base identity for supported path computation
       mechanisms";
   }

   identity path-locally-computed {
     base path-computation-method;
     description
       "indicates a constrained-path LSP in which the
       path is computed by the local LER";
   }

   identity path-externally-queried {
     base path-computation-method;
     description
      "Constrained-path LSP in which the path is
       obtained by querying an external source, such as a PCE server.
       In the case that an LSP is defined to be externally queried, it
       may also have associated explicit definitions (provided
       to the external source to aid computation); and the path that is
       returned by the external source is not required to provide a
       wholly resolved path back to the originating system - that is to
       say, some local computation may also be required";
   }

   identity path-explicitly-defined {
     base path-computation-method;
     description
      "constrained-path LSP in which the path is
       explicitly specified as a collection of strict or/and loose
       hops";
   }
```

```
  /**
   * Typedefs
   */

  typedef te-bandwidth {
    type string {
      pattern
        '0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
      + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?)?)?[pP](\+)?(12[0-7]|'
      + '1[01]\d|0?\d?\d)?)|0[xX][\da-fA-F]{1,8}|\d+'
      + '(,(0[xX](0((\.0?)?[pP](\+)?0?|(\.0?))|'
      + '1(\.([\da-fA-F]{0,5}[02468aAcCeE]?)?)?[pP](\+)?(12[0-7]|'
      + '1[01]\d|0?\d?\d)?)|0[xX][\da-fA-F]{1,8}|\d+))*';
    }
    description
      "This is the generic bandwidth type that is a string containing
       a list of numbers separated by commas, with each of these
       number can be non-negative decimal, hex integer, or hex float:
       (dec | hex | float)[*(','(dec | hex | float))]
       For packet switching type, a float number is used, such as
       0x1p10.
       For OTN switching type, a list of integers can be used, such
       as '0,2,3,1', indicating 2 odu0's and 1 odu3.
       For DWDM, a list of pairs of slot number and width can be
       used, such as '0, 2, 3, 3', indicating a frequency slot 0 with
       slot width 2 and a frequency slot 3 with slot width 3.";
  } // te-bandwidth

  typedef te-ds-class {
    type uint8 {
      range "0..7";
    }
    description
      "The Differentiatied Class-Type of traffic.";
    reference "RFC4124: section-4.3.1";
  }

  typedef te-link-direction {
    type enumeration {
      enum INCOMING {
        description
          "explicit route represents an incoming link on a node";
      }
      enum OUTGOING {
        description
          "explicit route represents an outgoing link on a node";
      }
    }
    description
     "enumerated type for specifying direction of link on a node";
  }

  typedef te-label-direction {
    type enumeration {
      enum FORWARD {
        description
          "Label allocated for the forward LSP direction";
      }
      enum REVERSE {
        description
          "Label allocated for the reverse LSP direction";
      }
    }
    description
     "enumerated type for specifying the forward or reverse
     label";
  }

  typedef te-hop-type {
    type enumeration {
      enum LOOSE {
        description
          "loose hop in an explicit path";
      }
      enum STRICT {
        description
          "strict hop in an explicit path";
      }
    }
    description
     "enumerated type for specifying loose or strict
      paths";
    reference "RFC3209: section-4.3.2";
  }

  identity LSP_METRIC_TYPE {
    description
      "Base identity for types of LSP metric specification";
  }

  identity LSP_METRIC_RELATIVE {
    base LSP_METRIC_TYPE;
    description
      "The metric specified for the LSPs to which this identity refers
      is specified as a relative value to the IGP metric cost to the
      LSP’s tail-end.";
  }

  identity LSP_METRIC_ABSOLUTE {
    base LSP_METRIC_TYPE;
    description
      "The metric specified for the LSPs to which this identity refers
      is specified as an absolute value";
  }

  identity LSP_METRIC_INHERITED {
    base LSP_METRIC_TYPE;
    description
      "The metric for for the LSPs to which this identity refers is
      not specified explicitly - but rather inherited from the IGP
      cost directly";
  }

  identity tunnel-type {
    description
      "Base identity from which specific tunnel types are
      derived.";
  }

  identity tunnel-p2p {
    base tunnel-type;
    description
      "TE point-to-point tunnel type.";
  }

  identity tunnel-p2mp {
    base tunnel-type;
    description
      "TE point-to-multipoint tunnel type.";
    reference "RFC4875";
  }

  identity tunnel-action-type {
    description
      "Base identity from which specific tunnel action types
       are derived.";
  }

  identity tunnel-action-resetup {
    base tunnel-action-type;
    description
      "TE tunnel action resetup. Tears the
      tunnel’s current LSP (if any) and
      attempts to re-establish a new LSP";
  }

  identity tunnel-action-reoptimize {
    base tunnel-action-type;
    description
      "TE tunnel action reoptimize.
       Reoptimizes placement of the tunnel LSP(s)";
  }

  identity tunnel-action-switchpath {
    base tunnel-action-type;
    description
      "TE tunnel action reoptimize
       Switches the tunnel’s LSP to use the specified path";
  }

  identity te-action-result {
    description
      "Base identity from which specific TE action results
       are derived.";
  }

  identity te-action-success {
    base te-action-result;
    description "TE action successul.";
  }

  identity te-action-fail {
    base te-action-result;
    description "TE action failed.";
  }

  identity tunnel-action-inprogress {
    base te-action-result;
    description "TE action inprogress.";
  }

  identity tunnel-admin-state-type {
    description
      "Base identity for TE tunnel admin states";
  }

  identity tunnel-admin-state-up {
    base tunnel-admin-state-type;
    description "Tunnel administratively state up";
  }

  identity tunnel-admin-state-down {
    base tunnel-admin-state-type;
    description "Tunnel administratively state down";
  }

  identity tunnel-state-type {
    description
      "Base identity for TE tunnel states";
  }

  identity tunnel-state-up {
    base tunnel-state-type;
    description "Tunnel state up";
  }

  identity tunnel-state-down {
    base tunnel-state-type;
    description "Tunnel state down";
  }

  identity lsp-state-type {
    description
      "Base identity for TE LSP states";
  }

  identity lsp-path-computing {
    base lsp-state-type;
    description
      "State path compute in progress";
  }

  identity lsp-path-computation-ok {
    base lsp-state-type;
    description
      "State path compute successful";
  }

  identity lsp-path-computatione-failed {
    base lsp-state-type;
    description
      "State path compute failed";
  }

  identity lsp-state-setting-up {
    base lsp-state-type;
    description
      "State setting up";
  }
  identity lsp-state-setup-ok {
    base lsp-state-type;
    description
      "State setup successful";
  }

  identity lsp-state-setup-failed {
    base lsp-state-type;
    description
      "State setup failed";
  }

  identity lsp-state-up {
    base lsp-state-type;
    description "State up";
  }

  identity lsp-state-tearing-down {
    base lsp-state-type;
    description
      "State tearing down";
  }

  identity lsp-state-down {
    base lsp-state-type;
    description "State down";
  }

  identity path-invalidation-action-type {
    description
      "Base identity for TE path invalidation action types";
  }

  identity path-invalidation-action-drop-type {
    base path-invalidation-action-type;
    description
      "TE path invalidation action drop";
  }

  identity path-invalidation-action-drop-tear {
    base path-invalidation-action-type;
    description
      "TE path invalidation action tear";
  }

  identity lsp-restoration-type {
    description
      "Base identity from which LSP restoration types are
```

```
       derived.";
  }

  identity lsp-restoration-restore-any {
    base lsp-restoration-type;
    description
      "Restores when any of the LSPs is affected by a failure";
  }

  identity lsp-restoration-restore-all {
    base lsp-restoration-type;
    description
      "Restores when all the tunnel LSPs are affected by failure";
  }

  identity restoration-scheme-type {
    description
      "Base identity for LSP restoration schemes";
    reference "RFC4872";
  }

  identity restoration-scheme-preconfigured {
    base restoration-scheme-type;
    description
      "Restoration LSP is preconfigured prior to the failure";
  }

  identity restoration-scheme-precomputed {
    base restoration-scheme-type;
    description
      "Restoration LSP is precomputed prior to the failure";
  }

  identity restoration-scheme-presignaled {
    base restoration-scheme-type;
    description
      "Restoration LSP is presignaledd prior to the failure";
  }

  identity lsp-protection-type {
    description
      "Base identity from which LSP protection types are
      derived.";
  }

  identity lsp-protection-unprotected {
    base lsp-protection-type;
    description
```

```
      "LSP protection 'Unprotected'";
    reference "RFC4872";
  }

  identity lsp-protection-reroute-extra {
    base lsp-protection-type;
    description
      "LSP protection '(Full) Rerouting'";
    reference "RFC4872";
  }

  identity lsp-protection-reroute {
    base lsp-protection-type;
    description
      "LSP protection 'Rerouting without Extra-Traffic'";
    reference "RFC4872";
  }

  identity lsp-protection-1-for-n {
    base lsp-protection-type;
    description
      "LSP protection '1:N Protection with Extra-Traffic'";
    reference "RFC4872";
  }

  identity lsp-protection-unidir-1-to-1 {
    base lsp-protection-type;
    description
      "LSP protection '1+1 Unidirectional Protection'";
    reference "RFC4872";
  }

  identity lsp-protection-bidir-1-to-1 {
    base lsp-protection-type;
    description
      "LSP protection '1+1 Bidirectional Protection'";
    reference "RFC4872";
  }

  identity lsp-protection-extra-traffic {
    base lsp-protection-type;
    description
      "LSP protection 'Extra-Traffic'";
    reference
      "ITU-T G.808, RFC 4427.";
  }

  identity lsp-protection-state {
    description
      "Base identity of protection states for reporting
       purposes.";
  }

  identity normal {
    base lsp-protection-state;
    description "Normal state.";
  }

  identity signal-fail-of-protection {
    base lsp-protection-state;
    description
        "There is a SF condition on the protection transport
        entity which has higher priority than the FS command.";
    reference
        "ITU-T G.873.1, G.8031, G.8131";
  }

  identity lockout-of-protection {
    base lsp-protection-state;
    description
        "A Loss of Protection (LoP) command is active.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity forced-switch {
    base lsp-protection-state;
    description
        "A forced switch (FS) command is active.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity signal-fail {
    base lsp-protection-state;
    description
        "There is a SF condition on either the working
        or the protection path.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity signal-degrade {
    base lsp-protection-state;
    description
        "There is an SD condition on either the working or the
```

```
         protection path.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity manual-switch {
    base lsp-protection-state;
    description
        "A manual switch (MS) command is active.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity wait-to-restore {
    base lsp-protection-state;
    description
        "A wait time to restore (WTR) is running.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity do-not-revert {
    base lsp-protection-state;
    description
        "A DNR condition is active because of a non-revertive
         behavior.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity failure-of-protocol {
    base lsp-protection-state;
    description
        "The protection is not working because of a failure of
         protocol condition.";
    reference
        "ITU-T G.873.1, G.8031, G.8131";
  }

  identity protection-external-commands {
    description
      "Protection external commands for trouble shooting
      purposes.";
  }

  identity action-freeze {
    base protection-external-commands;
    description
```

```
      "A temporary configuration action initiated by an operator
       command to prevent any switch action to be taken and as such
       freezes the current state.";
    reference
      "ITU-T G.808, RFC 4427";
  }

  identity clear-freeze {
    base protection-external-commands;
    description
      "An action that clears the active freeze state.";
    reference
      "ITU-T G.808, RFC 4427";
  }

  identity action-lockout-of-normal {
    base protection-external-commands;
    description
      "A temporary configuration action initiated by an operator
       command to ensure that the normal traffic is not allowed
       to use the protection transport entity.";
    reference
      "ITU-T G.808, RFC 4427";
  }

  identity clear-lockout-of-normal {
    base protection-external-commands;
    description
      "An action that clears the active lockout of normal state.";
    reference
      "ITU-T G.808, RFC 4427";
  }

  identity action-lockout-of-protection {
    base protection-external-commands;
    description
      "A temporary configuration action initiated by an operator
       command to ensure that the protection transport entity is
       temporarily not available to transport a traffic signal
       (either normal or extra traffic).";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity action-forced-switch {
    base protection-external-commands;
    description
        "A switch action initiated by an operator command to swith
```

```
         the extra traffic signal, the normal traffic signal, or the
         null signal to the protection transport entity, unless an
         equal or higher priority switch command is in effect.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity action-manual-switch {
    base protection-external-commands;
    description
        "A switch action initiated by an operator command to swith
         the extra traffic signal, the normal traffic signal #i, or
         the null signal to the protection transport entity, unless
         a fault condition exists on other transport entities or an
         equal or higher priority switch command is in effect.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity action-exercise {
    base protection-external-commands;
    description
        "An action to start testing if the APS communication is
         operating correctly. It is lower priority than any other
         state or command.";
    reference
        "ITU-T G.808, RFC 4427";
  }

  identity clear {
    base protection-external-commands;
    description
        "An action that clears the active near-end lockout of
         protection, forced switch, manual switch, WTR state,
         or exercise command.";
    reference
        "ITU-T G.808, RFC 4427";
  }



  identity switching-capabilities {
    description
      "Base identity for interface switching capabilities";
    reference "RFC3471";
  }

  identity switching-psc1 {
    base switching-capabilities;
    description
      "Packet-Switch Capable-1 (PSC-1)";
    reference "RFC3471";
  }

  identity switching-evpl {
    base switching-capabilities;
    description
      "Ethernet Virtual Private Line (EVPL)";
  }

  identity switching-l2sc {
    base switching-capabilities;
    description
      "Layer-2 Switch Capable (L2SC)";
    reference "RFC3471";
  }

  identity switching-tdm {
    base switching-capabilities;
    description
      "Time-Division-Multiplex Capable (TDM)";
    reference "RFC3471";
  }

  identity switching-otn {
    base switching-capabilities;
    description
      "OTN-TDM capable";
  }

  identity switching-dcsc {
    base switching-capabilities;
    description
      "Data Channel Switching Capable (DCSC)";
  }

  identity switching-lsc {
    base switching-capabilities;
    description
      "Lambda-Switch Capable (LSC)";
    reference "RFC3471";
  }

  identity switching-fsc {
    base switching-capabilities;
    description
```

```
        "Fiber-Switch Capable (FSC)";
      reference "RFC3471";
    }

    identity lsp-encoding-types {
      description
        "Base identity for encoding types";
      reference "RFC3471";
    }

    identity lsp-encoding-packet {
      base lsp-encoding-types;
      description
        "Packet LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-ethernet {
      base lsp-encoding-types;
      description
        "Ethernet LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-pdh {
      base lsp-encoding-types;
      description
        "ANSI/ETSI LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-sdh {
      base lsp-encoding-types;
      description
        "SDH ITU-T G.707 / SONET ANSI T1.105 LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-digital-wrapper {
      base lsp-encoding-types;
      description
        "Digital Wrapper LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-lambda {
      base lsp-encoding-types;
      description
```

