Network Working Group I. Bryskin
Internet-Draft Huawei Technologies
Intended status: Informational X. Liu
Expires: September 2, 2018 Jabil
March 1, 2018

SF Aware TE Topology YANG Model
draft-bryskin-teas-sf-aware-topo-model-01

Abstract

This document describes a YANG data model for TE network topologies that are network service and function aware.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Today a network offers to its clients far more services than just connectivity across the network. Large variety of physical, logical and/or virtual service functions, network functions and transport functions (collectively named in this document as SFs) could be allocated for and assigned to a client. As described in [I-D.bryskin-teas-use-cases-sf-aware-topo-model], there are some important use cases, in which the network needs to represent to the client SFs at the client's disposal as topological elements in relation to other elements of a topology (i.e. nodes, links, link and tunnel termination points) used by the network to describe itself to the client. Not only would such information allow for the client to auto-discover the network's SFs available for the services provisioned for the client, it would also allow for the client selecting the SFs, duel-optimizing the selection on the SF location on the network and connectivity means (e.g. TE tunnels) to inter-connect the SFs. Consequently thus would give to both the network and the client powerful means for the service function chain (SFC [RFC7498] [RFC7665]) negotiation to achieve most efficient and cost effective (from the network point of view) and most optimal yet satisfying all necessary constraints of SFCs(from the client's point of view).

This document defines a YANG data model that allows service functions to be represented along with TE topology elements.

1.1. Terminology

The keywords "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].

The following terms are defined in [RFC7950] and are not redefined here:

- augment
- data model
- data node

1.2. Tree Diagrams

A simplified graphical representation of the data model is presented in this document, by using the tree format defined in [I-D.ietf-netmod-yang-tree-diagrams].

1.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[I-D.ietf-i2rs-yang-network-topo]
nt	ietf-network-topology	[I-D.ietf-i2rs-yang-network-topo]
tet	ietf-te-topology	[I-D.ietf-teas-yang-te-topo]

Table 1: Prefixes and Corresponding YANG Modules

2. Modeling Considerations

The model introduced in this document is an augmentation of the TE Topology model defined in [I-D.ietf-teas-yang-te-topo]. SFs are modeled as child elements of a TE node similarly to how Link Termination Points (LTPs) and Tunnel Termination Points (TTPs) are modeled in the TE Topology model. The SFs are defined as opaque objects identified via topology unique service-function-id’s. Each SF has one or more Connection Points (CPs) identified via SF-unique sf-connection-point-id’s, over which the SF could be connected to other SFs resided on the same TE node, as well as to other elements of the TE node, in particular, to the node’s LTPs and/or TTPs. An interested client may use service-function-id’s to look up the SFs in TOSCA or YANG data store(s) defined by [ETSI-NFV-MAN] to retrieve the details of the SFs, for example, to understand the SF’s mutual substitutability.

The TE Topology model introduces a concept of Connectivity Matrix (CM), and uses the CM to describe which and at what costs a TE node’s LTPs could be inter-connected internally across the TE node. The model defined in this document heavily uses the same concept to describe the SF connectivity via introducing 3 additional CMs:

1. SF2SF CM. This CM describes which pairs of SFs could be locally inter-connected, and, if yes, in which direction, via which CPs and at what costs. In other words, the SF2SF CM describes how SFs residing on the same TE node could be inter-connected into local from the TE node’s perspective SFCs;

2. SF2LTP CM. This CM describes how, in which direction and at what costs the TE node’s SFs could be connected to the TE node’s LTPs and hence to SFs residing on neighboring TE nodes that are connected to LTPs at the remote ends of corresponding TE links;

3. SF2TTP CM. This CM describes how, in which direction and at what costs the TE node’s SFs could be connected to the TE node’s TTPs and hence to SFs residing on other TE nodes on the topology that could be inter-connected with the TE node in question via TE tunnels terminated by the corresponding TTPs.

In addition to SF2SF CM, the local SF chaining could be described with the help of ETSI models Virtual Links (VLs) [ETSI-NFV-MAN]. This option is especially useful when the costs of the local chaining are negligible as compared to ones of the end-to-end SFCs said local SFCs are part of.

