TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2019

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

October 20, 2018

A YANG Data Model for Traffic Engineering Tunnels and Interfaces
draft-ietf-teas-yang-te-17

Abstract

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

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 23, 2019.

Table of Contents

1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG has proved relevant beyond its initial confines, as bindings to other interfaces (e.g. RESTCONF [RFC8040]) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document describes YANG data model for TE Tunnels, Label Switched Paths (LSPs) and TE interfaces and covers data applicable to generic or device-independent, device-specific, and Multiprotocol Label Switching (MPLS) technology specific.

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

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

1.1. Terminology

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

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

1.2. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te	ietf-te	this document
te-dev	ietf-te-device	this document
te-types	ietf-te-types	[I-D.ietf-teas-yang-te-types]
te-mpls-types	ietf-te-mpls-types	[I-D.ietf-teas-yang-te-types]

Table 1: Prefixes and corresponding YANG modules

1.3. TE Technology Models

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

- Elements of the TE generic YANG data model, including TE tunnels, LSPs, and interfaces can be augmented with leaf(s) to identify the specific technology layer. This approach implies a single list of elements (e.g. TE tunnel(s)) in the model can carry elements of different technology layers.

- An instance of the TE generic 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.

The model defined in this document leverages the first apprach by relying on the LSP encoding type to identify the specific technology associated with a specific TE interface, tunnel or LSP. For example, for an MPLS TE LSP, the LSP encoding type is assumed to be of "te-types:lsp-encoding-packet".

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

1.4. State Data Organization

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

2. Model Overview

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

2.1. Module(s) Relationship

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

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

The TE data model for specific instances of signaling protocol are outside the scope of this document and defined in separate 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 YANG data model.

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

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

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

The TE device YANG module "ietf-te-device" imports the following module(s): - ietf-yang-types and ietf-inet-types defined in [RFC6991] - ietf-routing-types defined in [RFC8294] - ietf-te-types defined in [I-D.ietf-teas-yang-te-types] - ietf-te defined in this document

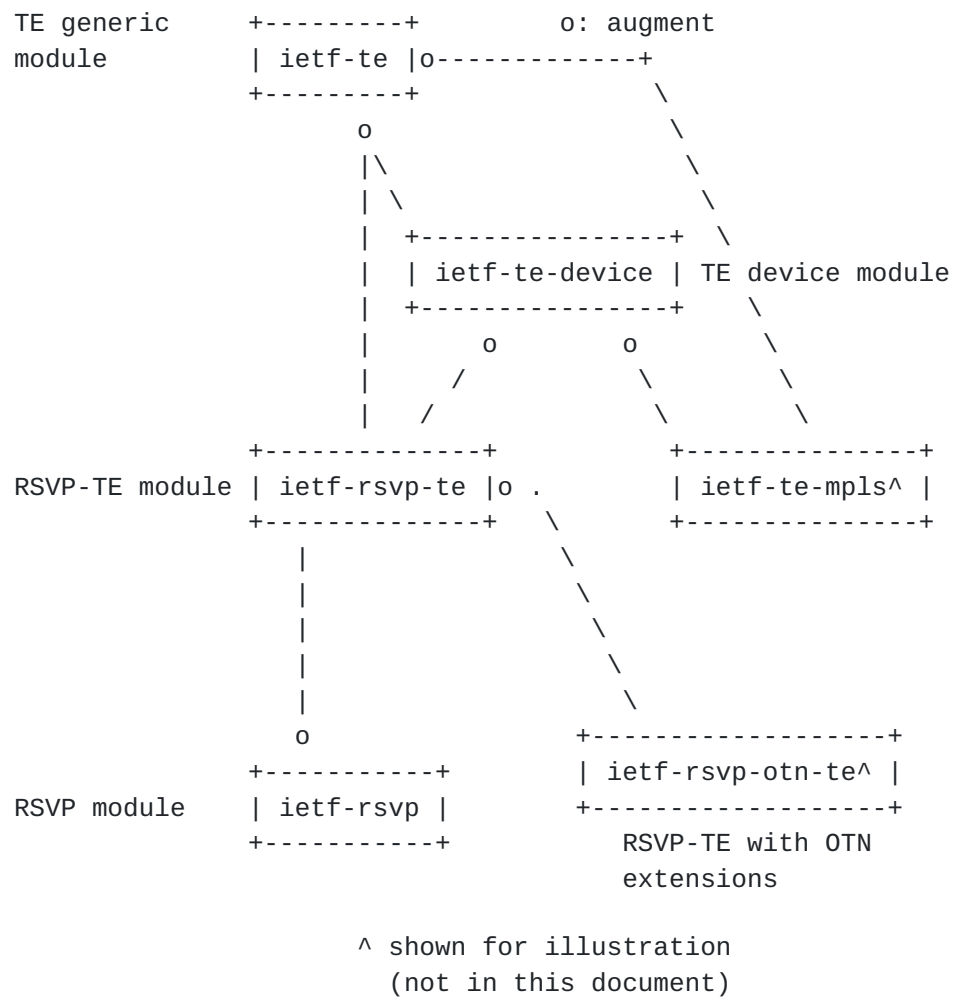
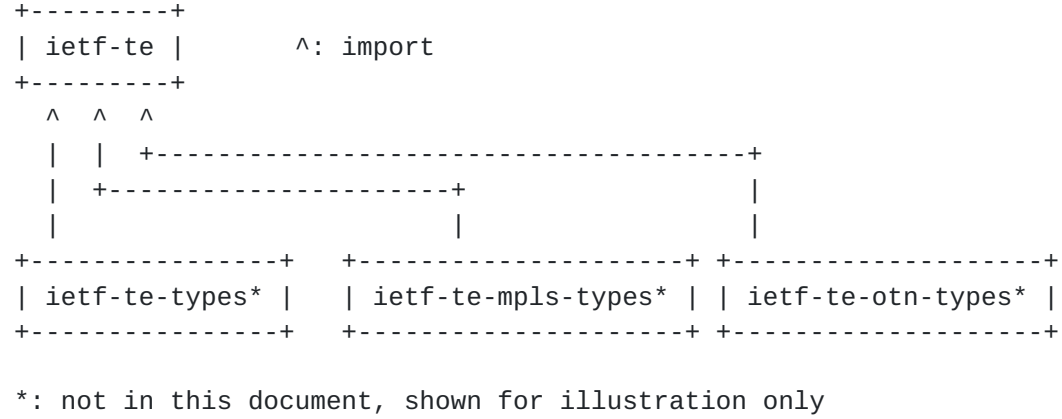