```
        "Lambda (photonic) LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-fiber {
      base lsp-encoding-types;
      description
        "Fiber LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-fiber-channel {
      base lsp-encoding-types;
      description
        "FiberChannel LSP encoding";
      reference "RFC3471";
    }

    identity lsp-encoding-oduk {
      base lsp-encoding-types;
      description
        "G.709 ODUk (Digital Path)LSP encoding";
    }

    identity lsp-encoding-optical-channel {
      base lsp-encoding-types;
      description
        "Line (e.g., 8B/10B) LSP encoding";
    }

    identity lsp-encoding-line {
      base lsp-encoding-types;
      description
        "Line (e.g., 8B/10B) LSP encoding";
    }

    identity path-signaling-type {
      description
        "base identity from which specific LSPs path
         setup types are derived";
    }

    identity path-setup-static {
      base path-signaling-type;
      description
        "Static LSP provisioning path setup";
    }
```

```
  identity path-setup-rsvp {
    base path-signaling-type;
    description
      "RSVP-TE signaling path setup";
    reference "RFC3209";
  }

  identity path-setup-sr {
    base path-signaling-type;
    description
      "Segment-routing path setup";
  }

  identity path-scope-type {
    description
      "base identity from which specific path
       scope types are derived";
  }

  identity path-scope-segment {
    base path-scope-type;
    description
      "Path scope segment";
  }

  identity path-scope-end-to-end {
    base path-scope-type;
    description
      "Path scope end to end";
  }

  /* TE basic features */
  feature p2mp-te {
    description
      "Indicates support for P2MP-TE";
    reference "RFC4875";
  }

  feature frr-te {
    description
      "Indicates support for TE FastReroute (FRR)";
    reference "RFC4090";
  }

  feature extended-admin-groups {
    description
      "Indicates support for TE link extended admin
      groups.";
    reference "RFC7308";
  }

  feature named-path-affinities {
    description
      "Indicates support for named path affinities";
  }

  feature named-extended-admin-groups {
    description
      "Indicates support for named extended admin groups";
  }

  feature named-srlg-groups {
    description
      "Indicates support for named SRLG groups";
  }

  feature named-path-constraints {
    description
      "Indicates support for named path constraints";
  }

  feature path-optimization-metric {
    description
      "Indicates support for path optimization metric";
  }

  feature path-optimization-objective-function {
    description
      "Indicates support for path optimization objective function";
  }

  identity route-usage-type {
    description
      "Base identity for route usage";
  }

  identity route-include-ero {
    base route-usage-type;
    description
      "Include ERO from route";
  }

  identity route-exclude-ero {
    base route-usage-type;
    description
      "Exclude ERO from route";
  }

  identity route-exclude-srlg {
    base route-usage-type;
    description
      "Exclude SRLG from route";
  }

  identity path-metric-type {
    description
      "Base identity for path metric type";
  }

  identity path-metric-te {
    base path-metric-type;
    description
      "TE path metric";
    reference "RFC3785";
  }

  identity path-metric-igp {
    base path-metric-type;
    description
      "IGP path metric";
    reference "RFC3785";
  }

  identity path-metric-hop {
    base path-metric-type;
    description
      "Hop path metric";
  }

  identity path-metric-delay-average {
    base path-metric-type;
    description
      "Unidirectional average link delay";
    reference "RFC7471";
  }

  identity path-metric-residual-bandwidth {
    base path-metric-type;
    description
      "Unidirectional Residual Bandwidth, which is defined to be
       Maximum Bandwidth [RFC3630] minus the bandwidth currently
       allocated to  LSPs.";
    reference "RFC7471";
  }
  identity path-metric-optimize-includes {
    base path-metric-type;
    description
      "A metric that optimizes the number of included resources
       specified in a set";
  }

  identity path-metric-optimize-excludes {
    base path-metric-type;
    description
      "A metric that optimizes the number of excluded resources
       specified in a set";
  }

  identity path-tiebreaker-type {
    description
      "Base identity for path tie-breaker type";
  }

  identity path-tiebreaker-minfill {
    base path-tiebreaker-type;
    description
      "Min-Fill LSP path placement";
  }

  identity path-tiebreaker-maxfill {
    base path-tiebreaker-type;
    description
      "Max-Fill LSP path placement";
  }

  identity path-tiebreaker-randoom {
    base path-tiebreaker-type;
    description
      "Random LSP path placement";
  }

  identity bidir-provisioning-mode {
    description
      "Base identity for bidirectional provisioning
      mode.";
    reference "RFC7551";
  }

  identity bidir-provisioning-single-sided {
    base bidir-provisioning-mode;
    description
      "Single-sided bidirectional provioning mode";
    reference "RFC7551";
  }

  identity bidir-provisioning-double-sided {
    base bidir-provisioning-mode;
    description
      "Double-sided bidirectional provioning mode";
    reference "RFC7551";
  }

  identity bidir-association-type {
    description
      "Base identity for bidirectional association type";
    reference "RFC7551";
  }

  identity bidir-assoc-corouted {
    base bidir-association-type;
    description
      "Co-routed bidirectional association type";
    reference "RFC7551";
  }

  identity bidir-assoc-non-corouted {
    base bidir-association-type;
    description
      "Non co-routed bidirectional association type";
    reference "RFC7551";
  }

  identity resource-affinities-type {
    description
      "Base identity for resource affinities";
    reference "RFC2702";
  }

  identity resource-aff-include-all {
    base resource-affinities-type;
    description
      "The set of attribute filters associated with a
      tunnel all of which must be present for a link
      to be acceptable";
    reference "RFC2702 and RFC3209";
  }

  identity resource-aff-include-any {
    base resource-affinities-type;
    description
```

```
        "The set of attribute filters associated with a
        tunnel any of which must be present for a link
        to be acceptable";
      reference "RFC2702 and RFC3209";
    }

    identity resource-aff-exclude-any {
      base resource-affinities-type;
      description
        "The set of attribute filters associated with a
        tunnel any of which renders a link unacceptable";
      reference "RFC2702 and RFC3209";
    }

    typedef optimization-goal {
      type enumeration {
        enum minimize {
          description "Pick lowest path metric goal";
        }
        enum maximize {
          description "Pick highest path metric goal";
        }
        enum randomize {
          description
            "Pick a path at random from list of
             equally favorable ones";
        }
      }
      description "TE optimization goal";
    }

    identity te-optimization-criterion {
      description
        "Base identity for TE optimization criterion.";
      reference
        "RFC3272: Overview and Principles of Internet Traffic
         Engineering.";
    }

    identity not-optimized {
      base te-optimization-criterion;
      description "Optimization is not applied.";
    }

    identity cost {
      base te-optimization-criterion;
      description "Optimized on cost.";
    }
```

```
  identity delay {
    base te-optimization-criterion;
    description "Optimized on delay.";
  }

  /*
   * Typedefs
   */

  typedef percentage {
    type uint8 {
      range "0..100";
    }
    description
      "Integer indicating a percentage value";
  }

  typedef performance-metric-normality {
    type enumeration {
      enum "unknown" {
        value 0;
        description
          "Unknown.";
      }
      enum "normal" {
        value 1;
        description
          "Normal.";
      }
      enum "abnormal" {
        value 2;
        description
          "Abnormal. The anomalous bit is set.";
      }
    }
    description
      "Indicates whether a performance metric is normal, abnormal, or
       unknown.";
    reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  typedef te-admin-status {
    type enumeration {
      enum up {
        description
          "Enabled.";
      }
      enum down {
        description
          "Disabled.";
      }
      enum testing {
        description
          "In some test mode.";
      }
      enum preparing-maintenance {
        description
          "Resource is disabled in the control plane to prepare for
           graceful shutdown for maintenance purposes.";
        reference
          "RFC5817: Graceful Shutdown in MPLS and Generalized MPLS
           Traffic Engineering Networks";
      }
      enum maintenance {
        description
          "Resource is disabled in the data plane for maintenance
           purposes.";
      }
    }
    description
      "Defines a type representing the administrative status of
       a TE resource.";
  }

  typedef te-global-id {
    type uint32;
    description
      "An identifier to uniquely identify an operator, which can be
       either a provider or a client.
       The definition of this type is taken from RFC6370 and RFC5003.
       This attribute type is used solely to provide a globally
       unique context for TE topologies.";
  }

  typedef te-link-access-type {
    type enumeration {
      enum point-to-point {
        description
          "The link is point-to-point.";
      }
      enum multi-access {
        description
          "The link is multi-access, including broacast and NBMA.";
      }
    }
    description
      "Defines a type representing the access type of a TE link.";
    reference
      "RFC3630: Traffic Engineering (TE) Extensions to OSPF
       Version 2.";
  }

  typedef te-node-id {
    type yang:dotted-quad;
    description
      "An identifier for a node in a topology.
       The identifier is represented as 32-bit unsigned integer in
       the dotted-quad notation.
       This attribute is mapped to Router ID in
       RFC3630, RFC5329, RFC5305, and RFC6119.";
  }

  typedef te-oper-status {
    type enumeration {
      enum up {
        description
        "Operational up.";
      }
      enum down {
        description
        "Operational down.";
      }
      enum testing {
        description
        "In some test mode.";
      }
      enum unknown {
        description
        "Status cannot be determined for some reason.";
      }
      enum preparing-maintenance {
        description
          "Resource is disabled in the control plane to prepare for
           graceful shutdown for maintenance purposes.";
        reference
          "RFC5817: Graceful Shutdown in MPLS and Generalized MPLS
           Traffic Engineering Networks";
      }
      enum maintenance {
        description
          "Resource is disabled in the data plane for maintenance
           purposes.";
      }
    }
    description
      "Defines a type representing the operational status of
       a TE resource.";
  }

  typedef te-path-disjointness {
    type bits {
      bit node {
        position 0;
        description "Node disjoint.";
      }
      bit link {
        position 1;
        description "Link disjoint.";
      }
      bit srlg {
        position 2;
        description "SRLG (Shared Risk Link Group) disjoint.";
      }
    }
    description
      "Type of the resource disjointness for a TE tunnel path.";
    reference
      "RFC4872: RSVP-TE Extensions in Support of End-to-End
       Generalized Multi-Protocol Label Switching (GMPLS)
       Recovery";
  } // te-path-disjointness

  typedef te-recovery-status {
    type enumeration {
      enum normal {
        description
          "Both the recovery and working spans are fully
           allocated and active, data traffic is being
           transported over (or selected from) the working
           span, and no trigger events are reported.";
      }
      enum recovery-started {
        description
          "The recovery action has been started, but not completed.";
      }
      enum recovery-succeeded {
        description
```

```
          "The recovery action has succeeded. The working span has
           reported a failure/degrade condition and the user traffic
           is being transported (or selected) on the recovery span.";
      }
      enum recovery-failed {
        description
          "The recovery action has failed.";
      }
      enum reversion-started {
        description
          "The reversion has started.";
      }
      enum reversion-failed {
        description
          "The reversion has failed.";
      }
      enum recovery-unavailable {
        description
          "The recovery is unavailable -- either as a result of an
           operator Lockout command or a failure condition detected
           on the recovery span.";
      }
      enum recovery-admin {
        description
          "The operator has issued a command switching the user
           traffic to the recovery span.";
      }
      enum wait-to-restore {
        description
          "The recovery domain is recovering from a failuer/degrade
           condition on the working span that is being controlled by
           the Wait-to-Restore (WTR) timer.";
      }
    }
    description
      "Defines the status of a recovery action.";
    reference
      "RFC4427: Recovery (Protection and Restoration) Terminology
       for Generalized Multi-Protocol Label Switching (GMPLS).
       RFC6378: MPLS Transport Profile (MPLS-TP) Linear Protection";
  }

  typedef te-template-name {
    type string {
      pattern '/?([a-zA-Z0-9\-_.]+)(/[a-zA-Z0-9\-_.]+)*';
    }
    description
      "A type for the name of a TE node template or TE link
```

```
         template.";
    }

    typedef te-topology-event-type {
      type enumeration {
        enum "add" {
          value 0;
          description
            "A TE node or te-link has been added.";
        }
        enum "remove" {
          value 1;
          description
            "A TE node or te-link has been removed.";
        }
        enum "update" {
          value 2;
          description
            "A TE node or te-link has been updated.";
        }
      }
      description "TE  Event type for notifications";
    } // te-topology-event-type

    typedef te-topology-id {
      type string {
        pattern
          '([a-zA-Z0-9\-_.]+:)*'
        + '/?([a-zA-Z0-9\-_.]+)(/[a-zA-Z0-9\-_.]+)*';
      }
      description
        "An identifier for a topology.
         It is optional to have one or more prefixes at the begining,
         separated by colons. The prefixes can be the network-types,
         defined in ietf-network.yang, to help user to understand the
         topology better before further inquiry.";
    }

    typedef te-tp-id {
      type union {
        type uint32;          // Unnumbered
        type inet:ip-address; // IPv4 or IPv6 address
      }
      description
        "An identifier for a TE link endpoint on a node.
         This attribute is mapped to local or remote link identifier in
         RFC3630 and RFC5305.";
    }
```