3. Model Structure

```
   module: ietf-te-topology-sf
     augment /nw:networks/nw:network/nw:network-types/tet:te-topology:
       +--rw sf!
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:te-node-attributes:
       +--rw service-function
```

```
          +--rw connectivity-matrices
          |  +--rw connectivity-matrix* [id]
          |     +--rw id                  uint32
          |     +--rw from
          |     |  +--rw service-function-id?      string
          |     |  +--rw sf-connection-point-id?   string
          |     +--rw to
          |     |  +--rw service-function-id?      string
          |     |  +--rw sf-connection-point-id?   string
          |     +--rw enabled?            boolean
          |     +--rw direction?          connectivity-direction
          |     +--rw virtual-link-id?    string
          +--rw link-terminations
             +--rw link-termination* [id]
                +--rw id           uint32
                +--rw from
                |  +--rw tp-ref?   -> ../../../../../../..
   /nt:termination-point/tp-id
                +--rw to
                |  +--rw service-function-id?      string
                |  +--rw sf-connection-point-id?   string
                +--rw enabled?     boolean
                +--rw direction?   connectivity-direction
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:information-source-entry:
       +--ro service-function
          +--ro connectivity-matrices
          |  +--ro connectivity-matrix* [id]
          |     +--ro id                  uint32
          |     +--ro from
          |     |  +--ro service-function-id?      string
          |     |  +--ro sf-connection-point-id?   string
          |     +--ro to
          |     |  +--ro service-function-id?      string
          |     |  +--ro sf-connection-point-id?   string
          |     +--ro enabled?            boolean
          |     +--ro direction?          connectivity-direction
          |     +--ro virtual-link-id?    string
          +--ro link-terminations
             +--ro link-termination* [id]
                +--ro id           uint32
                +--ro from
                +--ro to
                |  +--ro service-function-id?      string
                |  +--ro sf-connection-point-id?   string
                +--ro enabled?     boolean
                +--ro direction?   connectivity-direction
     augment /nw:networks/nw:network/nw:node/tet:te
   /tet:tunnel-termination-point:
       +--rw service-function
          +--rw tunnel-terminations
             +--rw tunnel-termination* [id]
                +--rw id                         uint32
                +--rw service-function-id?       string
                +--rw sf-connection-point-id?    string
                +--rw enabled?                   boolean
                +--rw direction?                 connectivity-direction
```