Figure 1: Relationship of TE module(s) with other signaling protocol modules



*: not in this document, shown for illustration only

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

2.2. Design Considerations

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

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

- reusable TE data types that are data plane specific (e.g. MPLS [RFC3473]) are defined in a data plane type module, e.g. "ietf-te-mpls-types.yang" as defined in [I-D.ietf-teas-yang-te-types]. Other data plane technology types are expected to be defined in separate module(s) as shown in Figure 2

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

- In general, minimal elements in the model are designated as "mandatory" to allow freedom to vendors to adapt the data model to their specific product implementation.

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

2.3. Model Tree Diagram

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

```
module: ietf-te
  +--rw te!
     +--rw globals
     |  +--rw named-admin-groups
     |  |  +--rw named-admin-group* [name]
   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 link-protection?            identityref
|  |     +--rw setup-priority?             uint8
|  |     +--rw hold-priority?              uint8
|  |     +--rw signaling-type?             identityref
|  |     +--rw path-metric-bounds
|  |     |  +--rw path-metric-bound* [metric-type]
|  |     |     +--rw metric-type    identityref
|  |     |     +--rw upper-bound?   uint64
|  |     +--rw path-affinities-values
|  |     |  +--rw path-affinities-value* [usage]
|  |     |     +--rw usage    identityref
|  |     |     +--rw value?   admin-groups
|  |     +--rw path-affinity-names
|  |     |  +--rw path-affinity-name* [usage]
|  |     |     +--rw usage            identityref
|  |     |     +--rw affinity-name* [name]
|  |     |        +--rw name    string
|  |     +--rw path-srlgs-values
|  |     |  +--rw usage?    identityref
|  |     |  +--rw values*   srlg
|  |     +--rw path-srlgs-names
|  |     |  +--rw path-srlgs-name* [usage]
|  |     |     +--rw usage        identityref
|  |     |     +--rw srlg-name* [name]
|  |     |        +--rw name    string
|  |     +--rw disjointness?
|  te-types:te-path-disjointness
|  |     +--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*   tunnel-ref
|  |     +--rw path-in-segment!
|  |     |  +--rw label-restrictions
|  |     |     +--rw label-restriction* [index]
|  |     |        +--rw restriction?    enumeration
|  |     |        +--rw index           uint32
|  |     |        +--rw label-start
|  |     |        |  +--rw te-label
|  |     |        |     +--rw (technology)?
|  |     |        |     |  +--:(generic)
|  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |     |        |     +--rw direction?       te-label-direction
|  |     |        +--rw label-end
```

```
|  |     |        |  +--rw te-label
|  |     |        |     +--rw (technology)?
|  |     |        |     |  +--:(generic)
|  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |     |        |     +--rw direction?       te-label-direction
|  |     |        +--rw label-step
|  |     |        |  +--rw (technology)?
|  |     |        |     +--:(generic)
|  |     |        |        +--rw generic?   int32
|  |     |        +--rw range-bitmap?   binary
|  |     +--rw path-out-segment!
|  |     |  +--rw label-restrictions
|  |     |     +--rw label-restriction* [index]
|  |     |        +--rw restriction?    enumeration
|  |     |        +--rw index           uint32
|  |     |        +--rw label-start
|  |     |        |  +--rw te-label
|  |     |        |     +--rw (technology)?
|  |     |        |     |  +--:(generic)
|  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |     |        |     +--rw direction?       te-label-direction
|  |     |        +--rw label-end
|  |     |        |  +--rw te-label
|  |     |        |     +--rw (technology)?
|  |     |        |     |  +--:(generic)
|  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |     |        |     +--rw direction?       te-label-direction
|  |     |        +--rw label-step
|  |     |        |  +--rw (technology)?
|  |     |        |     +--:(generic)
|  |     |        |        +--rw generic?   int32
|  |     |        +--rw range-bitmap?   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-requested-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-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?                              te-types:te-node-id
     |  |  +--rw destination?                         te-types:te-node-id
     |  |  +--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 link-protection?                   identityref
|  |  +--rw setup-priority?                    uint8
|  |  +--rw hold-priority?                     uint8
|  |  +--rw signaling-type?                    identityref
|  |  +--rw dependency-tunnels
|  |  |  +--rw dependency-tunnel* [name]
|  |  |     +--rw name              ->
../../../../../tunnels/tunnel/name
|  |  |     +--rw encoding?         identityref
|  |  |     +--rw switching-type?   identityref
|  |  +--rw hierarchical-link
|  |  |  +--rw local-te-node-id?         te-types:te-node-id
|  |  |  +--rw local-te-link-tp-id?      te-types:te-tp-id
|  |  |  +--rw remote-te-node-id?        te-types:te-node-id
|  |  |  +--rw te-topology-identifier