```
  typedef admin-group {
    type binary {
      length 4;
    }
    description
      "Administrative group/Resource class/Color.";
    reference "RFC3630 and RFC5305";
  }

  typedef extended-admin-group {
    type binary;
    description
      "Extended administrative group/Resource class/Color.";
    reference "RFC7308";
  }

  typedef admin-groups {
    type union {
      type admin-group;
      type extended-admin-group;
    }
    description "TE administrative group derived type";
  }

  typedef srlg {
    type uint32;
    description "SRLG type";
    reference "RFC4203 and RFC5307";
  }

  identity path-computation-srlg-type {
    description
      "Base identity for SRLG path computation";
  }

  identity srlg-ignore {
    base path-computation-srlg-type;
    description
      "Ignores SRLGs in path computation";
  }

  identity srlg-strict {
    base path-computation-srlg-type;
    description
      "Include strict SRLG check in path computation";
  }

  identity srlg-preferred {
    base path-computation-srlg-type;
    description
      "Include preferred SRLG check in path computation";
  }

  identity srlg-weighted {
    base path-computation-srlg-type;
    description
      "Include weighted SRLG check in path computation";
  }

  typedef te-metric {
    type uint32;
    description
      "TE link metric";
    reference "RFC3785";
  }

  /**
   * TE bandwidth groupings
   **/
  identity otn-rate-type {
    description
      "Base type to identify OTN bit rates of various information
       structures.";
    reference "RFC7139";
  }
  identity odu0 {
    base otn-rate-type;
    description
        "ODU0 bit rate.";
  }
  identity odu1 {
    base otn-rate-type;
    description
        "ODU1 bit rate.";
  }
  identity odu2 {
    base otn-rate-type;
    description
        "ODU2 bit rate.";
  }
  identity odu3 {
    base otn-rate-type;
    description
        "ODU3 bit rate.";
  }
  identity odu4 {
    base otn-rate-type;
    description
        "ODU4 bit rate.";
  }
  identity odu2e {
    base otn-rate-type;
    description
        "ODU2e bit rate.";
  }
  identity oduc {
    base otn-rate-type;
    description
        "ODUCn bit rate.";
  }
  identity oduflex {
    base otn-rate-type;
    description
        "ODUflex bit rate.";
  }

  identity wdm-spectrum-type {
    description
      "Base type to identify WDM spectrum type.";
  }
  identity cwdm {
    base wdm-spectrum-type;
    description "CWDM.";
    reference "RFC6205";
  }
  identity dwdm {
    base wdm-spectrum-type;
    description "DWDM.";
    reference "RFC6205";
  }
  identity flexible-grid {
    base wdm-spectrum-type;
    description "Flexible grid.";
    reference "RFC6205";
  }

  grouping te-bandwidth {
    description
      "This grouping defines the generic TE bandwidth.
       For some known data plane technologies, specific modeling
       structures are specified. The string encoded te-bandwidth
       type is used for un-specified technologies.
       The modeling structure can be augmented later for other
       technologies.";
    container te-bandwidth {
      description
        "Container that specifies TE bandwidth.";
      choice technology {
        default generic;
        description
          "Data plane technology type.";
        case generic {
          leaf generic {
            type te-bandwidth;
            description
              "Bandwidth specified in a generic format.";
          }
        }
      }
    }
  }

  /**
   * TE label groupings
   **/
  grouping te-label {
    description
      "This grouping defines the generic TE label.
       The modeling structure can be augmented for each technology.
       For un-specified technologies, rt-types:generalized-label
       is used.";
    container te-label {
      description
        "Container that specifies TE label.";
      choice technology {
        default generic;
        description
          "Data plane technology type.";
        case generic {
          leaf generic {
            type rt-types:generalized-label;
            description
              "TE label specified in a generic format.";
          }
        }
      }
      leaf direction {
        type te-label-direction;
        description "Label direction";
      }
    }
  }
  /**
   * TE performance metric groupings
   **/
  grouping performance-metric-container {
    description
      "A container containing performance metric attributes.";
    container performance-metric {
      description
        "Link performance information in real time.";
      reference
        "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
         RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
         RFC7823: Performance-Based Path Selection for Explicitly
         Routed Label Switched Paths (LSPs) Using TE Metric
         Extensions";
      container measurement {
        description
          "Measured performance metric values. Static configuration
           and manual overrides of these measurements are also
           allowed.";
        uses performance-metric-attributes;
      }
      container normality
      {
        description
          "Performance metric normality values.";
        uses performance-metric-normality-attributes;
      }
      uses performance-metric-throttle-container;
    }
  } // performance-metric-container

  grouping te-topology-identifier {
    description
      "Augmentation for TE topology.";
    container te-topology-identifier {
      description "TE topology identifier container";
      leaf provider-id {
        type te-types:te-global-id;
        description
          "An identifier to uniquely identify a provider.";
      }
      leaf client-id {
        type te-types:te-global-id;
        description
          "An identifier to uniquely identify a client.";
      }
      leaf topology-id {
        type te-types:te-topology-id;
        description
          "It is presumed that a datastore will contain many
           topologies. To distinguish between topologies it is
           vital to have UNIQUE topology identifiers.";
      }
    }
  }



  grouping performance-metric-attributes {
    description
      "Link performance information in real time.";
    reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
    leaf unidirectional-delay {
      type uint32 {
        range 0..16777215;
      }
      description "Delay or latency in micro seconds.";
    }
    leaf unidirectional-min-delay {
      type uint32 {
        range 0..16777215;
      }
      description "Minimum delay or latency in micro seconds.";
    }
    leaf unidirectional-max-delay {
      type uint32 {
        range 0..16777215;
      }
      description "Maximum delay or latency in micro seconds.";
    }
    leaf unidirectional-delay-variation {
      type uint32 {
        range 0..16777215;
      }
      description "Delay variation in micro seconds.";
    }
    leaf unidirectional-packet-loss {
      type decimal64 {
        fraction-digits 6;
        range "0 .. 50.331642";
      }
      description
        "Packet loss as a percentage of the total traffic sent
         over a configurable interval. The finest precision is
         0.000003%.";
    }
    leaf unidirectional-residual-bandwidth {
      type rt-types:bandwidth-ieee-float32;
      description
        "Residual bandwidth that subtracts tunnel
         reservations from Maximum Bandwidth (or link capacity)
         [RFC3630] and provides an aggregated remainder across QoS
         classes.";
    }
    leaf unidirectional-available-bandwidth {
      type rt-types:bandwidth-ieee-float32;
      description
        "Available bandwidth that is defined to be residual
         bandwidth minus the measured bandwidth used for the
         actual forwarding of non-RSVP-TE LSP packets.  For a
         bundled link, available bandwidth is defined to be the
         sum of the component link available bandwidths.";
    }
    leaf unidirectional-utilized-bandwidth {
      type rt-types:bandwidth-ieee-float32;
      description
        "Bandwidth utilization that represents the actual
         utilization of the link (i.e. as measured in the router).
         For a bundled link, bandwidth utilization is defined to
         be the sum of the component link bandwidth
         utilizations.";
    }
  } // performance-metric-attributes

  grouping performance-metric-normality-attributes {
    description
      "Link performance metric normality attributes.";
    reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
    leaf unidirectional-delay {
      type te-types:performance-metric-normality;
      description "Delay normality.";
    }
    leaf unidirectional-min-delay {
      type te-types:performance-metric-normality;
      description "Minimum delay or latency normality.";
    }
    leaf unidirectional-max-delay {
      type te-types:performance-metric-normality;
      description "Maximum delay or latency normality.";
    }
    leaf unidirectional-delay-variation {
      type te-types:performance-metric-normality;
      description "Delay variation normality.";
    }
    leaf unidirectional-packet-loss {
      type te-types:performance-metric-normality;
      description "Packet loss normality.";
    }
    leaf unidirectional-residual-bandwidth {
      type te-types:performance-metric-normality;
      description "Residual bandwidth normality.";
    }
    leaf unidirectional-available-bandwidth {
      type te-types:performance-metric-normality;
      description "Available bandwidth normality.";
    }
    leaf unidirectional-utilized-bandwidth {
      type te-types:performance-metric-normality;
      description "Bandwidth utilization normality.";
    }
  } // performance-metric-normality-attributes

  grouping performance-metric-throttle-container {
    description
      "A container controlling performance metric throttle.";
    container throttle {
      must "suppression-interval >= measure-interval" {
        error-message
          "suppression-interval cannot be less then
           measure-interval.";
        description
          "Constraint on suppression-interval and
           measure-interval.";
      }
      description
        "Link performance information in real time.";
      reference
        "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
         RFC7810: IS-IS Traffic Engineering (TE) Metric Extensions.
         RFC7823: Performance-Based Path Selection for Explicitly
         Routed Label Switched Paths (LSPs) Using TE Metric
          Extensions";
      leaf unidirectional-delay-offset {
        type uint32 {
          range 0..16777215;
        }
        description
          "Offset value to be added to the measured delay value.";
      }
      leaf measure-interval {
        type uint32;
        default 30;
        description
          "Interval in seconds to measure the extended metric
           values.";
      }
      leaf advertisement-interval {
        type uint32;
        description
          "Interval in seconds to advertise the extended metric
           values.";
      }
      leaf suppression-interval {
        type uint32 {
          range "1 .. max";
        }
        default 120;
        description
          "Interval in seconds to suppress advertising the extended
           metric values.";
      }
      container threshold-out {
        uses performance-metric-attributes;
        description
          "If the measured parameter falls outside an upper bound
           for all but the min delay metric (or lower bound for
           min-delay metric only) and the advertised value is not
           already outside that bound, anomalous announcement will be
           triggered.";
      }
      container threshold-in {
        uses performance-metric-attributes;
        description
          "If the measured parameter falls inside an upper bound
           for all but the min delay metric (or lower bound for
           min-delay metric only) and the advertised value is not
           already inside that bound, normal (anomalous-flag cleared)
           announcement will be triggered.";
      }
      container threshold-accelerated-advertisement {
        description
          "When the difference between the last advertised value and
           current measured value exceed this threshold, anomalous
           announcement will be triggered.";
        uses performance-metric-attributes;
      }
    }
  } // performance-metric-throttle-container

  /**
   * TE tunnel generic groupings
   **/

  /* Tunnel path selection parameters */
  grouping explicit-route-hop {
    description
      "The explicit route subobject grouping";
    leaf index {
      type uint32;
      description "ERO subobject index";
    }
    choice type {
      description
        "The explicit route subobject type";
      case num-unnum-hop {
        container num-unnum-hop {
          leaf node-id {
            type te-types:te-node-id;
            description
              "The identifier of a node in the TE topology.";
          }
          leaf link-tp-id {
            type te-types:te-tp-id;
              description
                "TE link termination point identifier. The combination
                 of TE link ID and the TE node ID is used to identify an
                 unnumbered TE link.";
          }
          leaf hop-type {
            type te-hop-type;
            description "strict or loose hop";
          }
          leaf direction {
            type te-link-direction;
            description "Unnumbered Link ERO direction";
          }
          description
```

```
            "Numbered and Unnumbered link/node explicit route
             subobject";
          reference
            "RFC3209: section 4.3 for EXPLICIT_ROUTE in RSVP-TE
             RFC3477: Signalling Unnumbered Links in RSVP-TE";
        }
      }
      case as-number {
        container as-number-hop {
          leaf as-number {
            type binary {
              length 16;
            }
            description "AS number";
          }
          leaf hop-type {
            type te-hop-type;
              description
                "strict or loose hop";
          }
          description
            "Autonomous System explicit route subobject";
        }
      }
      case label {
        container label-hop {
          description "Label hop type";
          uses te-label;
        }
        description
          "The Label ERO subobject";
      }
    }
  }

  grouping record-route-subobject_state {
    description
      "The record route subobject grouping";
    leaf index {
      type uint32;
      description "RRO subobject index";
    }
    choice type {
      description
        "The record route subobject type";
      case numbered {
        leaf address {
          type te-types:te-tp-id;
          description
            "Numbered link TE termination point address.";
        }
        leaf ip-flags {
          type binary {
            length 8;
          }
          description
            "RRO IP address sub-object flags";
          reference "RFC3209";
        }
      }
      case unnumbered {
        leaf node-id {
          type te-types:te-node-id;
          description
            "The identifier of a node in the TE topology.";
        }
        leaf link-tp-id {
          type te-types:te-tp-id;
            description
              "TE link termination point identifier, used
               together with te-node-id to identify the
               link termination point";
        }
        description
          "Unnumbered link record route subobject";
        reference
          "RFC3477: Signalling Unnumbered Links in
           RSVP-TE";
      }
      case label {
        container label-hop {
          description "Label hop type";
          uses te-label;
          leaf label-flags {
            type binary {
              length 8;
            }
            description
              "Label sub-object flags";
            reference "RFC3209";
          }
        }
        description
          "The Label RRO subobject";
      }
    }
```