4. YANG Modules

```
   <CODE BEGINS> file "ietf-te-topology-sf@2018-02-27.yang"
   module ietf-te-topology-sf {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-sf";

     prefix "tet-sf";

     import ietf-network {
       prefix "nw";
     }

     import ietf-network-topology {
       prefix "nt";
     }

     import ietf-te-topology {
       prefix "tet";
     }

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

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

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

                  Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>";

     description
```

```
       "Network service and function aware aware TE topology model.

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

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

     revision 2018-02-27 {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Typedefs
      */
     typedef connectivity-direction {
       type enumeration {
         enum "to" {
           description
           "The direction is uni-directional, towards the ’to’
            entity direction.";
         }
         enum "from" {
           description
           "The direction is uni-directional, from the ’to’
            entity direction.";
         }
         enum "bidir" {
           description
           "The direction is bi-directional.";
         }
       }
       description
         "A type used to indicates whether a connectivity is
          uni-directional, or bi-directional. If the relation is
          uni-directional, the value of this type indicates the
          direction.";
     } // connectivity-direction

     /*
      * Groupings
      */
     grouping service-function-node-augmentation {
       description
         "Augmenting a TE node to be network service and function
          aware.";
       container service-function {
         description
           "Containing attributes related to network services and
            network functions";
         container connectivity-matrices {
           description
             "Connectivity relations between network services/functions
              on a TE node, which can be either abstract or physical.";
           reference
             "ETSI GS NFV-MAN 01: Network Functions Virtualisation
              (NFV); Management and Orchestration.
              RFC7665: Service Function Chaining (SFC) Architecture.";
           list connectivity-matrix {
             key "id";
             description
               "Represents the connectivity relations between network
                services/functions on a TE node.";
             leaf id {
               type uint32;
               description "Identifies the connectivity-matrix entry.";
             }

             container from {
               description
                 "Reference to the source network service or
                  network function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
               }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // from
             container to {
               description
                 "Reference to the destination network service or
                  network function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
                 }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // to
             leaf enabled {
               type boolean;
               description
                 "’true’ if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
                 "Indicates whether this connectivity is
                  uni-directional, or bi-directional. If the
                  relation is uni-directional, the value of
                  this leaf indicates the direction.";
             }
             leaf virtual-link-id {
               type string;
               description
                 "Reference to a virtual link that models this
                  conectivity relation in the network function
                  model.";
             }
           } // connectivity-matrix
         } // connectivity-matrices

         container link-terminations {
           description
             "Connectivity relations between network services/functions
              and link termination points on a TE node, which can be
              either abstract or physical.";
           reference
             "ETSI GS NFV-MAN 01: Network Functions Virtualisation
              (NFV); Management and Orchestration.
              RFC7665: Service Function Chaining (SFC) Architecture.";
           list link-termination {
             key "id";
             description
               "Each entry of the list represents the connectivity
                relation between a network service/function and
                a link termination point on a TE node.";
             leaf id {
               type uint32;
               description "Identifies the termination entry.";
             }

             container from {
               description
                 "Reference to the link termination point.";
             } // from
             container to {
               description
                 "Reference to the network service or network
                  function.";
               leaf service-function-id {
                 type string;
                 description
                   "Reference to a network service or a network
                    function.";
                 }
               leaf sf-connection-point-id {
                 type string;
                 description
                   "Reference to a connection point on a network
                    service or a network function.";
               }
             } // to
             leaf enabled {
               type boolean;
               description
                 "’true’ if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
                 "Indicates whether this connectivity is
                  uni-directional, or bi-directional. If the
                  relation is uni-directional, the value of
                  this leaf indicates the direction.";
             }
           } // link-termination
         }
       }
     } // service-function-node-augmentation

     grouping service-function-ttp-augmentation {
       description
         "Augmenting a tunnel termination point to be network service
```

```
          aware.";
       container service-function {
         description
           "Containing attributes related to network services and
            network functions";
         container tunnel-terminations {
           description
             "Connectivity relations between network services/functions
              and tunnel termination points on a TE node, which can be
              either abstract or physical.";
           reference
             "ETSI GS NFV-MAN 01: Network Functions Virtualisation
              (NFV); Management and Orchestration.
              RFC7665: Service Function Chaining (SFC) Architecture.";
           list tunnel-termination {
             key "id";
             description
               "Each entry of the list represents the connectivity
                relation between a network service/function and
                a tunnel termination point on a TE node.";
             leaf id {
               type uint32;
               description "Identifies the termination entry.";
             }

             leaf service-function-id {
               type string;
               description
                 "Reference to a network service or a network
                  function.";
             }
             leaf sf-connection-point-id {
               type string;
               description
                 "Reference to a connection point on a network
                  service or a network function.";
             }
             leaf enabled {
               type boolean;
               description
                 "'true' if this connectivity entry is enabled.";
             }
             leaf direction {
               type connectivity-direction;
               description
                 "Indicates whether this connectivity is
                  uni-directional, or bi-directional. If the
                  relation is uni-directional, the value of
                  this leaf indicates the direction.";
             }
           } // link-termination
         }
       }
     } // service-function-ttp-augmentation

     grouping sf-topology-type {
       description
         "Identifies the SF aware TE topology type.";
       container sf {
         presence "Indidates that the TE topology is SF aware.";
         description
           "Its presence identifies that the TE topology is SF aware.";
       }
     } // sf-topology-type

     /*
      * Augmentations
      */
     /* Augmentations to network-types/te-topology */
     augment "/nw:networks/nw:network/nw:network-types/"
       + "tet:te-topology" {
       description
         "Defines the SF aware TE topology type.";
       uses sf-topology-type;
     }

     /* Augmentations to te-node-attributes */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-node-augmentation;
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
           + "tet:information-source-entry" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-node-augmentation;
     }

     /* Augmentations to tunnel-termination-point */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point" {
       description
         "Parameters for SF aware TE topology.";
       uses service-function-ttp-augmentation;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet-sf:service-function/"
       + "tet-sf:link-terminations/tet-sf:link-termination/"
       + "tet-sf:from" {
       description
         "Add reference to the link termination point.
          This portion cannot be shared with the state module.";
       leaf tp-ref {
         type leafref {
           path "../../../../../../../nt:termination-point/"
             + "nt:tp-id";
         }
         description
           "Reference to the link termination point.";
       }
     }
   }
   <CODE ENDS>
```

5. IANA Considerations

RFC Ed.: In this section, replace all occurrences of ’XXXX’ with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

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

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

This document registers the following YANG modules in the YANG Module Names registry [RFC7950]:

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

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

6. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

7. References

7.1. Normative References

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

[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>.

[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>.

[ETSI-NFV-MAN]
ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration", ETSI GS NFV-MAN 001 V1.1.1, December 2014.

[I-D.bryskin-teas-use-cases-sf-aware-topo-model]
Bryskin, I., Liu, X., Guichard, J., Lee, Y., Contreras, L., and D. Ceccarelli, "Use Cases for SF Aware Topology Models", draft-bryskin-teas-use-cases-sf-aware-topo-model-01 (work in progress), October 2017.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-20 (work in progress), December 2017.

[I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-14 (work in progress), February 2018.

[I-D.ietf-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-10 (work in progress), January 2018.

7.2. Informative References

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[I-D.ietf-netmod-yang-tree-diagrams]
Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-06 (work in progress), February 2018.

Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations

The YANG module ietf-te-topology-sf defined in this document is designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [I-D.ietf-netmod-revised-datastores]. In order to allow implementations to use the model even in cases when NMDA is not supported, the following companion module, ietf-te-topology-sf-state, is defined as state model, which mirrors the module ietf-te-topology-sf defined earlier in this document. However, all data nodes in the companion module are non-configurable, to represent the applied configuration or the derived operational states.

The companion module, ietf-te-topology-sf-state, is redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the companion module mirrors that of the coorespinding NMDA model, the YANG tree of the companion module is not depicted separately.

A.1. SF Aware TE Topology State Module

```
<CODE BEGINS> file "ietf-te-topology-sf-state@2018-02-27.yang"
module ietf-te-topology-sf-state {
  yang-version 1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-sf-state";

  prefix "tet-sf-s";

  import ietf-te-topology-sf {
    prefix "tet-sf";
  }

  import ietf-network-state {
    prefix "nw-s";
  }

  import ietf-network-topology-state {
    prefix "nt-s";
  }

  import ietf-te-topology-state {
    prefix "tet-s";
  }

  organization
    "Traffic Engineering Architecture and Signaling (TEAS)
```

```
        Working Group";

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

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

                  Xufeng Liu
                  <mailto:Xufeng_Liu@jabil.com>";

     description
       "Network service and function aware aware TE topology operational
        state model for non-NMDA compliant implementations.

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

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

     revision 2018-02-27 {
       description "Initial revision";
       reference "TBD";
     }

     /*
      * Augmentations
      */
     /* Augmentations to network-types/te-topology */
     augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
       + "tet-s:te-topology" {
       description
         "Defines the SF aware TE topology type.";
       uses tet-sf:sf-topology-type;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:te-node-attributes" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-node-augmentation;
     }

     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
           + "tet-s:information-source-entry" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-node-augmentation;
     }

     /* Augmentations to tunnel-termination-point */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:tunnel-termination-point" {
       description
         "Parameters for SF aware TE topology.";
       uses tet-sf:service-function-ttp-augmentation;
     }

     /* Augmentations to connectivity-matrix */
     augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
       + "tet-s:te-node-attributes/tet-sf-s:service-function/"
       + "tet-sf-s:link-terminations/tet-sf-s:link-termination/"
       + "tet-sf-s:from" {
       description
         "Add reference to the link termination point.
          This portion cannot be shared with the state module.";
       leaf tp-ref {
         type leafref {
           path "../../../../../../../nt-s:termination-point/"
             + "nt-s:tp-id";
         }
         description
           "Reference to the link termination point.";
       }
     }
   }
   <CODE ENDS>
```

Appendix B. Data Examples

B.1. A Topology with Multiple Connected Network Functions

```
                            Node-1
         +----o--o--------------------------o-------+
         |    |  |                          |       |
         |    \__/                          \__     |
         |    *\/ TTP-1   * * * * * * * * * *\/*    |
   LTP-4 |* * *         *                TTP-2  *   | LTP-1
         o------------*-----------------------------o
         |          *                             * |
   LTP-3 |* * * * *                                *| LTP-2
         o---                                  -----o
         |   \                                /     |
         |    \                              /      |
         |     \ CP01                   CP02/       |
         | +----o--------------------------o------+ |
         | | VL1|                       VL4|      | |
         | |    |CP11                      |CP33  | |
         | |  +-o--+        +----+       +-o--+   | |
         | |  |VNF1|        |VNF2|       |VNF3|   | |
         | |  +-o-o+  VL2   +--o-+  VL2  +-o-o+   | |
         | |CP12| |\----------/ \---------/| |CP32| |
         | |    | |CP13      CP21      CP31| |    | |
         | |    | |      VL2               | |    | |
         | |    | +------------------------+ |    | |
         | |    +----------------------------+    | |
         | |             VL3                      | |
         | |                  Network Service 1   | |
         | +--------------------------------------+ |
         +------------------------------------------+
```