|  |  |     +--rw provider-id?   te-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 k-requested-paths?               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 link-protection?                 identityref
     |  |  |     +--rw setup-priority?                  uint8
     |  |  |     +--rw hold-priority?                   uint8
     |  |  |     +--rw signaling-type?                  identityref
     |  |  |     +--rw path-metric-bounds
     |  |  |     |  +--rw path-metric-bound* [metric-type]
     |  |  |     |     +--rw metric-type    identityref
     |  |  |     |     +--rw upper-bound?   uint64
     |  |  |     +--rw path-affinities-values
     |  |  |     |  +--rw path-affinities-value* [usage]
     |  |  |     |     +--rw usage    identityref
     |  |  |     |     +--rw value?   admin-groups
     |  |  |     +--rw path-affinity-names
     |  |  |     |  +--rw path-affinity-name* [usage]
     |  |  |     |     +--rw usage            identityref
     |  |  |     |     +--rw affinity-name* [name]
     |  |  |     |        +--rw name    string
     |  |  |     +--rw path-srlgs-values
```

```
|  |  |     |  +--rw usage?    identityref
|  |  |     |  +--rw values*   srlg
|  |  |     +--rw path-srlgs-names
|  |  |     |  +--rw path-srlgs-name* [usage]
|  |  |     |     +--rw usage        identityref
|  |  |     |     +--rw srlg-name* [name]
|  |  |     |        +--rw name    string
|  |  |     +--rw disjointness?
te-types:te-path-disjointness
|  |  |     +--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*   tunnel-ref
|  |  |     +--rw path-in-segment!
|  |  |     |  +--rw label-restrictions
|  |  |     |     +--rw label-restriction* [index]
|  |  |     |        +--rw restriction?    enumeration
|  |  |     |        +--rw index           uint32
|  |  |     |        +--rw label-start
|  |  |     |        |  +--rw te-label
|  |  |     |        |     +--rw (technology)?
|  |  |     |        |     |  +--:(generic)
|  |  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |  |     |        |     +--rw direction?
te-label-direction
|  |  |     |        +--rw label-end
|  |  |     |        |  +--rw te-label
|  |  |     |        |     +--rw (technology)?
|  |  |     |        |     |  +--:(generic)
|  |  |     |        |     |     +--rw generic?
rt-types:generalized-label
|  |  |     |        |     +--rw direction?
te-label-direction
|  |  |     |        +--rw label-step
|  |  |     |        |  +--rw (technology)?
|  |  |     |        |     +--:(generic)
|  |  |     |        |        +--rw generic?   int32
|  |  |     |        +--rw range-bitmap?   binary
|  |  |     +--rw path-out-segment!
|  |  |     |  +--rw label-restrictions
|  |  |     |     +--rw label-restriction* [index]
|  |  |     |        +--rw restriction?    enumeration
|  |  |     |        +--rw index           uint32
|  |  |     |        +--rw label-start
|  |  |     |        |  +--rw te-label
|  |  |     |        |     +--rw (technology)?
|  |  |     |        |     |  +--:(generic)
|  |  |     |        |     |     +--rw generic?
```

```
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-end
     |  |  |     |        |  +--rw te-label
     |  |  |     |        |     +--rw (technology)?
     |  |  |     |        |     |  +--:(generic)
     |  |  |     |        |     |     +--rw generic?
     rt-types:generalized-label
     |  |  |     |        |     +--rw direction?
     te-label-direction
     |  |  |     |        +--rw label-step
     |  |  |     |        |  +--rw (technology)?
     |  |  |     |        |     +--:(generic)
     |  |  |     |        |        +--rw generic?   int32
     |  |  |     |        +--rw range-bitmap?   binary
     |  |  |     +--ro state
     |  |  |     |  +--ro computed-paths-properties
     |  |  |     |  |  +--ro computed-path-properties* [k-index]
     |  |  |     |  |     +--ro k-index            uint8
     |  |  |     |  |     +--ro path-properties
     |  |  |     |  |        +--ro path-metric* [metric-type]
     |  |  |     |  |        |  +--ro metric-type    ->
     ../state/metric-type
     |  |  |     |  |        |  +--ro state
     |  |  |     |  |        |     +--ro metric-type?
     identityref
     |  |  |     |  |        |     +--ro accumulative-value?   uint64
     |  |  |     |  |        +--ro path-affinities-values
     |  |  |     |  |        |  +--ro path-affinities-value* [usage]
     |  |  |     |  |        |     +--ro usage    identityref
     |  |  |     |  |        |     +--ro value?   admin-groups
     |  |  |     |  |        +--ro path-affinity-names
     |  |  |     |  |        |  +--ro path-affinity-name* [usage]
     |  |  |     |  |        |     +--ro usage            identityref
     |  |  |     |  |        |     +--ro affinity-name* [name]
     |  |  |     |  |        |        +--ro name    string
     |  |  |     |  |        +--ro path-srlgs-values
     |  |  |     |  |        |  +--ro usage?    identityref
     |  |  |     |  |        |  +--ro values*   srlg
     |  |  |     |  |        +--ro path-srlgs-names
     |  |  |     |  |        |  +--ro path-srlgs-name* [usage]
     |  |  |     |  |        |     +--ro usage        identityref
     |  |  |     |  |        |     +--ro srlg-name* [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*
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?
     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-values
|  |  |     |  |     |  |  +--ro path-affinities-value* [usage]
|  |  |     |  |     |  |     +--ro usage    identityref