```
  }

  grouping label-restriction-info {
    description "Label set item info";
    leaf restriction {
      type enumeration {
        enum inclusive {
          description "The label or label range is inclusive.";
        }
        enum exclusive {
          description "The label or label range is exclusive.";
        }
      }
      description
        "Whether the list item is inclusive or exclusive.";
    }
    leaf index {
      type uint32;
      description
        "Then index of the label restriction list entry.";
    }
    container label-start {
      description
        "This is the starting label if a label range is specified.
         This is the label value if a single label is specified,
         in which case, attribute 'label-end' is not set.";
      uses te-label;
    }
    container label-end {
      description
        "The ending label if a label range is specified;
         This attribute is not set, If a single label is
         specified.";
      uses te-label;
    }
    leaf range-bitmap {
      type binary;
      description
        "When there are gaps between label-start and label-end,
         this attribute is used to specified the possitions
         of the used labels.";
    }
  }

  grouping label-set-info {
    description
      "Grouping for List of label restrictions specifying what labels
       may or may not be used on a link connectivity.";
    container label-restrictions {
      description
        "The label restrictions container";
      list label-restriction {
        key "index";
        description
          "The absence of label-set implies that all labels are
           acceptable; otherwise only restricted labels are
           available.";
        reference
          "RFC7579: General Network Element Constraint Encoding
           for GMPLS-Controlled Networks";
        uses label-restriction-info;
      }
    }
  }

  /*** End of TE tunnel groupings ***/
  grouping optimizations_config {
    description "Optimization metrics configuration grouping";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description "TE path metric type";
    }
    leaf weight {
      type uint8;
      description "TE path metric normalization weight";
    }
    container explicit-route-exclude-objects {
      when "../metric-type = " +
           "'te-types:path-metric-optimize-excludes'";
      description
        "Container for the exclude route object list";
      uses path-route-exclude-objects;
    }
    container explicit-route-include-objects {
      when "../metric-type = " +
           "'te-types:path-metric-optimize-includes'";
      description
        "Container for the include route object list";
      uses path-route-include-objects;
    }
  }

  grouping common-constraints_config {
    description
```

```
      "Common constraints grouping that can be set on
       a constraint set or directly on the tunnel";

    uses te-types:te-bandwidth {
      description
        "A requested bandwidth to use for path computation";
    }

    leaf setup-priority {
      type uint8 {
        range "0..7";
      }
      description
        "TE LSP requested setup priority";
      reference "RFC3209";
    }
    leaf hold-priority {
      type uint8 {
        range "0..7";
      }
      description
        "TE LSP requested hold priority";
      reference "RFC3209";
    }
    leaf signaling-type {
      type identityref {
        base te-types:path-signaling-type;
      }
      description "TE tunnel path signaling type";
    }
  }

  grouping tunnel-constraints_config {
    description
      "Tunnel constraints grouping that can be set on
       a constraint set or directly on the tunnel";
    uses te-types:te-topology-identifier;
    uses te-types:common-constraints_config;
  }

  grouping path-metrics-bounds_config {
    description "TE path metric bounds grouping";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description "TE path metric type";
    }
    leaf upper-bound {
      type uint64;
      description "Upper bound on end-to-end TE path metric";
    }
  }

  grouping path-objective-function_config {
    description "Optimization metrics configuration grouping";
    leaf objective-function-type {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Objective function entry";
    }
  }

  /**
   * TE interface generic groupings
   **/
  grouping path-route-objects {
    description
      "List of EROs to be included or excluded when performing
       the path computation.";
    container explicit-route-objects {
      description
        "Container for the exclude route object list";
      list route-object-exclude-always {
        key index;
        description
          "List of explicit route objects to always exclude
           from path computation";
        uses te-types:explicit-route-hop;
      }
      list route-object-include-exclude {
        key index;
        description
          "List of explicit route objects to include or
           exclude in path computation";
        leaf explicit-route-usage {
          type identityref {
            base te-types:route-usage-type;
          }
          description "Explicit-route usage.";
        }
        uses te-types:explicit-route-hop {
          augment "type" {
            case srlg {
              container srlg {
                description "SRLG container";
                leaf srlg {
                  type uint32;
                  description "SRLG value";
                }
              }
              description "An SRLG value to be included or excluded";
            }
            description
              "Augmentation to generic explicit route for SRLG
               exclusion";
          }
        }
      }
    }
  }

  grouping path-route-include-objects {
    description
      "List of EROs to be included when performing
       the path computation.";
    list route-object-include-object {
      key index;
      description
        "List of explicit route objects to be included
         in path computation";
      uses te-types:explicit-route-hop;
    }
  }

  grouping path-route-exclude-objects {
    description
      "List of EROs to be included when performing
       the path computation.";
    list route-object-exclude-object {
      key index;
      description
        "List of explicit route objects to be excluded
         in path computation";
      uses te-types:explicit-route-hop {
        augment "type" {
          case srlg {
            container srlg {
              description "SRLG container";
              leaf srlg {
                type uint32;
                description "SRLG value";
              }
            }
            description "An SRLG value to be included or excluded";
          }
          description
            "Augmentation to generic explicit route for SRLG exclusion";
        }
      }
    }
  }

  grouping generic-path-metric-bounds {
    description "TE path metric bounds grouping";
    container path-metric-bounds {
      description "TE path metric bounds container";
      list path-metric-bound {
        key metric-type;
        description "List of TE path metric bounds";
        uses path-metrics-bounds_config;
      }
    }
  }

  grouping generic-path-optimization {
    description "TE generic path optimization grouping";

    container optimizations {
      description
        "The objective function container that includes
         attributes to impose when computing a TE path";

      choice algorithm {
        description "Optimizations algorithm.";
        case metric {
          if-feature path-optimization-metric;
          /* Optimize by metric */
          list optimization-metric {
            key "metric-type";
            description "TE path metric type";
            uses optimizations_config;
          }
          /* Tiebreakers */
          container tiebreakers {
            description
              "The list of tiebreaker criterion to apply
               on an equally favored set of paths to pick best";
            list tiebreaker {
              key "tiebreaker-type";
              description
                "The list of tiebreaker criterion to apply
                 on an equally favored set of paths to pick best";
              leaf tiebreaker-type {
                type identityref {
                  base te-types:path-metric-type;
                }
                description "The objective function";
              }
            }
          }
        }
        case objective-function {
          if-feature path-optimization-objective-function;
          /* Objective functions */
          container objective-function {
            description
              "The objective function container that includes
               attributes to impose when computing a TE path";
            uses path-objective-function_config;
          }
        }
      }
    }
  }

  grouping generic-path-affinities {
    description
      "Path affinities grouping";
    container path-affinities {
      description
        "Path affinities container";
      list constraint {
        key "usage";
        description
          "List of named affinity constraints";
        leaf usage {
          type identityref {
            base resource-affinities-type;
          }
          description "Affinities usage";
        }
        leaf value {
          type admin-groups;
          description "Affinity value";
        }
      }
    }
  }

  grouping generic-path-srlgs {
    description
      "Path SRLG grouping";
    container path-srlgs {
      description
        "Path SRLG properties container";
      leaf usage {
        type identityref {
          base te-types:route-exclude-srlg;
        }
        description "SRLG usage";
      }
      leaf-list values {
        type srlg;
        description "SRLG value";
      }
    }
  }

  grouping generic-path-disjointness {
    description "Path disjointness grouping";
    leaf disjointness {
      type te-types:te-path-disjointness;
      description
        "The type of resource disjointness.
         Under primary path, disjointness level applies to
         all secondary LSPs. Under secondary, disjointness
         level overrides the one under primary";
    }
  }

  grouping common-path-constraints-attributes {
    description
      "Common path constraints configuration grouping";
    uses common-constraints_config;
    uses generic-path-metric-bounds;
    uses generic-path-affinities;
    uses generic-path-srlgs;
  }

  grouping generic-path-constraints {
    description
      "Global named path constraints configuration
      grouping";
    container path-constraints {
      description "TE named path constraints container";
      uses common-path-constraints-attributes;
      uses generic-path-disjointness;
    }
  }

  grouping generic-path-properties {
    description "TE generic path properties grouping";
    container path-properties {
      config false;
      description "The TE path properties";
      list path-metric {
        key metric-type;
        description "TE path metric type";
        leaf metric-type {
          type identityref {
            base te-types:path-metric-type;
          }
          description "TE path metric type";
        }
        leaf accumulative-value {
          type uint64;
          description "TE path metric accumulative value";
        }
      }
      uses generic-path-affinities;
      uses generic-path-srlgs;
      container path-route-objects {
        description
          "Container for the list of route objects either returned by
           the computation engine or actually used by an LSP";
        list path-route-object {
          key index;
          description
            "List of route objects either returned by the computation
             engine or actually used by an LSP";
          uses explicit-route-hop;
        }
      }
    }
  }
}
<CODE ENDS>
```

Figure 7: TE basic types YANG module

```
<CODE BEGINS> file "ietf-te@2018-07-01.yang"
module ietf-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te";

  /* Replace with IANA when assigned */
  prefix "te";

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

  import ietf-inet-types {
    prefix inet;
  }

  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>

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

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

     Editor:   Tarek Saad
               <mailto:tsaad@cisco.com>

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

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

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

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

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

  description
    "YANG data module for TE configuration,
    state, RPC and notifications.";

  revision "2018-07-01" {
    description "Latest update to TE generic YANG module.";
    reference "TBA";
  }

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

  typedef path-ref {
    type union {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-primary-paths/te:p2p-primary-path/te:name";
      }
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/" +
             "te:p2p-secondary-paths/te:p2p-secondary-path/te:name";
      }
    }
    description
      "This type is used by data models that need to reference
       configured primary or secondary path of a TE tunnel.";
  }

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

  /**
   * TE tunnel generic groupings
   */
  grouping path-affinities-contents_config {
    description
      "Path affinities constraints grouping";
    reference "RFC3630 and RFC5305";
    leaf usage {
      type identityref {
        base te-types:resource-affinities-type;
      }
      description "Affinities usage";
    }
    choice style {
      description
        "Path affinities representation style";
      case value {
        leaf value {
          type te-types:admin-groups;
          description
            "Bitmap indicating what bits are of significance";
        }
      }
      case named {
        list affinity-names {
          key "name";
          leaf name {
            type string;
            description "Affinity name";
          }
          description "List of named affinities";
        }
      }
    }
  }

  grouping path-affinities {
    description "Path affinities grouping";
    reference "RFC 3209";
    container path-affinities {
      description "Path affinities container";
      list constraints {
        key "usage";
        description "List of named affinity constraints";
        uses path-affinities-contents_config;
      }
    }
  }

  grouping path-srlgs-values_config {
    description "Path SRLG values properties grouping";
    reference "RFC4203 and RFC5307";
    leaf usage {
      type identityref {
        base te-types:route-exclude-srlg;
      }
      description "SRLG usage";
    }
    leaf-list values {
      type te-types:srlg;
      description "SRLG value";
      reference "RFC4203 and RFC5307";
    }
  }

  grouping path-srlgs {
    description "Path SRLG properties grouping";
    container path-srlgs {
      description "Path SRLG properties container";
      choice style {
        description "Type of SRLG representation";
        case values {
          uses path-srlgs-values_config;
        }
        case named {
          container constraints {
            description "SRLG named constraints";
            list constraint {
              key "usage";
              leaf usage {
                type identityref {
                  base te-types:route-exclude-srlg;
                }
                description "SRLG usage";
              }
              container constraint {
                description "Container for named SRLG list";
                list srlg-names {
                  key "name";
                  leaf name {
                    type string;
                    description "The SRLG name";
                  }
                  description "List named SRLGs";
                }
              }
              description "List of named SRLG constraints";
            }
          }
        }
      }
    }
```

```
  }

  grouping p2p-reverse-primary-path-properties {
    description "tunnel path properties.";
    reference "RFC7551";
    container p2p-reverse-primary-path {
      description "Tunnel reverse primary path properties";
      uses p2p-path-reverse-properties_config;
      uses path-constraints-common_config;
      container state {
        config false;
        description
          "Configuration applied parameters and state";
        uses p2p-path-properties_state;
      }
      container p2p-reverse-secondary-path {
        description "Tunnel reverse secondary path properties";
        uses p2p-reverse-path-candidate-secondary-path-config;
      }
    }
  }

  grouping p2p-secondary-path-properties {
    description "tunnel path properties.";
    uses p2p-path-properties_config;
    uses path-constraints-common_config;
    uses protection-restoration-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses p2p-path-properties_state;
    }
  }

  grouping p2p-primary-path-properties {
    description
      "TE tunnel primary path properties grouping";
    uses p2p-path-properties_config;
    uses path-constraints-common_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses p2p-path-properties_state;
    }
  }
  grouping path-properties_state {
    description "Computed path properties grouping";
    leaf metric-type {
      type identityref {
        base te-types:path-metric-type;
      }
      description "TE path metric type";
    }
    leaf accumulative-value {
      type uint64;
      description "TE path metric accumulative value";
    }
  }

  grouping path-properties {
    description "TE computed path properties grouping";
    container path-properties {
      description "The TE path computed properties";
      list path-metric {
        key metric-type;
        description "TE path metric type";
        leaf metric-type {
          type leafref {
            path "../state/metric-type";
          }
          description "TE path metric type";
        }
        container state {
          config false;
          description
            "Configuration applied parameters and state";
          uses path-properties_state;
        }
      }
      uses path-affinities;
      uses path-srlgs;
      container path-route-objects {
        description
          "Container for the list of computed route objects
           as returned by the computation engine";
        list path-computed-route-object {
          key index;
          description
            "List of computed route objects returned by the
             computation engine";
          leaf index {
            type leafref {
              path "../state/index";
            }
            description "Index of computed route object";
          }
          container state {
            config false;
            description
              "Configuration applied parameters and state";
            uses te-types:explicit-route-hop;
          }
        }
      }
      uses shared-resources-tunnels;
    }
  }

  grouping p2p-path-properties_state {
    description "TE per path state parameters";
    uses path-properties {
      description "The TE path computed properties";
    }
    container lsps {
      description "TE LSPs container";
      list lsp {
        key
          "source destination tunnel-id lsp-id "+
          "extended-tunnel-id";
        description "List of LSPs associated with the tunnel.";
        uses lsp-properties_state;
        uses shared-resources-tunnels_state;
        uses lsp-record-route-information_state;
        uses path-properties {
          description "The TE path actual properties";
        }
      }
    }
  }

  grouping p2p-path-properties-common_config {
    description
      "TE tunnel common path properties configuration grouping";
    leaf name {
      type string;
      description "TE path name";
    }
    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
```