The configuration instance data for Node-1 in the above figure could be as follows:

```
   {
     "networks": {
       "network": [
         {
           "network-types": {
             "te-topology": {
               "sf": {}
             }
           },
           "network-id": "network-sf-aware",
           "provider-id": 201,
           "client-id": 300,
           "te-topology-id": "te-topology:network-sf-aware",
           "node": [
             {
               "node-id": "Node-1",
               "te-node-id": "2.0.1.1",
               "te": {
                 "te-node-attributes": {
                   "domain-id": 1,
                   "is-abstract": [null],
                   "connectivity-matrices": {
                   },
                   "service-function": {
                     "connectivity-matrices": {
                       "connectivity-matrix": [
                         {
                           "id": 10,
                           "from": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP01"
                           },
                           "to": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP11"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL1"
                         },
                         {
                           "id": 13,
                           "from": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP12"
                           },
                           "to": {
                             "service-function-id": "VNF3",
                             "sf-connection-point-id": "CP32"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL3"
                         },
                         {
                           "id": 12,
                           "from": {
                             "service-function-id": "VNF1",
                             "sf-connection-point-id": "CP13"
                           },
                           "to": {
                             "service-function-id": "VNF2",
                             "sf-connection-point-id": "CP21"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL2"
                         },
                         {
                           "id": 23,
                           "from": {
                             "service-function-id": "VNF2",
                             "sf-connection-point-id": "CP21"
                           },
                           "to": {
                             "service-function-id": "VNF3"
                             "sf-connection-point-id": "CP31"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL2"
                         },
                         {
                           "id": 30,
                           "from": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP02"
                           },
                           "to": {
                             "service-function-id": "VNF3",
                             "sf-connection-point-id": "CP33"
                           }
                           "direction": "bidir",
                           "virtual-link-id": "VL4"
                         }
                       ]
                     },
                     "link-terminations": {
                       "link-termination": [
                         {
                           "id": 2,
                           "from": {
                             "tp-ref": "LTP-2"
                           },
                           "to": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP02"
                           }
                           "direction": "bidir"
                         },
                         {
                           "id": 3,
                           "from": {
                             "tp-ref": "LTP-3"
                           },
                           "to": {
                             "service-function-id": "Network Service 1",
                             "sf-connection-point-id": "CP01"
                           }
                           "direction": "bidir"
                         }
                       ]
                     }
                   }
                 }
                 "tunnel-termination-point": [
                   {
                     "tunnel-tp-id": 10001,
                     "name": "TTP-1",
                     "service-function-terminations": {
                     }
                   },
                   {
                     "tunnel-tp-id": 10002,
                     "name": "TTP-2",
                     "service-function-terminations": {
                     }
                   }
                 ]
               },
               "termination-point": [
                 {
                   "tp-id": "LTP-1",
                   "te-tp-id": 10001
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-2",
                   "te-tp-id": 10002
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-3",
                   "te-tp-id": 10003
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "LTP-4",
                   "te-tp-id": 10004
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-l2sc",
                         "encoding": "lsp-encoding-ethernet"
                       }
                     ]
                   }
                 }
               ]
             }
           ]
         }
       ]
     }
   }
```

B.2. A Topology with an Encapsulated Network Service

In this example, a network service consists of several interconnected network functions (NFs), and is represented by this model as an encapsulated opaque object without the details between its internals.

```
                         Node-1
      +----o--o--------------------------o-------+
      |    |  |                          |       |
      |    \__/                          \__     |
      |    *\/ TTP-1   * * * * * * * * * *\/*    |
LTP-4 |* * *         *               TTP-2  *   | LTP-1
      o------------*-----------------------------o
      |          *                             * |
LTP-3 |* * * * *                                *| LTP-2
      o---                                  -----o
      |   \                                /     |
      |    \                              /      |
      |     \ CP01                   CP02/       |
      | +----o--------------------------o------+ |
      | |                                      | |
      | |                  Network Service 1   | |
      | +--------------------------------------+ |
      +------------------------------------------+
```

The configuration instance data for Node-1 in the above figure could be as follows:

```
{
  "networks": {
    "network": [
      {
        "network-types": {
          "te-topology": {
            "sf": {}
          }
        },
        "network-id": "network-sf-aware",
        "provider-id": 201,
        "client-id": 300,
        "te-topology-id": "te-topology:network-sf-aware",
        "node": [
          {
            "node-id": "Node-1",
            "te-node-id": "2.0.1.1",
            "te": {
              "te-node-attributes": {
                "domain-id": 1,
                "is-abstract": [null],
                "connectivity-matrices": {
                },
                "service-function": {
                  "connectivity-matrices": {
                  },
                  "link-terminations": {
                    "link-termination": [
                      {
                        "id": 2,
                        "from": {
                          "tp-ref": "LTP-2"
                        },
                        "to": {
                          "service-function-id": "Network Service 1",
                          "sf-connection-point-id": "CP02"
                        }
                        "direction": "bidir"
                      },
                      {
                        "id": 3,
                        "from": {
                          "tp-ref": "LTP-3"
                        },
                        "to": {
                          "service-function-id": "Network Service 1",
                          "sf-connection-point-id": "CP01"
                        }
                        "direction": "bidir"
                      }
                    ]
                  }
                }
              }
              "tunnel-termination-point": [
                {
                  "tunnel-tp-id": 10001,
                  "name": "TTP-1",
                  "service-function-terminations": {
                  }
                },
                {
                  "tunnel-tp-id": 10002,
                  "name": "TTP-2",
                  "service-function-terminations": {
                  }
                }
              ]
            },
            "termination-point": [
              {
                "tp-id": "LTP-1",
                "te-tp-id": 10001
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-2",
                "te-tp-id": 10002
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-3",
                "te-tp-id": 10003
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              },
              {
                "tp-id": "LTP-4",
                "te-tp-id": 10004
                "te": {
                  "interface-switching-capability": [
                    {
                      "switching-capability": "switching-l2sc",
                      "encoding": "lsp-encoding-ethernet"
                    }
                  ]
                }
              }
            ]
          }
        ]
      }
```

```
        ]
      }
    }
```