|  |  |     |  |     |  |     +--ro value?   admin-groups
|  |  |     |  |     |  +--ro path-affinity-names
|  |  |     |  |     |  |  +--ro path-affinity-name* [usage]
|  |  |     |  |     |  |     +--ro usage            identityref
|  |  |     |  |     |  |     +--ro affinity-name* [name]
|  |  |     |  |     |  |        +--ro name    string
|  |  |     |  |     |  +--ro path-srlgs-values
|  |  |     |  |     |  |  +--ro usage?    identityref
|  |  |     |  |     |  |  +--ro values*   srlg
|  |  |     |  |     |  +--ro path-srlgs-names
|  |  |     |  |     |  |  +--ro path-srlgs-name* [usage]
|  |  |     |  |     |  |     +--ro usage        identityref
|  |  |     |  |     |  |     +--ro srlg-name* [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*
     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:performance-metric-one-way
     |  |  |     |  |     |  +--ro te-mpls:one-way-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-min-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-max-delay?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-delay-variation?
     uint32
     |  |  |     |  |     |  +--ro te-mpls:one-way-packet-loss?
     decimal64
     |  |  |     |  |     |  +--ro te-mpls:one-way-residual-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     |  +--ro te-mpls:one-way-available-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     |  +--ro te-mpls:one-way-utilized-bandwidth?
     rt-types:bandwidth-ieee-float32
     |  |  |     |  |     +--ro te-mpls:performance-metric-two-way
     |  |  |     |  |        +--ro te-mpls:two-way-delay?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-min-delay?
     uint32
     |  |  |     |  |        +--ro te-mpls:two-way-max-delay?
     uint32
```

```
|  |  |     |  |        +--ro te-mpls:two-way-delay-variation?
uint32
|  |  |     |  |        +--ro te-mpls:two-way-packet-loss?
decimal64
|  |  |     |  +--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 link-protection?             identityref
     |  |  |     |  +--rw setup-priority?              uint8
     |  |  |     |  +--rw hold-priority?               uint8
     |  |  |     |  +--rw signaling-type?              identityref
     |  |  |     |  +--rw path-metric-bounds
     |  |  |     |  |  +--rw path-metric-bound* [metric-type]
     |  |  |     |  |     +--rw metric-type    identityref
     |  |  |     |  |     +--rw upper-bound?   uint64
     |  |  |     |  +--rw path-affinities-values
     |  |  |     |  |  +--rw path-affinities-value* [usage]
     |  |  |     |  |     +--rw usage    identityref
     |  |  |     |  |     +--rw value?   admin-groups
     |  |  |     |  +--rw path-affinity-names
     |  |  |     |  |  +--rw path-affinity-name* [usage]
     |  |  |     |  |     +--rw usage            identityref
     |  |  |     |  |     +--rw affinity-name* [name]
     |  |  |     |  |        +--rw name    string
     |  |  |     |  +--rw path-srlgs-values
     |  |  |     |  |  +--rw usage?    identityref
     |  |  |     |  |  +--rw values*   srlg
     |  |  |     |  +--rw path-srlgs-names
     |  |  |     |  |  +--rw path-srlgs-name* [usage]
     |  |  |     |  |     +--rw usage        identityref
     |  |  |     |  |     +--rw srlg-name* [name]
     |  |  |     |  |        +--rw name    string
     |  |  |     |  +--rw disjointness?
     te-types:te-path-disjointness
     |  |  |     |  +--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*   tunnel-ref
      |  |  |     |  +--rw path-in-segment!
      |  |  |     |  |  +--rw label-restrictions
      |  |  |     |  |     +--rw label-restriction* [index]
      |  |  |     |  |        +--rw restriction?    enumeration
      |  |  |     |  |        +--rw index           uint32
      |  |  |     |  |        +--rw label-start
      |  |  |     |  |        |  +--rw te-label
      |  |  |     |  |        |     +--rw (technology)?
      |  |  |     |  |        |     |  +--:(generic)
      |  |  |     |  |        |     |     +--rw generic?
      rt-types:generalized-label
```

```
|  |  |     |  |        |     +--rw direction?
te-label-direction
|  |  |     |  |        +--rw label-end
|  |  |     |  |        |  +--rw te-label
|  |  |     |  |        |     +--rw (technology)?
|  |  |     |  |        |     |  +--:(generic)
|  |  |     |  |        |     |     +--rw generic?
rt-types:generalized-label
|  |  |     |  |        |     +--rw direction?
te-label-direction
|  |  |     |  |        +--rw label-step
|  |  |     |  |        |  +--rw (technology)?
|  |  |     |  |        |     +--:(generic)
|  |  |     |  |        |        +--rw generic?   int32
|  |  |     |  |        +--rw range-bitmap?   binary
|  |  |     |  +--rw path-out-segment!
|  |  |     |  |  +--rw label-restrictions
|  |  |     |  |     +--rw label-restriction* [index]
|  |  |     |  |        +--rw restriction?    enumeration
|  |  |     |  |        +--rw index           uint32
|  |  |     |  |        +--rw label-start
|  |  |     |  |        |  +--rw te-label
|  |  |     |  |        |     +--rw (technology)?
|  |  |     |  |        |     |  +--:(generic)
|  |  |     |  |        |     |     +--rw generic?
rt-types:generalized-label
|  |  |     |  |        |     +--rw direction?
te-label-direction
|  |  |     |  |        +--rw label-end
|  |  |     |  |        |  +--rw te-label
|  |  |     |  |        |     +--rw (technology)?
|  |  |     |  |        |     |  +--:(generic)
|  |  |     |  |        |     |     +--rw generic?