```
        description
          "Signaling protocol used to set up this tunnel";
      }
      leaf path-computation-method {
        type identityref {
          base te-types:path-computation-method;
        }
        default te-types:path-locally-computed;
        description
          "The method used for computing the path, either
          locally computed, queried from a server or not
          computed at all (explicitly configured).";
      }
      leaf path-computation-server {
        when "../path-computation-method = "+
        "'te-types:path-externally-queried'" {
          description
            "The path-computation server when the path is
             externally queried";
        }
        type inet:ip-address;
        description
          "Address of the external path computation
           server";
      }
      leaf compute-only {
        type empty;
        description
          "When set, the path is computed and updated whenever
           the topology is updated. No resources are committed
           or reserved in the network.";
      }
      leaf use-path-computation {
        when "../path-computation-method =" +
        " 'te-types:path-locally-computed'";
        type boolean;
        description "A CSPF dynamically computed path";
      }
      leaf lockdown {
        type empty;
        description
          "Indicates no reoptimization to be attempted for
           this path.";
      }
      leaf path-scope {
        type identityref {
          base te-types:path-scope-type;
        }
        default te-types:path-scope-end-to-end;
        description "Path scope if segment or an end-to-end path";
      }
    }

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

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

    /* TE tunnel configuration data */
    grouping tunnel-p2mp-params_config {
      description
        "Configuration parameters relating to TE tunnel";
      leaf name {
        type string;
        description "TE tunnel name.";
      }
      leaf identifier {
        type uint16;
        description
          "TE tunnel Identifier.";
        reference "RFC 3209";
      }
      leaf description {
        type string;
        description
          "Textual description for this TE tunnel";
      }
    }

    grouping hierarchical-link_config {
      description
        "Hierarchical link configuration grouping";
      reference "RFC4206";
      leaf local-te-node-id {
        type te-types:te-node-id;
        description
          "Local TE node identifier";
      }
      leaf local-te-link-tp-id {
        type te-types:te-tp-id;
        description
          "Local TE link termination point identifier";
      }
      leaf remote-te-node-id {
        type te-types:te-node-id;
        description
          "Remote TE node identifier";
      }
      uses te-types:te-topology-identifier;
    }

    grouping hierarchical-link {
      description
        "Hierarchical link grouping";
      reference "RFC4206";
      container hierarchical-link {
        description
          "Identifies a hierarchical link (in client layer)
           that this tunnel is associated with.";
        uses hierarchical-link_config;
      }
    }

    grouping protection-restoration-params_state {
      description
        "Protection parameters grouping";
      leaf lockout-of-normal {
        type boolean;
        description
          "
            When set to 'True', it represents a lockout of normal
            traffic external command. When set to 'False', it
            represents a clear lockout of normal traffic external
            command. The lockout of normal traffic command applies
            to this Tunnel.
          ";
        reference
          "ITU-T G.808, RFC 4427";
      }
      leaf freeze {
        type boolean;
        description
          "
            When set to 'True', it represents a freeze external
            command. When set to 'False', it represents a clear
            freeze external command. The freeze command command
            applies to all the Tunnels which are sharing the
            protection resources with this Tunnel.
          ";
        reference
          "ITU-T G.808, RFC 4427";
      }
      leaf lsp-protection-role {
        type enumeration {
          enum working {
            description
              "A working LSP must be a primary LSP whilst a protecting
               LSP can be either a primary or a secondary LSP. Also,
               known as protected LSPs when working LSPs are associated
               with protecting LSPs.";
          }
          enum protecting {
            description
              "A secondary LSP is an LSP that has been provisioned
               in the control plane only; e.g. resource allocation
               has not been committed at the data plane";
          }
        }
        description "LSP role type";
        reference "rfc4872, section 4.2.1";
      }

      leaf lsp-protection-state {
        type identityref {
          base te-types:lsp-protection-state;
        }
        description
          "The state of the APS state machine controlling which
           tunnels is using the resources of the protecting LSP.";
      }
      leaf protection-group-ingress-node-id {
        type te-types:te-node-id;
        description
          "Indicates the te-node-id of the protection group
          ingress node when the APS state represents an extenal
          command (LoP, SF, MS) applied to it or a WTR timer
          running on it. If the external command is not applied to
          the ingress node or the WTR timer is not running on it,
          this attribute is not specified. If value 0.0.0.0 is used
          when the te-node-id of the protection group ingress node is
          unknown (e.g., because the ingress node is outside the scope
          of control of the server)";
      }
      leaf protection-group-egress-node-id {
        type te-types:te-node-id;
        description
          "Indicates the te-node-id of the protection group egress node
          when the APS state represents an extenal command (LoP, SF,
          MS) applied to it or a WTR timer running on it. If the
          external command is not applied to the ingress node or
          the WTR timer is not running on it, this attribute is not
          specified. If value 0.0.0.0 is used when the te-node-id of
          the protection group ingress node is unknown (e.g., because
          the ingress node is outside the scope of control of the
          server)";
      }
    }

    grouping protection-restoration-params_config {
      description "Protection and restoration parameters";
      container protection {
        description "Protection parameters";
        leaf enable {
          type boolean;
          default 'false';
          description
            "A flag to specify if LSP protection is enabled";
          reference "rfc4427";
        }
        leaf protection-type {
          type identityref {
            base te-types:lsp-protection-type;
          }
          description "LSP protection type.";
        }
        leaf protection-reversion-disable {
          type boolean;
          description "Disable protection reversion to working path";
        }
        leaf hold-off-time {
          type uint32;
          units "milli-seconds";
          default 0;
          description
            "The time between the declaration of an SF or SD condition
             and the initialization of the protection switching
             algorithm.";
          reference "rfc4427";
        }
        leaf wait-to-revert {
          type uint16;
          units seconds;
          description
           "Time to wait before attempting LSP reversion";
          reference "rfc4427";
        }
        leaf aps-signal-id {
          type uint8 {
            range "1..255";
          }
          description
            "The APS signal number used to reference the traffic of this
             tunnel. The default value for normal traffic is 1.
             The default value for extra-traffic is 255. If not specified,
             non-default values can be assigned by the server,
             if and only if, the server controls both endpoints.";
          reference
            "ITU-T G.808.1";
        }
      }
      container restoration {
        description "Restoration parameters";
        leaf enable {
          type boolean;
          default 'false';
          description
            "A flag to specify if LSP restoration is enabled";
          reference "rfc4427";
        }
        leaf restoration-type {
          type identityref {
            base te-types:lsp-restoration-type;
          }
          description "LSP restoration type.";
        }
        leaf restoration-scheme {
          type identityref {
            base te-types:restoration-scheme-type;
          }
          description "LSP restoration scheme.";
        }
        leaf restoration-reversion-disable {
          type boolean;
          description "Disable restoration reversion to working path";
        }
        leaf hold-off-time {
          type uint32;
          units "milli-seconds";
          description
            "The time between the declaration of an SF or SD condition
             and the initialization of the protection switching
             algorithm.";
          reference "rfc4427";
        }
        leaf wait-to-restore {
          type uint16;
          units seconds;
          description
           "Time to wait before attempting LSP restoration";
          reference "rfc4427";
        }
        leaf wait-to-revert {
          type uint16;
          units seconds;
          description
           "Time to wait before attempting LSP reversion";
          reference "rfc4427";
        }
      }
    }

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

    grouping tunnel-p2p-params_config {
      description
        "Configuration parameters relating to TE tunnel";
      leaf name {
        type string;
        description "TE tunnel name.";
      }
      leaf identifier {
        type uint16;
        description
```

```
        "TE tunnel Identifier.";
      reference "RFC3209";
    }
    leaf description {
      type string;
      description
        "Textual description for this TE tunnel";
    }
    leaf encoding {
      type identityref {
        base te-types:lsp-encoding-types;
      }
      description "LSP encoding type";
      reference "RFC3945";
    }
    leaf switching-type {
      type identityref {
        base te-types:switching-capabilities;
      }
      description "LSP switching type";
      reference "RFC3945";
    }
    leaf provisioning-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      description "TE tunnel administrative state.";
    }
    leaf preference {
      type uint8 {
        range "1..255";
      }
      description
        "Specifies a preference for this tunnel.
        A lower number signifies a better preference";
    }
    leaf reoptimize-timer {
      type uint16;
      units seconds;
      description
       "frequency of reoptimization of
        a traffic engineered LSP";
    }
    leaf source {
      type te-types:te-node-id;
      description
        "TE tunnel source node ID.";
    }
    leaf destination {
      type te-types:te-node-id;
      description
        "TE tunnel destination node ID";
    }
    leaf src-tp-id {
      type binary;
      description
        "TE tunnel source termination point identifier.";
    }
    leaf dst-tp-id {
      type binary;
      description
        "TE tunnel destination termination point identifier.";
    }
    leaf bidirectional {
      type boolean;
      default ’false’;
      description "TE tunnel bidirectional";
    }
    uses tunnel-p2p-associations_config;
    uses protection-restoration-params_config;
    uses te-types:tunnel-constraints_config;
    uses p2p-dependency-tunnels_config;
    uses hierarchical-link;
  }

  grouping tunnel-p2p-associations_config {
    description "TE tunnel association grouping";
    container association-objects {
      description "TE tunnel associations";
      list association-object {
        key "type ID source global-source";
        description "List of association base objects";
        reference "RFC4872";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
          reference "RFC4872";
        }
        leaf ID {
          type uint16;
          description "Association ID";
          reference "RFC4872";
        }
        leaf source {
          type inet:ip-address;
          description "Association source";
          reference "RFC4872";
        }
        leaf global-source {
          type inet:ip-address;
          description "Association global source";
          reference "RFC4872";
        }
      }
      list association-object-extended {
        key "type ID source global-source extended-ID";
        description "List of extended association objects";
        reference "RFC6780";
        leaf type {
          type identityref {
            base te-types:association-type;
          }
          description "Association type";
        }
        leaf ID {
          type uint16;
          description "Association ID";
          reference "RFC4872";
        }
        leaf source {
          type inet:ip-address;
          description "Association source";
        }
        leaf global-source {
          type inet:ip-address;
          description "Association global source";
          reference "RFC4872";
        }
        leaf extended-ID {
          type binary;
          description "Association extended ID";
          reference "RFC4872";
        }
      }
    }
  }

  grouping tunnel-p2p-params_state {
    description
      "State parameters relating to TE tunnel";
    leaf operational-state {
      type identityref {
        base te-types:tunnel-state-type;
      }
      default te-types:tunnel-state-up;
      description "TE tunnel administrative state.";
    }
  }

  grouping access-segment-info {
    description
      "info related to a segment";
    container forward {
      description
        "for the forward direction of this tunnel";
      uses te-types:label-set-info;
    }
    container reverse {
      description
        "for the reverse direction of this tunnel";
        uses te-types:label-set-info;
    }
  }

  grouping path-access-segment-info {
    description
      "If an end-to-end tunnel crosses multiple domains using
       the same technology, some additional constraints have to be
       taken in consideration in each domain";
    container path-in-segment {
      presence
        "The end-to-end tunnel starts in a previous domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses access-segment-info;
    }
    container path-out-segment {
      presence
        "The end-to-end tunnel is not terminated in this domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses access-segment-info;
    }
  }
  /* TE tunnel configuration/state grouping */
  grouping tunnel-p2mp-properties {
    description
      "Top level grouping for P2MP tunnel properties.";
    uses tunnel-p2mp-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      leaf operational-state {
        type identityref {
          base te-types:tunnel-state-type;
        }
        default te-types:tunnel-state-up;
        description "TE tunnel administrative state.";
      }
    }
  }

  grouping p2p-path-candidate-secondary-path-config {
    description
      "Configuration parameters relating to a secondary path which
      is a candidate for a particular primary path";

    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
  }

  grouping p2p-reverse-path-candidate-secondary-path-config {
    description
      "Configuration parameters relating to a secondary path which
      is a candidate for a particular primary path";
    leaf secondary-path {
      type leafref {
        path "../../../../../p2p-secondary-paths/" +
             "p2p-secondary-path/name";
      }
      description
        "A reference to the secondary path that should be utilised
        when the containing primary path option is in use";
    }

    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
  }

  grouping p2p-path-candidate-secondary-path-state {
    description
      "Operational state parameters relating to a secondary path
      which is a candidate for a particular primary path";

    leaf active {
      type boolean;
      description
        "Indicates the current active path option that has
        been selected of the candidate secondary paths";
    }
  }

  grouping tunnel-p2p-properties {
    description
      "Top level grouping for tunnel properties.";
    uses tunnel-p2p-params_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses tunnel-p2p-params_state;
    }
    container p2p-primary-paths {
      description "Set of P2P primary aths container";
      list p2p-primary-path {
        key "name";
        description
          "List of primary paths for this tunnel.";
        uses p2p-primary-path-properties;
        uses p2p-reverse-primary-path-properties;
        container candidate-p2p-secondary-paths {
          description
            "The set of candidate secondary paths which may be used
            for this primary path. When secondary paths are specified
            in the list the path of the secondary LSP in use must be
            restricted to those path options referenced. The
            priority of the secondary paths is specified within the
            list. Higher priority values are less preferred - that is
            to say that a path with priority 0 is the most preferred
            path. In the case that the list is empty, any secondary
            path option may be utilised when the current primary path
            is in use.";
          list candidate-p2p-secondary-path {
            key "secondary-path";
            description
              "List of secondary paths for this tunnel.";
            uses p2p-path-candidate-secondary-path-config;

            container state {
              config false;
              description
                "Configuration applied parameters and state";
              uses p2p-path-candidate-secondary-path-state;
            }
          }
        }
      }
    }
    container p2p-secondary-paths {
      description "Set of P2P secondary paths container";
      list p2p-secondary-path {
        key "name";
        description
          "List of secondary paths for this tunnel.";
        uses p2p-secondary-path-properties;
      }
    }
  }

  grouping shared-resources-tunnels_state {
    description
      "The specific tunnel that is using the shared secondary path
       resources";
    leaf lsp-shared-resources-tunnel {
      type te:tunnel-ref;
      description
```