Authors' Addresses

Igor Bryskin
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Xufeng Liu
Jabil

EMail: Xufeng_Liu@jabil.com

TEAS Working Group
Internet Draft
Intended status: Informational

Igor Bryskin
Huawei Technologies
Vishnu Pavan Beeram
Juniper Networks
Tarek Saad
Cisco Systems Inc
Xufeng Liu
Jabil

Expires: April 23, 2017
October 23, 2017

TE Topology and Tunnel Modeling for Transport Networks
draft-bryskin-teas-te-topo-and-tunnel-modeling-01

Status of this Memo

Copyright Notice

Abstract

This document describes how to model TE topologies and tunnels for transport networks, by using the TE topology YANG model [I-D.ietf-teas-yang-te-topo] and the TE tunnel YANG model [I-D.ietf-teas-yang-te].

Conventions used in this document

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

Table of Contents

1. Modeling Considerations

1.1. TE Topology Model

The TE Topology Model is written in YANG modeling language. It is defined and developed by the IETF TEAS WG and is documented as "YANG Data Model for TE Topologies" [I-D.ietf-teas-yang-te-topo]. The model describes a TE network provider's Traffic Engineering data store as it is seen by a client. It allows for the provider to convey to each of its clients:

- information on network resources available to the client in the form of one or several native TE topologies (for example, one for each layer network supported by the provider);

- one or several abstract TE topologies, customized on per-client basis and sorted according to the provider's preference as to how the abstract TE topologies are to be used by the client;

- updates with incremental changes happened to the previously provided abstract/native TE topology elements;

- updates on telemetry/state information the client has expressed interest in;

- overlay/underlay relationships between the TE topologies provided to the client (e.g. TE path computed in an underlay TE topology supporting a TE link in an overlay TE topology);

- client/server inter-layer adaptation relationships between the TE topologies provided to the client in the form of TE inter-layer locks or transitional links;

The TE Topology Model allows a network client to:

- (Re-)configure/negotiate abstract TE topologies provided to the client by a TE network provider, so that said abstract TE topologies optimally satisfy the client’s needs, constraints and optimization criteria, based on the client’s network planning, service forecasts, telemetry information extracted from the network, previous history of service provisioning and performance monitoring, etc.;

- Obtain abstract/native TE topologies from multiple providers and lock them horizontally (inter-domain) and vertically (inter-layer) into the client’s own native TE topologies;

- Configure, with each provider the trigger, frequency and contents of the TE topology update notifications;

- Configure, with each provider the trigger, frequency and contents of the TE topology telemetry (e.g. statistics counters) update notifications.

1.2. TE Topology Modeling Constructs

Figure 1. TE Topology

- o TE domain - a multi-layer traffic engineered network under direct and complete control of a single authority, network provider. TE domain can be described by one or more TE topologies. For example, separate TE topologies can describe each of the domain’s layer networks. TE domain can hierarchically encompass/parent other (child) TE domains, and can be encompassed by its own parent.

- o TE topology - a graphical representation of a TE domain. TE topology is comprised of TE nodes (TE graph vertices) interconnected via TE links (TE graph edges).

```
/* TE topology */
augment /nw:networks/nw:network:
   /* TE topology global ID */
   +--rw provider-id?      te-types:te-global-id
   +--rw client-id?        te-types:te-global-id
   +--rw te-topology-id?   te-types:te-topology-id
..................................................................
   /* TE topology general parameters */
     |  +--rw preference?               uint8
     |  +--rw optimization-criterion?   identityref
..................................................................
```

```
               /* TE topology list of TE nodes */
      augment /nw:networks/nw:network/nw:node:
         +--rw te-node-id?   te-types:te-node-id
      ..................................................................
               /* TE topology list of TE links */
      augment /nw:networks/nw:network/nt:link:
      ..................................................................
              /* TE topology list of TE link termination points */
      augment /nw:networks/nw:network/nw:node/nt:termination-point:
         +--rw te-tp-id?   te-types:te-tp-id
      ..................................................................
```