rt-types:generalized-label
|  |  |     |  |        |     +--rw direction?
te-label-direction
|  |  |     |  |        +--rw label-step
|  |  |     |  |        |  +--rw (technology)?
|  |  |     |  |        |     +--:(generic)
|  |  |     |  |        |        +--rw generic?   int32
|  |  |     |  |        +--rw range-bitmap?   binary
|  |  |     |  +--ro state
|  |  |     |  |  +--ro computed-paths-properties
|  |  |     |  |  |  +--ro computed-path-properties* [k-index]
|  |  |     |  |  |     +--ro k-index            uint8
|  |  |     |  |  |     +--ro path-properties
|  |  |     |  |  |        +--ro path-metric* [metric-type]
|  |  |     |  |  |        |  +--ro metric-type    ->
```

```
../state/metric-type
|  |  |     |  |  |        |  +--ro state
|  |  |     |  |  |        |     +--ro metric-type?
identityref
|  |  |     |  |  |        |     +--ro accumulative-value?   uint64
|  |  |     |  |  |        +--ro path-affinities-values
|  |  |     |  |  |        |  +--ro path-affinities-value* [usage]
|  |  |     |  |  |        |     +--ro usage    identityref
|  |  |     |  |  |        |     +--ro value?   admin-groups
|  |  |     |  |  |        +--ro path-affinity-names
|  |  |     |  |  |        |  +--ro path-affinity-name* [usage]
|  |  |     |  |  |        |     +--ro usage            identityref
|  |  |     |  |  |        |     +--ro affinity-name* [name]
|  |  |     |  |  |        |        +--ro name    string
|  |  |     |  |  |        +--ro path-srlgs-values
|  |  |     |  |  |        |  +--ro usage?    identityref
|  |  |     |  |  |        |  +--ro values*   srlg
|  |  |     |  |  |        +--ro path-srlgs-names
|  |  |     |  |  |        |  +--ro path-srlgs-name* [usage]
|  |  |     |  |  |        |     +--ro usage        identityref
|  |  |     |  |  |        |     +--ro srlg-name* [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*
     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?
     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-values
|  |  |     |  |          |  +--ro path-affinities-value* [usage]
|  |  |     |  |          |     +--ro usage    identityref
|  |  |     |  |          |     +--ro value?   admin-groups
|  |  |     |  |          +--ro path-affinity-names
|  |  |     |  |          |  +--ro path-affinity-name* [usage]
|  |  |     |  |          |     +--ro usage            identityref
|  |  |     |  |          |     +--ro affinity-name* [name]
|  |  |     |  |          |        +--ro name    string
|  |  |     |  |          +--ro path-srlgs-values
|  |  |     |  |          |  +--ro usage?    identityref
|  |  |     |  |          |  +--ro values*   srlg
|  |  |     |  |          +--ro path-srlgs-names
|  |  |     |  |          |  +--ro path-srlgs-name* [usage]
|  |  |     |  |          |     +--ro usage        identityref
|  |  |     |  |          |     +--ro srlg-name* [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*
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 k-requested-paths?         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 link-protection?           identityref
     |  |  |     +--rw setup-priority?            uint8
```

```
      |  |  |     +--rw hold-priority?              uint8
      |  |  |     +--rw signaling-type?             identityref
      |  |  |     +--rw path-metric-bounds
      |  |  |     |  +--rw path-metric-bound* [metric-type]
      |  |  |     |     +--rw metric-type    identityref
      |  |  |     |     +--rw upper-bound?   uint64
      |  |  |     +--rw path-affinities-values
      |  |  |     |  +--rw path-affinities-value* [usage]
      |  |  |     |     +--rw usage    identityref
      |  |  |     |     +--rw value?   admin-groups
      |  |  |     +--rw path-affinity-names
      |  |  |     |  +--rw path-affinity-name* [usage]
      |  |  |     |     +--rw usage            identityref
      |  |  |     |     +--rw affinity-name* [name]
      |  |  |     |        +--rw name    string
      |  |  |     +--rw path-srlgs-values
      |  |  |     |  +--rw usage?    identityref
      |  |  |     |  +--rw values*   srlg
      |  |  |     +--rw path-srlgs-names
      |  |  |     |  +--rw path-srlgs-name* [usage]
      |  |  |     |     +--rw usage        identityref
      |  |  |     |     +--rw srlg-name* [name]
      |  |  |     |        +--rw name    string
      |  |  |     +--rw disjointness?
     te-types:te-path-disjointness
      |  |  |     +--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*   tunnel-ref
      |  |  |     +--rw path-in-segment!
      |  |  |     |  +--rw label-restrictions
      |  |  |     |     +--rw label-restriction* [index]
      |  |  |     |        +--rw restriction?    enumeration
      |  |  |     |        +--rw index           uint32
      |  |  |     |        +--rw label-start
      |  |  |     |        |  +--rw te-label
      |  |  |     |        |     +--rw (technology)?
      |  |  |     |        |     |  +--:(generic)
      |  |  |     |        |     |     +--rw generic?
      rt-types:generalized-label
      |  |  |     |        |     +--rw direction?
      te-label-direction
      |  |  |     |        +--rw label-end
      |  |  |     |        |  +--rw te-label
      |  |  |     |        |     +--rw (technology)?
      |  |  |     |        |     |  +--:(generic)
      |  |  |     |        |     |     +--rw generic?
      rt-types:generalized-label
      |  |  |     |        |     +--rw direction?
```