```
          "Reference to the tunnel that sharing secondary path
          resources with this tunnel";
      }
    }
    grouping shared-resources-tunnels {
      description
        "Set of tunnels that share secondary path resources with
        this tunnnel";
      container shared-resources-tunnels {
        description
          "Set of tunnels that share secondary path resources with
          this tunnnel";
        leaf-list lsp-shared-resources-tunnel {
          type te:tunnel-ref;
          description
            "Reference to the tunnel that sharing secondary path
            resources with this tunnel";
        }
      }
    }

    grouping tunnel-actions {
      description "Tunnel actions";
      action tunnel-action {
        description "Tunnel action";
        input {
          leaf action-type {
            type identityref {
              base te-types:tunnel-action-type;
            }
            description "Tunnel action type";
          }
        }
        output {
          leaf action-result {
            type identityref {
              base te-types:te-action-result;
            }
            description "The result of the RPC operation";
          }
        }
      }
    }
    grouping tunnel-protection-actions {
      description
        "Protection external command actions";
      action protection-external-commands {
        input {
```

```
        leaf protection-external-command {
          type identityref {
            base te-types:protection-external-commands;
          }
          description
            "Protection external command";
        }
        leaf protection-group-ingress-node-id {
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on ingress node.
             By default, if neither ingress nor egress node-id
             is set, the the action applies to ingress node only.";
        }
        leaf protection-group-egress-node-id {
          type te-types:te-node-id;
          description
            "When specified, indicates whether the action is
             applied on egress node.
             By default, if neither ingress nor egress node-id
             is set, the the action applies to ingress node only.";
        }
        leaf path-ref {
          type path-ref;
          description
            "Indicates to which path the external command applies to.";
        }
        leaf traffic-type {
          type enumeration {
            enum normal-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the normal traffic (this Tunnel).";
            }
            enum null-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the null traffic.";
            }
            enum extra-traffic {
              description
                "The manual-switch or forced-switch command applies to
                 the extra traffic (the extra-traffic Tunnel sharing
                 protection bandwidth with this Tunnel).";
            }
          }
          description
```

```
            "Indicates whether the manual-switch or forced-switch
             commands applies to the normal traffic, the null traffic
             or the extra-traffic.";
          reference
            "ITU-T G.808, RFC 4427";
        }
        leaf extra-traffic-tunnel-ref {
          type te:tunnel-ref;
          description
            "In case there are multiple extra-traffic tunnels sharing
             protection bandwidth with this Tunnel (m:n protection),
             represents which extra-traffic Tunnel the manual-switch or
             forced-switch to extra-traffic command applies to.";
        }
      }
    }
  }

  /*** End of TE tunnel groupings ***/

  /**
   * LSP related generic groupings
   */
  grouping lsp-record-route-information_state {
    description "recorded route information grouping";
    container lsp-record-route-subobjects {
      description "RSVP recorded route object information";
      list record-route-subobject {
        when "../../origin-type = ’ingress’" {
          description "Applicable on non-ingress LSPs only";
        }
        key "index";
        description "Record route sub-object list";
        uses te-types:record-route-subobject_state;
      }
    }
  }

  grouping lsps-state-grouping {
    description
      "LSPs state operational data grouping";
    container lsps-state {
      config false;
      description "TE LSPs state container";
      list lsp {
        key
          "source destination tunnel-id lsp-id "+
          "extended-tunnel-id";
        description "List of LSPs associated with the tunnel.";
        uses lsp-properties_state;
        uses lsp-record-route-information_state;
      }
    }
  }

  /*** End of TE LSP groupings ***/

  /**
   * TE global generic groupings
   */

  /* Global named admin-groups configuration data */
  grouping named-admin-groups_config {
    description
      "Global named administrative groups configuration
      grouping";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
        interface named admin-group";
    }
    leaf bit-position {
      type uint32;
      description
        "Bit position representing the administrative group";
      reference "RFC3209 and RFC7308";
    }
  }
  grouping named-admin-groups {
    description
      "Global named administrative groups configuration
      grouping";
    container named-admin-groups {
      description "TE named admin groups container";
      list named-admin-group {
        if-feature te-types:extended-admin-groups;
        if-feature te-types:named-extended-admin-groups;
        key "name";
        description
          "List of named TE admin-groups";
        uses named-admin-groups_config;
      }
    }
  }
  /* Global named admin-srlgs configuration data */
  grouping named-srlgs_config {
    description
      "Global named SRLGs configuration grouping";
    leaf name {
      type string;
      description
        "A string name that uniquely identifies a TE
        interface named srlg";
    }
    leaf group {
      type te-types:srlg;
      description "An SRLG value";
    }
    leaf cost {
      type uint32;
      description
        "SRLG associated cost. Used during path to append
         the path cost when traversing a link with this SRLG";
    }
  }

  grouping named-srlgs {
    description
      "Global named SRLGs configuration grouping";
    container named-srlgs {
      description "TE named SRLGs container";
      list named-srlg {
        if-feature te-types:named-srlg-groups;
        key "name";
        description
          "A list of named SRLG groups";
        uses named-srlgs_config;
      }
    }
  }

  /* Global named paths constraints configuration data */
  grouping path-constraints_state {
    description
      "TE path constraints state";
    leaf bandwidth-generic_state {
      type te-types:te-bandwidth;
      description
        "A technology agnostic requested bandwidth to use
         for path computation";
    }
    leaf disjointness_state {
      type te-types:te-path-disjointness;
      description
        "The type of resource disjointness.";
    }
  }

  grouping path-constraints-common_config {
    description
      "Global named path constraints configuration
      grouping";
    uses te-types:common-path-constraints-attributes;
    uses te-types:generic-path-disjointness;
    uses te-types:path-route-objects;
    uses shared-resources-tunnels {
      description
        "Set of tunnels that are allowed to share secondary path
         resources of this tunnel";
    }
    uses path-access-segment-info {
      description
        "Tunnel constraints induced by other segments.";
    }
  }

  grouping path-constraints {
    description "Per path constraints";
    uses path-constraints-common_config;
    container state {
      config false;
      description
        "Configuration applied parameters and state";
      uses path-constraints_state;
    }
  }

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

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

  /* TE tunnels container data */
  grouping tunnels-grouping {
    description
      "Tunnels TE configuration data grouping";
    container tunnels {
      description
        "Tunnels TE configuration data container";

      list tunnel {
        key "name";
        description "P2P TE tunnels list.";
        uses tunnel-p2p-properties;
        uses tunnel-actions;
        uses tunnel-protection-actions;
      }
      list tunnel-p2mp {
        key "name";
        unique "identifier";
        description "P2MP TE tunnels list.";
        uses tunnel-p2mp-properties;
      }
    }
  }

  /* TE LSPs ephemeral state container data */
  grouping lsp-properties_state {
    description
      "LSPs state operational data grouping";
    leaf source {
      type inet:ip-address;
      description
        "Tunnel sender address extracted from
        SENDER_TEMPLATE  object";
      reference "RFC3209";
    }
    leaf destination {
      type inet:ip-address;
      description
        "Tunnel endpoint address extracted from
        SESSION object";
      reference "RFC3209";
    }
    leaf tunnel-id {
      type uint16;
      description
        "Tunnel identifier used in the SESSION
        that remains constant over the life
        of the tunnel.";
      reference "RFC3209";
    }
    leaf lsp-id {
      type uint16;
      description
        "Identifier used in the SENDER_TEMPLATE
        and the FILTER_SPEC that can be changed
        to allow a sender to share resources with
        itself.";
      reference "RFC3209";
    }
    leaf extended-tunnel-id {
      type inet:ip-address;
       description
        "Extended Tunnel ID of the LSP.";
      reference "RFC3209";
    }
    leaf operational-state {
      type identityref {
        base te-types:lsp-state-type;
      }
      description "LSP operational state.";
    }
    leaf path-setup-protocol {
      type identityref {
        base te-types:path-signaling-type;
      }
      description
        "Signaling protocol used to set up this tunnel";
    }
    leaf origin-type {
      type enumeration {
        enum ingress {
          description
            "Origin ingress";
        }
        enum egress {
          description
            "Origin egress";
        }
        enum transit {
          description
            "transit";
        }
      }
      description
        "Origin type of LSP relative to the location
        of the local switch in the path.";
    }

    leaf lsp-resource-status {
      type enumeration {
        enum primary {
          description
            "A primary LSP is a fully established LSP for
             which the resource allocation has been committed
             at the data plane";
        }
        enum secondary {
          description
            "A secondary LSP is an LSP that has been provisioned
             in the control plane only; e.g. resource allocation
             has not been committed at the data plane";
        }
      }
      description "LSP resource allocation type";
      reference "rfc4872, section 4.2.1";
    }

    uses protection-restoration-params_state;
  }
  /*** End of TE global groupings ***/

  /**
```

```
      * TE configurations container
      */
     container te {
       presence "Enable TE feature.";
       description
          "TE global container.";

       /* TE Global Configuration Data */
       uses globals-grouping;

       /* TE Tunnel Configuration Data */
       uses tunnels-grouping;

       /* TE LSPs State Data */
       uses lsps-state-grouping;

     }

     /* TE Global RPCs/execution Data */
     rpc globals-rpc {
       description
         "Execution data for TE global.";
     }

     /* TE interfaces RPCs/execution Data */
     rpc interfaces-rpc {
       description
         "Execution data for TE interfaces.";
     }

     /* TE Tunnel RPCs/execution Data */
     rpc tunnels-rpc {
       description "TE tunnels RPC nodes";
       input {
         container tunnel-info {
           description "Tunnel Identification";
           choice type {
             description "Tunnel information type";
             case tunnel-p2p {
               leaf p2p-id {
                 type te:tunnel-ref;
                 description "P2P TE tunnel";
               }
             }
             case tunnel-p2mp {
               leaf p2mp-id {
                 type te:tunnel-p2mp-ref;
                 description "P2MP TE tunnel";
```

```
            }
          }
        }
      }
    }
    output {
      container result {
        description
          "The container result of the RPC operation";
        leaf result {
          type enumeration {
            enum success {
              description "Origin ingress";
            }
            enum in-progress {
              description "Origin egress";
            }
            enum fail {
              description "transit";
            }
          }
          description "The result of the RPC operation";
        }
      }
    }
  }

  /* TE Global Notification Data */
  notification globals-notif {
    description
      "Notification messages for Global TE.";
  }

  /* TE Tunnel Notification Data */
  notification tunnels-notif {
    description
      "Notification messages for TE tunnels.";
  }
}
<CODE ENDS>
```

Figure 8: TE generic YANG module

```
   <CODE BEGINS> file "ietf-te-device@2018-02-15.yang"
   module ietf-te-device {

     namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";
     /* Replace with IANA when assigned */
     prefix "te-dev";

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

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

     import ietf-interfaces {
       prefix if;
     }

     import ietf-inet-types {
       prefix inet;
     }

     import ietf-routing-types {
       prefix "rt-types";
     }

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

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

        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>

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

        Editor:   Tarek Saad
                  <mailto:tsaad@cisco.com>

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

        Editor:   Vishnu Pavan Beeram
                  <mailto:vbeeram@juniper.net>
        Editor:   Himanshu Shah
                  <mailto:hshah@ciena.com>

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

        Editor:   Xia Chen
                  <mailto:jescia.chenxia@huawei.com>

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

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

     description
       "YANG data module for TE device configurations,
       state, RPC and notifications.";

     revision "2018-02-15" {
       description "Latest update to TE device YANG module.";
       reference "TBA";
     }


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

         leaf time-to-install {
           type uint32;
           units seconds;
           description
             "lsp installation delay time";
         }
         leaf time-to-destroy {
           type uint32;
           units seconds;
           description
             "lsp expiration delay time";
         }
       }

       container downstream-info {
         when "../te:origin-type != 'egress'" {
           description "Applicable to ingress LSPs only";
         }
         description
           "downstream information";

         leaf nhop {
           type inet:ip-address;
           description
             "downstream nexthop.";
         }

         leaf outgoing-interface {
           type if:interface-ref;
           description
             "downstream interface.";
         }

         leaf neighbor {
           type inet:ip-address;
           description
             "downstream neighbor.";
         }

         leaf label {
           type rt-types:generalized-label;
           description
             "downstream label.";
         }
       }

       container upstream-info {
         when "../te:origin-type != 'ingress'" {
           description "Applicable to non-ingress LSPs only";
         }
         description
           "upstream information";

         leaf phop {
           type inet:ip-address;
           description
             "upstream nexthop or previous-hop.";
         }

         leaf neighbor {
           type inet:ip-address;
           description
             "upstream neighbor.";
         }

         leaf label {
           type rt-types:generalized-label;
           description
             "upstream label.";
         }
       }
     }