Figure 2. TE Node

- TE node - an element of a TE topology (appears as a vertex on TE graph). A TE node represents one or several nodes (physical switches), or a fraction of a node. A TE node belongs to and is fully defined in exactly one TE topology. A TE node is assigned a TE topology scope-unique ID. TE node attributes include information related to the data plane aspects of the associated node(s) (e.g. TE node’s connectivity matrix), as well as configuration data (such as TE node name). A given TE node can be reached on the TE graph, representing the TE topology, over one of TE links terminated by the TE node.

```
/* TE node */
augment /nw:networks/nw:network/nw:node:
   /* TE node ID */
   +--rw te-node-id?   te-types:te-node-id
..................................................................
   /* TE node general attributes */
      |  +--rw te-node-attributes */
..................................................................
   /* TE node connectivity matrices */
      |     +--rw connectivity-matrices
..................................................................
   /* TE node underlay TE topology */
           |     +--rw underlay-topology {te-topology-hierarchy}?
           |        +--rw network-ref?   leafref
..................................................................
   /* TE node information sources*/
      |  +--ro information-source-entry* [information-source]
..................................................................
  /* TE node statistics */
     +--ro statistics
..................................................................
  /* TE node TTP list */
     +--rw tunnel-termination-point* [tunnel-tp-id]
..................................................................
```

- TE link - an element of a TE topology (appears as an edge on TE graph), TE link is unidirectional and its arrow indicates the TE link's direction. Edges with two arrows on the TE topology graph (see Figure 1) represent bi-directional combinations of two parallel oppositely directed TE links. A TE link represents one or several physical links or a fraction of a physical link. A TE link belongs to and is fully defined in exactly one TE topology. A TE link is assigned a TE topology scope-unique ID. TE link attributes include parameters related to the data plane aspects of the associated link(s) (e.g. unreserved bandwidth, resource maps/pools, etc.), as well as the configuration data (such as remote node/link IDs, SRLGs, administrative colors, etc.) A TE link is connected to a TE node, terminating the TE link via exactly one TE link termination point (LTP).

```
   /* TE link */
   augment /nw:networks/nw:network/nt:link:
   /* TE link bundle information */
         |  +--rw (bundle-stack-level)?
         |  |  |  +--rw bundled-links
         |  |     +--rw component-links
   ..................................................................
   /* TE link general attributes */
       |  +--rw te-link-attributes

   ..................................................................
   /* TE link underlay TE topology */
         |     +--rw underlay! {te-topology-hierarchy}?
         |     |  +--rw primary-path
         |     |  +--rw backup-path* [index]

   ..................................................................
   /* TE link layer network */
        |     +--rw interface-switching-capability* [switching-
   capability encoding]

   ..................................................................
   /* TE link protection type */
        |     |  +--rw protection-type?   uint16

   ..................................................................

   /* TE link supporting TE tunnels */
       |     |  +--rw tunnels
```

```
      ..................................................................
      /* TE link transitional link flag */
           |  +--ro is-transitional?           empty


      ..................................................................
      /* TE link information sources */
            |  +--ro information-source?        te-info-source

      ..................................................................
      /* TE link statistics */
           +--ro statistics

      ..................................................................
```

- o Intra-domain TE link - TE link connecting two TE nodes within the same TE topology representing a TE network domain (e.g. L14 in Figure 1). From the point of view of the TE topology where the intra-domain TE link is defined, the TE link is close-ended, that is, both local and remote TE nodes of the link are defined in the same TE topology.

- o Inter-domain TE link - TE link connecting two border TE nodes that belong to separate TE topologies describing neighboring TE network domains (e.g. L3x in Figure 1). From the point of view of the TE topology where the inter-domain TE link is defined, the TE link is open-ended, that is, the remote TE node of the link is not defined in the TE topology where the local TE node and the TE link itself are defined.

  [Note: from the point of view of a TE node terminating an inter-domain TE link there is no difference between inter-domain and access TE links]

- o Access TE link - TE link connecting a border TE node of a TE topology describing a TE network domain to a TE node of a TE topology describing a customer network site (e.g. L1x in Figure 1) From the point of view of the TE topology where the access TE link is defined, the TE link is open-ended, that is, the remote TE node of the link (t.e. TE node representing customer network element(s)) is not defined in the TE topology where the local TE node and the TE link itself are defined.