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

```
|  |  |     |  +--ro computed-paths-properties
|  |  |     |  |  +--ro computed-path-properties* [k-index]
|  |  |     |  |     +--ro k-index            uint8
|  |  |     |  |     +--ro path-properties
|  |  |     |  |        +--ro path-metric* [metric-type]
|  |  |     |  |        |  +--ro metric-type    ->
../state/metric-type
|  |  |     |  |        |  +--ro state
|  |  |     |  |        |     +--ro metric-type?
identityref
|  |  |     |  |        |     +--ro accumulative-value?   uint64
|  |  |     |  |        +--ro path-affinities-values
|  |  |     |  |        |  +--ro path-affinities-value* [usage]
|  |  |     |  |        |     +--ro usage    identityref
|  |  |     |  |        |     +--ro value?   admin-groups
|  |  |     |  |        +--ro path-affinity-names
|  |  |     |  |        |  +--ro path-affinity-name* [usage]
|  |  |     |  |        |     +--ro usage            identityref
|  |  |     |  |        |     +--ro affinity-name* [name]
|  |  |     |  |        |        +--ro name    string
|  |  |     |  |        +--ro path-srlgs-values
|  |  |     |  |        |  +--ro usage?    identityref
|  |  |     |  |        |  +--ro values*   srlg
|  |  |     |  |        +--ro path-srlgs-names
|  |  |     |  |        |  +--ro path-srlgs-name* [usage]
|  |  |     |  |        |     +--ro usage        identityref
|  |  |     |  |        |     +--ro srlg-name* [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*
     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?
      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-values
      |  |  |     |  |     |  |  +--ro path-affinities-value* [usage]
      |  |  |     |  |     |  |     +--ro usage    identityref
      |  |  |     |  |     |  |     +--ro value?   admin-groups
      |  |  |     |  |     |  +--ro path-affinity-names
      |  |  |     |  |     |  |  +--ro path-affinity-name* [usage]
      |  |  |     |  |     |  |     +--ro usage            identityref
      |  |  |     |  |     |  |     +--ro affinity-name* [name]
      |  |  |     |  |     |  |        +--ro name    string
      |  |  |     |  |     |  +--ro path-srlgs-values
      |  |  |     |  |     |  |  +--ro usage?    identityref
      |  |  |     |  |     |  |  +--ro values*   srlg
      |  |  |     |  |     |  +--ro path-srlgs-names
      |  |  |     |  |     |  |  +--ro path-srlgs-name* [usage]
      |  |  |     |  |     |  |     +--ro usage        identityref
```

```
|  |  |     |  |     |  |     +--ro srlg-name* [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*
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?           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-requested-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?   rt-types:percentage
|  |        +--rw te-mpls:overflow
|  |        |  +--rw te-mpls:enabled?              boolean
|  |        |  +--rw te-mpls:overflow-threshold?
rt-types:percentage
|  |        |  +--rw te-mpls:trigger-event-count?   uint16
|  |        +--rw te-mpls:underflow
|  |           +--rw te-mpls:enabled?              boolean
|  |           +--rw te-mpls:underflow-threshold?
rt-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?          rt-types:percentage
   +--rw te-dev:threshold-specification?   enumeration
   +--rw te-dev:up-thresholds*             rt-types:percentage
   +--rw te-dev:down-thresholds*           rt-types:percentage
   +--rw te-dev:up-down-thresholds*        rt-types:percentage
   +--rw te-dev:interface* [interface]
      +--rw te-dev:interface
      if:interface-ref
      +--rw te-dev:te-metric?
      te-types:te-metric
      +--rw (te-dev:admin-group-type)?
      |  +--:(te-dev:value-admin-groups)
      |  |  +--rw (te-dev:value-admin-group-type)?
      |  |     +--:(te-dev:admin-groups)
      |  |     |  +--rw te-dev:admin-group?
      te-types:admin-group
      |  |     +--:(te-dev:extended-admin-groups)
      {te-types:extended-admin-groups}?
      |  |        +--rw te-dev:extended-admin-group?
      te-types:extended-admin-group
```

```
           |  +--:(te-dev:named-admin-groups)
           |     +--rw te-dev:named-admin-groups* [named-admin-group]
{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?
           rt-types:percentage
           +--rw te-dev:threshold-specification?            enumeration
           +--rw te-dev:up-thresholds*
           rt-types:percentage
           +--rw te-dev:down-thresholds*
           rt-types:percentage
           +--rw te-dev:up-down-thresholds*
           rt-types:percentage
           +--rw te-dev:switching-capabilities* [switching-capability]
           |  +--rw te-dev:switching-capability    identityref
           |  +--rw te-dev:encoding?               identityref
           +--ro te-dev:state
              +--ro te-dev:te-advertisements_state
                 +--ro te-dev:flood-interval?           uint32
                 +--ro te-dev:last-flooded-time?        uint32
                 +--ro te-dev:next-flooded-time?        uint32
                 +--ro te-dev:last-flooded-trigger?     enumeration
                 +--ro te-dev:advertized-level-areas* [level-area]
                    +--ro te-dev:level-area    uint32

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