     /**
      * Device general groupings.
      */
     grouping tunnel-device_config {
       description "Device TE tunnel configs";
       leaf path-invalidation-action {
         type identityref {
           base te-types:path-invalidation-action-type;
         }
         description "Tunnel path invalidtion action";
       }
     }

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

     /**
      * TE global device generic groupings
      */

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

     /**
      * TE interface device generic groupings
      */
     grouping te-admin-groups_config {
       description
         "TE interface affinities grouping";
       choice admin-group-type {
         description
           "TE interface administrative groups
           representation type";
         case value-admin-groups {
           choice value-admin-group-type {
             description "choice of admin-groups";
             case admin-groups {
               description
                 "Administrative group/Resource
                 class/Color.";
               leaf admin-group {
                 type te-types:admin-group;
                 description
                   "TE interface administrative group";
               }
             }
             case extended-admin-groups {
               if-feature te-types:extended-admin-groups;
               description
                 "Extended administrative group/Resource
                 class/Color.";
               leaf extended-admin-group {
                 type te-types:extended-admin-group;
                 description
                   "TE interface extended administrativei
                   group";
               }
             }
           }
         }
         case named-admin-groups {
           list named-admin-groups {
             if-feature te-types:extended-admin-groups;
             if-feature te-types:named-extended-admin-groups;
             key named-admin-group;
             description
               "A list of named admin-group entries";
             leaf named-admin-group {
               type leafref {
                 path "../../../../te:globals/" +
                   "te:named-admin-groups/te:named-admin-group/" +
                   "te:name";
               }
               description "A named admin-group entry";
             }
           }
         }
       }
     }

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

     grouping te-igp-flooding-bandwidth_config {
       description
         "Configurable items for igp flooding bandwidth
         threshold configuration.";
       leaf threshold-type {
         type enumeration {
           enum DELTA {
             description
               "DELTA indicates that the local
               system should flood IGP updates when a
               change in reserved bandwidth >= the specified
               delta occurs on the interface.";
           }
           enum THRESHOLD_CROSSED {
             description
               "THRESHOLD-CROSSED indicates that
               the local system should trigger an update (and
               hence flood) the reserved bandwidth when the
               reserved bandwidth changes such that it crosses,
               or becomes equal to one of the threshold values.";
           }
         }
         description
           "The type of threshold that should be used to specify the
           values at which bandwidth is flooded. DELTA indicates that
           the local system should flood IGP updates when a change in
           reserved bandwidth >= the specified delta occurs on the
           interface. Where THRESHOLD_CROSSED is specified, the local
           system should trigger an update (and hence flood) the
           reserved bandwidth when the reserved bandwidth changes such
           that it crosses, or becomes equal to one of the threshold
           values";
       }

       leaf delta-percentage {
         when "../threshold-type = 'DELTA'" {
           description
             "The percentage delta can only be specified when the
             threshold type is specified to be a percentage delta of
             the reserved bandwidth";
         }
         type te-types:percentage;
         description
           "The percentage of the maximum-reservable-bandwidth
           considered as the delta that results in an IGP update
           being flooded";
       }
       leaf threshold-specification {
         when "../threshold-type = 'THRESHOLD_CROSSED'" {
           description
             "The selection of whether mirrored or separate threshold
             values are to be used requires user specified thresholds to
             be set";
         }
         type enumeration {
           enum MIRRORED_UP_DOWN {
             description
               "MIRRORED_UP_DOWN indicates that a single set of
               threshold values should be used for both increasing
               and decreasing bandwidth when determining whether
               to trigger updated bandwidth values to be flooded
```

```
                in the IGP TE extensions.";
            }
            enum SEPARATE_UP_DOWN {
              description
                "SEPARATE_UP_DOWN indicates that a separate
                threshold values should be used for the increasing
                and decreasing bandwidth when determining whether
                to trigger updated bandwidth values to be flooded
                in the IGP TE extensions.";
            }
          }
          description
            "This value specifies whether a single set of threshold
            values should be used for both increasing and decreasing
            bandwidth when determining whether to trigger updated
            bandwidth values to be flooded in the IGP TE extensions.
            MIRRORED-UP-DOWN indicates that a single value (or set of
            values) should be used for both increasing and decreasing
            values, where SEPARATE-UP-DOWN specifies that the increasing
            and decreasing values will be separately specified";
        }

        leaf-list up-thresholds {
          when "../threshold-type = 'THRESHOLD_CROSSED'" +
            "and ../threshold-specification = 'SEPARATE_UP_DOWN'" {
              description
                "A list of up-thresholds can only be specified when the
                bandwidth update is triggered based on crossing a
                threshold and separate up and down thresholds are
                required";
          }
          type te-types:percentage;
          description
            "The thresholds (expressed as a percentage of the maximum
            reservable bandwidth) at which bandwidth updates are to be
            triggered when the bandwidth is increasing.";
        }

        leaf-list down-thresholds {
          when "../threshold-type = 'THRESHOLD_CROSSED'" +
            "and ../threshold-specification = 'SEPARATE_UP_DOWN'" {
              description
                "A list of down-thresholds can only be specified when the
                bandwidth update is triggered based on crossing a
                threshold and separate up and down thresholds are
                required";
          }
          type te-types:percentage;
```

```
         description
           "The thresholds (expressed as a percentage of the maximum
           reservable bandwidth) at which bandwidth updates are to be
           triggered when the bandwidth is decreasing.";
       }

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

     /* TE interface metric */
     grouping te-metric_config {
       description "Interface TE metric grouping";
       leaf te-metric {
         type te-types:te-metric;
         description "Interface TE metric.";
       }
     }

     /* TE interface switching capabilities */
     grouping te-switching-cap_config {
       description
         "TE interface switching capabilities";
       list switching-capabilities {
         key "switching-capability";
         description
           "List of interface capabilities for this interface";
         leaf switching-capability {
           type identityref {
             base te-types:switching-capabilities;
           }
           description
             "Switching Capability for this interface";
         }
         leaf encoding {
           type identityref {
             base te-types:lsp-encoding-types;
           }
           description
             "Encoding supported by this interface";
         }
       }
     }

     grouping te-advertisements_state {
       description
         "TE interface advertisements state grouping";
       container te-advertisements_state {
         description
           "TE interface advertisements state container";
         leaf flood-interval {
           type uint32;
           description
             "The periodic flooding interval";
         }
         leaf last-flooded-time {
           type uint32;
           units seconds;
           description
             "Time elapsed since last flooding in seconds";
         }
         leaf next-flooded-time {
           type uint32;
           units seconds;
           description
             "Time remained for next flooding in seconds";
         }
         leaf last-flooded-trigger {
           type enumeration {
             enum link-up {
               description "Link-up flooding trigger";
             }
             enum link-down {
               description "Link-up flooding trigger";
             }
             enum threshold-up {
               description
                 "Bandwidth reservation up threshold";
             }
             enum threshold-down {
               description
                 "Bandwidth reservation down threshold";
             }
             enum bandwidth-change {
               description "Banwidth capacity change";
             }
             enum user-initiated {
               description "Initiated by user";
             }
             enum srlg-change {
               description "SRLG property change";
             }
             enum periodic-timer {
               description "Periodic timer expired";
             }
           }
           description "Trigger for the last flood";
         }
         list advertized-level-areas {
           key level-area;
           description
             "List of areas the TE interface is advertised
             in";
           leaf level-area {
             type uint32;
             description
               "The IGP area or level where the TE
               interface state is advertised in";
           }
         }
       }
     }

     /* TE interface attributes grouping */
     grouping te-attributes {
       description "TE attributes configuration grouping";
       uses te-metric_config;
       uses te-admin-groups_config;
       uses te-srlgs_config;
       uses te-igp-flooding-bandwidth_config;
       uses te-switching-cap_config;
       container state {
         config false;
         description
           "State parameters for interface TE metric";
         uses te-advertisements_state;
       }
     }

     grouping te-all-attributes {
       description
         "TE attributes configuration grouping for all
          interfaces";
       uses te-igp-flooding-bandwidth_config;
     }
     /*** End of TE interfaces device groupings ***/


     /**
      * TE device augmentations
      */
     augment "/te:te" {
       description "TE global container.";
       /* TE Interface Configuration Data */
       uses interfaces-grouping;
     }

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

     /* TE tunnels device configuration augmentation */
     augment "/te:te/te:tunnels/te:tunnel" {
       description
         "Tunnel device dependent augmentation";
       uses lsp-device-timers_config;
     }
     augment "/te:te/te:tunnels/te:tunnel/te:state" {
       description
         "Tunnel device dependent augmentation";
       uses lsp-device-timers_config;
     }

     /* TE LSPs device state augmentation */
     augment "/te:te/te:lsps-state/te:lsp" {
        description
          "LSP device dependent augmentation";
        uses lsps-device_state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
       "/te:p2p-secondary-path/te:state/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device_state;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
       "/te:p2p-primary-path/te:state/te:lsps/te:lsp" {
       description
         "LSP device dependent augmentation";
       uses lsps-device_state;
     }

     /* TE interfaces RPCs/execution Data */
     rpc interfaces-rpc {
       description
         "Execution data for TE interfaces.";
     }

     /* TE Interfaces Notification Data */
     notification interfaces-notif {
       description
         "Notification messages for TE interfaces.";
     }
   }
   <CODE ENDS>
```

Figure 9: TE MPLS specific types YANG module

```
<CODE BEGINS> file "ietf-te-mpls@2018-02-15.yang"
module ietf-te-mpls {

  namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls";

  /* Replace with IANA when assigned */
  prefix "te-mpls";

  /* Import TE base model */
  import ietf-te {
    prefix te;
  }

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

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

  import ietf-mpls-static {
    prefix mpls-static;
  }

  import ietf-inet-types {
    prefix inet;
  }

  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>

     WG Chair: Lou Berger
               <mailto:lberger@labn.net>

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

     Editor:   Tarek Saad
               <mailto:tsaad@cisco.com>

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

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

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

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

     Editor:   Xia Chen
               <mailto:jescia.chenxia@huawei.com>

     Editor:   Raqib Jones
               <mailto:raqib@Brocade.com>
     Editor:   Bin Wen
               <mailto:Bin_Wen@cable.comcast.com>";

  description
    "YANG data module for MPLS TE configurations,
    state, RPC and notifications.";

  revision "2018-02-15" {
    description "Latest update to MPLS TE YANG module.";
    reference "TBD";
  }

  /* MPLS TE tunnel properties*/

  grouping tunnel-igp-shortcut_config {
    description "TE tunnel IGP shortcut configs";
    leaf shortcut-eligible {
      type boolean;
      default "true";
      description
        "Whether this LSP is considered to be eligible for us as a
        shortcut in the IGP. In the case that this leaf is set to
        true, the IGP SPF calculation uses the metric specified to
        determine whether traffic should be carried over this LSP";
    }
    leaf metric-type {
      type identityref {
        base te-types:LSP_METRIC_TYPE;
      }
      default te-types:LSP_METRIC_INHERITED;
      description
        "The type of metric specification that should be used to set
        the LSP(s) metric";
    }
    leaf metric {
      type int32;
      description
        "The value of the metric that should be specified. The value
        supplied in this leaf is used in conjunction with the metric
        type to determine the value of the metric used by the system.
        Where the metric-type is set to LSP_METRIC_ABSOLUTE - the
        value of this leaf is used directly; where it is set to
        LSP_METRIC_RELATIVE, the relevant (positive or negative)
        offset is used to formulate the metric; where metric-type
        is LSP_METRIC_INHERITED, the value of this leaf is not
        utilised";
    }
    leaf-list routing-afs {
      type inet:ip-version;
      description
        "Address families";
    }
  }

  grouping tunnel-igp-shortcuts {
    description
      "TE tunnel IGP shortcut grouping";
    container tunnel-igp-shortcut {
      description
        "Tunnel IGP shortcut properties";
      uses tunnel-igp-shortcut_config;
    }
  }

  grouping tunnel-forwarding-adjacency_configs {
    description "Tunnel forwarding adjacency grouping";
    leaf binding-label {
      type rt-types:mpls-label;
      description "MPLS tunnel binding label";
    }
    leaf load-share {
      type uint32 {
        range "1..4294967295";
      }
      description "ECMP tunnel forwarding
        load-share factor.";
    }
    leaf policy-class {
      type uint8 {
        range "1..7";
      }
      description
        "The class associated with this tunnel";
    }
  }

  grouping tunnel-forwarding-adjacency {
    description "Properties for using tunnel in forwarding.";
    container forwarding {
      description
        "Tunnel forwarding properties container";
      uses tunnel-forwarding-adjacency_configs;
    }
  }

  /*** End of MPLS TE tunnel configuration/state */
  grouping te-lsp-auto-bandwidth_config {
    description
      "Configuration parameters related to autobandwidth";

    leaf enabled {
      type boolean;
      default false;
      description
        "enables mpls auto-bandwidth on the
         lsp";
    }

    leaf min-bw {
      type te-mpls-types:bandwidth-kbps;
      description
        "set the minimum bandwidth in Kbps for an
         auto-bandwidth LSP";
    }

    leaf max-bw {
      type te-mpls-types:bandwidth-kbps;
      description
        "set the maximum bandwidth in Kbps for an
         auto-bandwidth LSP";
    }

    leaf adjust-interval {
      type uint32;
      description
        "time in seconds between adjustments to
         LSP bandwidth";
    }

    leaf adjust-threshold {
      type te-types:percentage;
      description
        "percentage difference between the LSP’s
         specified bandwidth and its current bandwidth
         allocation -- if the difference is greater than the
         specified percentage, auto-bandwidth adjustment is
         triggered";
    }
  }

  grouping te-lsp-overflow_config {
    description
     "configuration for mpls lsp bandwidth
      overflow adjustment";
```