```
          +--ro result
             +--ro result?   enumeration

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

  rpcs:
    +---x interfaces-rpc

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

Figure 3: TE generic model configuration and state tree

3. Model Organization

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

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

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

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

      +--rw tunnels
         .
         .

      +-- lsps-state

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

Figure 4: TE generic highlevel model view

3.1. Global Configuration and State Data

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

- 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

3.2. Interfaces Configuration and State Data

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

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

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 5 below:

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

Figure 5: TE interface state

This covers state data for TE interfaces such as:

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

3.3. Tunnels Configuration and State Data

This branch covers data related to TE tunnels configuration and state. Data that is device independent is defined in the TE generic YANG module "ietf-te", where as the device dependent data is defined in the device module "ietf-te-device". The derived state associated with tunnels is grouped under a state container as shown in Figure 6.

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

Figure 6: TE interface state tree

Examples of tunnel configuration data for TE tunnels:

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

3.3.1. Tunnel Compute-Only Mode

A configured TE tunnel, by default, is provisioned so it can carry traffic as soon as a valid path is computed and an LSP instantiated. In some cases, however, a TE tunnel may be provisioned for the only purpose of computing a path and reporting it without the need to instantiate the LSP or commit any resources. In such a case, the tunnel is configured in "compute-only" mode to distinguish it from default tunnel behavior.

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

3.3.2. Tunnel Hierarchical Link Endpoint

TE LSPs can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used to form links to carry traffic in in other (client) networks [RFC6107]. In this case, the model introduces the TE tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the 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 3.

3.5. Global RPC Data

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

- Clear global TE statistics of various features

3.6. Interface RPC Data

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

- Clear TE statistics for all or for individual TE interfaces

- Trigger immediate flooding for one or all TE interfaces

3.7. Tunnel RPC Data

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

4. TE Generic and Helper YANG Modules

```
<CODE BEGINS> file "ietf-te@2018-10-10.yang"
module ietf-te {
  yang-version 1.1;
```

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

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

  /* Import TE generic types */
  import ietf-te-types {
    prefix te-types;
    reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
               Common Traffic Engineering Types";
  }

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

  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 "2018-10-10" {
    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 tunnel-p2mp-ref {
    type leafref {
      path "/te:te/te:tunnels/te:tunnel-p2mp/te:name";
    }
    description
      "This type is used by data models that need to reference
       configured P2MP TE tunnel.";
    reference "RFC4875";
  }

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

  /**
   * TE tunnel generic groupings
   */
  grouping p2p-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 te-types:generic-path-affinities;
      uses te-types:generic-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";
            leaf index {
              type uint32;
              description "ERO subobject index";
            }
            uses te-types:explicit-route-hop;
          }
        }
      }
      uses shared-resources-tunnels;
    }
  }

  grouping p2p-path-properties_state {
    description "TE per path state parameters";
    container computed-paths-properties {
      description "Computed path properties container";
      list computed-path-properties {
        key k-index;
        description "List of computed paths";
        leaf k-index {
          type uint8;
          description
            "The k-th path returned from the computation server.";
        }
        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 k-requested-paths {
      type uint8;
      description
        "The number of k-shortest-paths  requested from the path
         computation server and returned sorted by its optimization
         objective";
    }
    leaf named-path-constraint {
      if-feature te-types:named-path-constraints;
      type leafref {
        path "../../../../../globals/"
        + "named-path-constraints/named-path-constraint/"
        + "name";
      }
      description
        "Reference to a globally defined named path
        constraint set";
    }
  }

  /* 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 path-access-segment-info {
    description
      "If an end-to-end tunnel crosses multiple domains using
       the same technology, some additional constraints have to be
       taken in consideration in each domain";
    container path-in-segment {
      presence
        "The end-to-end tunnel starts in a previous domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
    container path-out-segment {
      presence
        "The end-to-end tunnel is not terminated in this domain;
         this tunnel is a segment in the current domain.";
      description
        "This tunnel is a segment that needs to be coordinated
```

```
         with previous segment stitched on head-end side.";
      uses te-types:label-set-info;
    }
  }

  /* TE tunnel configuration/state grouping */
  grouping tunnel-p2mp-properties {
    description
      "Top level grouping for P2MP tunnel properties.";
    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 tunnel-ref;
      description
        "Reference to the tunnel that sharing secondary path
        resources with this tunnel";
    }
  }
  grouping shared-resources-tunnels {
    description
      "Set of tunnels that share secondary path resources with
      this tunnnel";
    container shared-resources-tunnels {
      description
        "Set of tunnels that share secondary path resources with
        this tunnnel";
      leaf-list lsp-shared-resources-tunnel {
        type tunnel-ref;
        description
          "Reference to the tunnel that sharing secondary path
          resources with this tunnel";
      }
    }
  }

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

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

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

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

  /**
   * LSP related generic groupings
   */
  grouping lsp-record-route-information_state {
    description "recorded route information grouping";
    container lsp-record-route-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 tunnel-ref;
              description "P2P TE tunnel";
            }
```

```
          }
          case tunnel-p2mp {
            leaf p2mp-id {
              type tunnel-p2mp-ref;
              description "P2MP TE tunnel";
            }
          }
        }
      }
    }
    output {
      container result {
        description
          "The container result of the RPC operation";
        leaf result {
          type enumeration {
            enum success {
              description "Origin ingress";
            }
            enum in-progress {
              description "Origin egress";
            }
            enum fail {
              description "transit";
            }
          }
          description "The result of the RPC operation";
        }
      }
    }
  }

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

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

Figure 7: TE generic YANG module

```
   <CODE BEGINS> file "ietf-te-device@2018-10-10.yang"
   module ietf-te-device {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

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

     /* Import TE generic types */
     import ietf-te {
       prefix te;
       reference "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
                  Engineering Tunnels and Interfaces";
     }

     /* Import TE generic types */
     import ietf-te-types {
       prefix te-types;
       reference "draft-ietf-teas-yang-te-types: A YANG Data Model for
                  Common Traffic Engineering Types";
     }

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

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

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

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

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

        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 "2018-10-10" {
       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 rt-types:percentage;
          description
            "The percentage of the maximum-reservable-bandwidth
            considered as the delta that results in an IGP update
            being flooded";
        }
        leaf threshold-specification {
          when "../threshold-type = 'THRESHOLD_CROSSED'" {
```

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

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

```
      }

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

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

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

    /* TE interface switching capabilities */
    grouping te-switching-cap_config {
      description
```

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

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

```
              description "Link-up flooding trigger";
            }
            enum link-down {
              description "Link-up flooding trigger";
            }
            enum threshold-up {
              description
                "Bandwidth reservation up threshold";
            }
            enum threshold-down {
              description
                "Bandwidth reservation down threshold";
            }
            enum bandwidth-change {
              description "Banwidth capacity change";
            }
            enum user-initiated {
              description "Initiated by user";
            }
            enum srlg-change {
              description "SRLG property change";
            }
            enum periodic-timer {
              description "Periodic timer expired";
            }
          }
          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 8: TE device specific YANG module

5. IANA Considerations

This document registers the following URIs in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following 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.

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

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-path-computation]
Busi, I., Belotti, S., Lopezalvarez, V., Dios, O., Sharma, A., Shi, Y., Vilata, R., and K. Sethuraman, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-02 (work in progress), June 2018.

[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.

[I-D.ietf-teas-yang-te-types]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types-01 (work in progress), October 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>.

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

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

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

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

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

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

Authors' Addresses

Tarek Saad
Cisco Systems Inc

Email: tsaad@cisco.com

Rakesh Gandhi
Cisco Systems Inc

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Himanshu Shah
Ciena

Email: hshah@ciena.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com