```
        leaf enabled {
          type boolean;
          default false;
          description
           "enables mpls lsp bandwidth overflow
            adjustment on the lsp";
        }

        leaf overflow-threshold {
          type te-types:percentage;
          description
           "bandwidth percentage change to trigger
            an overflow event";

        }

        leaf trigger-event-count {
          type uint16;
          description
           "number of consecutive overflow sample
            events needed to trigger an overflow adjustment";
        }
      }

      grouping te-lsp-underflow_config {
        description
          "configuration for mpls lsp bandwidth
          underflow adjustment";

        leaf enabled {
          type boolean;
          default false;
          description
           "enables bandwidth underflow
            adjustment on the lsp";
        }

        leaf underflow-threshold {
          type te-types:percentage;
          description
           "bandwidth percentage change to trigger
            and underflow event";
        }

        leaf trigger-event-count {
          type uint16;
          description
           "number of consecutive underflow sample
            events needed to trigger an underflow adjustment";
        }
      }
      grouping te-tunnel-bandwidth_config {
        description
          "Configuration parameters related to bandwidth for a tunnel";

        leaf specification-type {
          type te-mpls-types:te-bandwidth-type;
          default SPECIFIED;
          description
            "The method used for settign the bandwidth, either explicitly
            specified or configured";
        }

        leaf set-bandwidth {
          when "../specification-type = 'SPECIFIED'" {
           description
             "The bandwidth value when bandwidth is explicitly
              specified";
          }
          type te-mpls-types:bandwidth-kbps;
          description
           "set bandwidth explicitly, e.g., using
            offline calculation";
        }
        leaf class-type {
          type te-types:te-ds-class;
          description
            "The Class-Type of traffic transported by the LSP.";
          reference "RFC4124: section-4.3.1";
        }
      }

      grouping te-tunnel-bandwidth_state {
        description
          "Operational state parameters relating to bandwidth for a tunnel";

        leaf signaled-bandwidth {
          type te-mpls-types:bandwidth-kbps;
          description
            "The currently signaled bandwidth of the LSP. In the case where
            the bandwidth is specified explicitly, then this will match the
            value of the set-bandwidth leaf; in cases where the bandwidth is
            dynamically computed by the system, the current value of the
            bandwidth should be reflected.";
        }
      }
      grouping tunnel-bandwidth_top {
        description
          "Top level grouping for specifying bandwidth for a tunnel";

        container bandwidth-mpls {
          description
            "Bandwidth configuration for TE LSPs";

          uses te-tunnel-bandwidth_config;

          container state {
            config false;
            description
              "State parameters related to bandwidth
              configuration of TE tunnels";
            uses te-tunnel-bandwidth_state;
          }

          container auto-bandwidth {
            when "../specification-type = 'AUTO'" {
              description
                "Include this container for auto bandwidth
                specific configuration";
            }
            description
              "Parameters related to auto-bandwidth";

            uses te-lsp-auto-bandwidth_config;

            container overflow {
              description
                "configuration of MPLS overflow bandwidth
                adjustement for the LSP";

              uses te-lsp-overflow_config;
            }

            container underflow {
              description
                "configuration of MPLS underflow bandwidth
                adjustement for the LSP";

              uses te-lsp-underflow_config;
            }
          }
        }
      }
      grouping te-path-bandwidth_top {
        description
          "Top level grouping for specifying bandwidth for a TE path";

        container bandwidth {
          description
            "Bandwidth configuration for TE LSPs";

          uses te-tunnel-bandwidth_config;
          container state {
            config false;
            description
              "State parameters related to bandwidth
              configuration of TE tunnels";
            uses te-tunnel-bandwidth_state;
          }
        }
      }



      /**
       * MPLS TE augmentations
       */

      /* MPLS TE tunnel augmentations */
      augment "/te:te/te:tunnels/te:tunnel" {
        description "MPLS TE tunnel config augmentations";
        uses tunnel-igp-shortcuts;
        uses tunnel-forwarding-adjacency;
        uses tunnel-bandwidth_top;
      }

      /* MPLS TE LSPs augmentations */
      augment "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-primary-paths/te:p2p-primary-path" {
        when "/te:te/te:tunnels/te:tunnel" +
          "/te:p2p-primary-paths/te:p2p-primary-path" +
          "/te:path-setup-protocol = 'te-types:path-setup-static'" {
          description
          "When the path is statically provisioned";
        }
        description "MPLS TE LSP augmentation";
        leaf static-lsp-name {
          type mpls-static:static-lsp-ref;
          description "Static LSP name";
        }
      }
      augment "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-primary-paths/te:p2p-primary-path/" +
              "te:state" {
        description "MPLS TE LSP augmentation";
        leaf static-lsp-name {
          type mpls-static:static-lsp-ref;
          description "Static LSP name";
        }
      }
      augment "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-secondary-paths/te:p2p-secondary-path" {
        when "/te:te/te:tunnels/te:tunnel" +
          "/te:p2p-secondary-paths/te:p2p-secondary-path/" +
          "te:path-setup-protocol = 'te-types:path-setup-static'" {
          description
          "When the path is statically provisioned";
        }
        description "MPLS TE LSP augmentation";
        leaf static-lsp-name {
          type mpls-static:static-lsp-ref;
          description "Static LSP name";
        }
      }
      augment "/te:te/te:tunnels/te:tunnel/" +
              "te:p2p-secondary-paths/te:p2p-secondary-path/" +
              "te:state" {
        description "MPLS TE LSP augmentation";
        leaf static-lsp-name {
          type mpls-static:static-lsp-ref;
          description "Static LSP name";
        }
      }

      augment "/te:te/te:globals/te:named-path-constraints/" +
              "te:named-path-constraint" {
        description "foo";
        uses te-path-bandwidth_top;
      }
    }
    <CODE ENDS>
```

Figure 10: TE MPLS YANG module

```
   <CODE BEGINS> file "ietf-te-mpls-types@2018-02-15.yang"
   module ietf-te-mpls-types {

     namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls-types";
     /* Replace with IANA when assigned */
     prefix "te-mpls-types";

     organization
       "IETF TEAS Working Group";

     contact "Fill me";

     description
       "This module contains a collection of generally
       useful TE specific YANG data type defintions.";

     revision "2018-02-15" {
       description "Latest revision of TE MPLS types";
       reference "RFC3209";
     }

     identity backup-protection-type {
       description
         "Base identity for backup protection type";
     }

     identity backup-protection-link {
       base backup-protection-type;
       description
         "backup provides link protection only";
     }

     identity backup-protection-node-link {
       base backup-protection-type;
       description
         "backup offers node (preferred) or link protection";
     }

     identity bc-model-type {
       description
         "Base identity for Diffserv-TE bandwidth constraint
         model type";
     }

     identity bc-model-rdm {
       base bc-model-type;
       description
         "Russian Doll bandwidth constraint model type.";
     }

     identity bc-model-mam {
       base bc-model-type;
       description
         "Maximum Allocation bandwidth constraint
         model type.";
     }

     identity bc-model-mar {
       base bc-model-type;
       description
         "Maximum Allocation with Reservation
         bandwidth constraint model type.";
     }

     typedef bandwidth-kbps {
       type uint64;
       units "Kbps";
       description
         "Bandwidth values expressed in kilobits per second";
     }

     typedef bandwidth-mbps {
       type uint64;
       units "Mbps";
       description
         "Bandwidth values expressed in megabits per second";
     }

     typedef bandwidth-gbps {
       type uint64;
       units "Gbps";
       description
         "Bandwidth values expressed in gigabits per second";
     }

     typedef te-bandwidth-type {
       type enumeration {
         enum SPECIFIED {
           description
             "Bandwidth is explicitly specified";
         }
         enum AUTO {
           description
             "Bandwidth is automatically computed";
         }
       }
       description
         "enumerated type for specifying whether bandwidth is
          explicitly specified or automatically computed";
     }
     typedef bfd-type {
       type enumeration {
         enum classical {
           description "BFD classical session type.";
         }
         enum seamless {
           description "BFD seamless session type.";
         }
       }
       default "classical";
       description
         "Type of BFD session";
     }

     typedef bfd-encap-mode-type {
       type enumeration {
         enum gal {
           description
             "BFD with GAL mode";
         }
         enum ip {
           description
             "BFD with IP mode";
         }
       }
       default ip;
       description
         "Possible BFD transport modes when running over TE
         LSPs.";
     }
   }
   <CODE ENDS>
```

Figure 11: TE MPLS types YANG module

```
   <CODE BEGINS> file "ietf-te-sr-mpls@2018-02-15.yang"
   module ietf-te-sr-mpls {

     namespace "urn:ietf:params:xml:ns:yang:ietf-te-sr-mpls";

     /* Replace with IANA when assigned */
     prefix "te-sr-mpls";

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

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

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

        WG Chair: Lou Berger
                  <mailto:lberger@labn.net>

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

        Editor:   Tarek Saad
                  <mailto:tsaad@cisco.com>

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

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

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

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

        Editor:   Xia Chen
                  <mailto:jescia.chenxia@huawei.com>

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

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

     description
       "YANG data module for MPLS TE configurations,
       state, RPC and notifications.";

     revision "2018-02-15" {
       description "Latest update to MPLS TE YANG module.";
       reference "TBD";
     }

     identity sr-protection-type {
       description
          "The Adj-SID base protection types";
     }

     identity sr-protection-type-protected {
       base sr-protection-type;
       description
          "The Adj-SID is eligible if protected";
     }

     identity sr-protection-type-unprotected {
       base sr-protection-type;
       description
          "The Adj-SID is eligible if unprotected";
     }

     identity sr-protection-type-any {
       base sr-protection-type;
       description
          "The Adj-SID is eligible if protected or unprotected";
     }

     typedef te-sid-selection-mode {
       type enumeration {
         enum ADJ_SID_ONLY {
           description
             "The SR-TE tunnel should only use adjacency SIDs
             to build the SID stack to be pushed for the LSP";
         }
         enum MIXED_MODE {
           description
             "The SR-TE tunnel can use a mix of adjacency
             and prefix SIDs to build the SID stack to be pushed
             to the LSP";
         }
       }
       description "SID selection mode type";
     }

     /* MPLS SR-TE tunnel properties*/
     grouping tunnel-sr-mpls-properties_config {
       description "MPLS TE SR tunnel properties";
       leaf path-signaling-type {
         type identityref {
           base te-types:path-signaling-type;
         }
         description "TE tunnel path signaling type";
       }
     }

     grouping te-sr-named-path-constraints_config {
       description
         "Configuration parameters relating to SR-TE LSPs";

       leaf sid-selection-mode {
         type te-sid-selection-mode;
         default MIXED_MODE;
         description
           "The restrictions placed on the SIDs to be selected by the
           calculation method for the explicit path when it is
           instantiated for a SR-TE LSP";
       }

       leaf sid-protection {
         type identityref {
           base sr-protection-type;
         }
         default sr-protection-type-any;
         description
           "When set to protected only SIDs that are
           protected are to be selected by the calculating method
           when the explicit path is instantiated by a SR-TE LSP.";
       }
     }

     grouping te-sr-named-path-constraints {
       description "Named TE SR path constraints grouping";
       uses te-sr-named-path-constraints_config;
     }

     /*** End of MPLS SR-TE tunnel configuration/state */

     /**
      * MPLS TE augmentations
      */
     augment "/te:te/te:globals/te:named-path-constraints" +
             "/te:named-path-constraint" {
       description
           "Augmentations for MPLS SR-TE config named constraints";
       uses te-sr-named-path-constraints;
     }
     /* MPLS TE tunnel augmentations */

     /* MPLS TE LSPs augmentations */
   }
   <CODE ENDS>
```

Figure 12: SR TE MPLS YANG module

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

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

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

URI: urn:ietf:params:xml:ns:yang:ietf-te-sr-mpls XML: N/A, the requested URI is an XML namespace.

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

URI: urn:ietf:params:xml:ns:yang:ietf-te-mpls-types XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-te namespace: urn:ietf:params:xml:ns:yang:ietf-te prefix: ietf-te reference: RFC3209

name: ietf-te-device namespace: urn:ietf:params:xml:ns:yang:ietf-te prefix: ietf-te-device reference: RFC3209

name: ietf-te-mpls namespace: urn:ietf:params:xml:ns:yang:ietf-te-mpls prefix: ietf-te-mpls reference: RFC3209

name: ietf-te-sr-mpls namespace: urn:ietf:params:xml:ns:yang:ietf-te-sr-mpls prefix: ietf-te-sr-mpls name: ietf-te-types namespace: urn:ietf:params:xml:ns:yang:ietf-te-types prefix: ietf-te-types reference: RFC3209

name: ietf-te-mpls-types namespace: urn:ietf:params:xml:ns:yang:ietf-te-mpls-types prefix: ietf-te-mpls-types reference: RFC3209

6. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC8341] provides means to restrict access for particular NETCONF

users to a pre-configured subset of all available NETCONF protocol operations and content.

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

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

"/te/tunnels": This list specifies the configured TE tunnels on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/lsps-state": This list specifies the state derived LSPs. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

"/te/interfaces": This list specifies the configured TE interfaces on a device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

7. Acknowledgement

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

The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua

Guo, Dhruv Dhody, Anurag Sharma, and Xian Zhang for their comments and providing valuable feedback on this document.

8. Contributors

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

9. Normative References

[I-D.ietf-teas-yang-rsvp]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp-09 (work in progress), May 2018.

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

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

[RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <https://www.rfc-editor.org/info/rfc3473>.

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

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

[RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <https://www.rfc-editor.org/info/rfc6107>.

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

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

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

[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>.

[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>.

Authors’ Addresses

Tarek Saad (editor)
Cisco Systems Inc

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Jabil

Email: Xufeng_Liu@jabil.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Himanshu Shah
Ciena

Email: hshah@ciena.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com