[Note: from the point of view of a TE node terminating an access TE link there is no difference between access and inter-domain TE links]

- Dynamic TE link - a TE link that shows up in (and disappears from) a TE topology as a result of multi-layer traffic engineering. Dynamic TE link (supported by a hierarchy TE tunnel dynamically set up in a server layer network) is automatically (i.e. without explicit configuration request) added to a client layer network TE topology to augment the topology with additional flexibility to ensure successful completion of the path computation for and provisioning of a client layer network connection/LSP. For example, an ODUk hierarchy TE tunnel can support a dynamic Ethernet layer TE link to enable provisioning of an Ethernet layer connection on a network that does not have sufficient static Ethernet layer connectivity. Likewise, dynamic TE link is automatically removed from the TE topology (and its supporting hierarchy TE tunnel released) as soon as the TE link stops carrying client layer connections/LSPs.

- TE link termination point (LTP) - a conceptual point of connection of a TE node to one of the TE links terminated by the TE node (see Figure 2a). Unlike TE link, LTP is bi-directional - an inbound TE link and an oppositely directed outbound TE link have to be connected to the TE node via the same LTP to constitute a bi-directional TE link combination.

Figure 2a. Bi-directional TE link combination (left), independent uni-directional TE links (right)

```
/* LTP */
augment /nw:networks/nw:network/nw:node/nt:termination-point:
/* LTP ID */
   +--rw te-tp-id?   te-types:te-tp-id
/* LTP network layer ID */
      |  +--rw interface-switching-capability* [switching-
capability encoding]
      |  |  +--rw switching-capability    identityref
```

```
            |  |  +--rw encoding                identityref
      /* LTP bandwidth information */
            |  |  +--rw max-lsp-bandwidth* [priority]
            |  |     +--rw priority     uint8
            |  |     +--rw bandwidth?   te-bandwidth
      /* LTP inter-layer locks */
            |  +--rw inter-layer-lock-id?             uint32

      ..................................................................
```

o TE tunnel termination point (TTP) - an element of TE topology representing one or several potential TE tunnel termination/adaptation points (e.g. OCh layer transponder). A TTP is hosted by exactly one TE node (see Figure 2). A TTP is assigned a TE node scope-unique ID. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links originated/terminated from/by the TE node. TTP's important attributes include Local Link Connectivity List, Adaptation Client Layer List, TE inter-layer locks (see below), Unreserved Adaptation Bandwidth (announcing the TTP's remaining adaptation resources sharable between all potential client LTPs), and Property Flags (indicating miscellaneous properties of the TTP, such as capability to support 1+1 protection for a TE tunnel terminated on the TTP).

```
      /* TTP */
           +--rw tunnel-termination-point* [tunnel-tp-id]
      /* TTP ID */
               +--rw tunnel-tp-id                          binary
      /* TTP layer network ID */
               |  +--rw switching-capability?        identityref
               |  +--rw encoding?                    identityref
      //* Inter-layer-locks supported by TTP */
               |  +--rw inter-layer-lock-id?         uint32
      /* TTP's protection capabilities */
               |  +--rw protection-type?             identityref
      /* TTP's list of client layer users */
               |  +--rw client-layer-adaptation

      ..................................................................
      /* TTP's Local Link Connectivity List (LLCL) */
```

```
               |  +--rw local-link-connectivities

      ..................................................................
```

o Label - in the context of circuit switched layer networks identifies a particular resource on a TE link (e.g. Och wavelength, ODUk container)

```
      +--:(label)
         +--rw value?   rt-types:generalized-label
```

Figure 3. TTP Local Link Connectivity List

- TTP basic local link connectivity list (basic LLCL) - a list of TE link/label combinations terminated by the TTP-hosting TE node (effectively the same as LTP/label pairs), which the TTP could be connected to (see Figure 3, upper left). From the point of view of a potential TE path, basic LLCL provides a list of permissible LTP/label pairs the TE path needs to start/stop on for a connection, taking the TE path, to be successfully terminated on the TTP in question.

- TTP detailed local link connectivity list (detailed LLCL) - basic LLCL extended to provide a set of costs (such as intra-node summary TE metric, delay, SRLGs, etc.) associated with each LLCL entry (see Figure 3, upper right)

```
   /* TTP LLCL */
   |  +--rw local-link-connectivities
       |     +--rw number-of-entries?         uint16
       /* LLCL entry */

       /* LLCL entry LTP */
       |        +--rw link-tp-ref                leafref

   ..................................................................

   /* LLC entry label range */
   |     +--rw label-restriction* [inclusive-exclusive label-start]
   |     |  +--rw inclusive-exclusive    enumeration
   |     |  +--rw label-start            rt-types:generalized-label
         |     |  +--rw label-end?             rt-types:generalized-
   label
         |     |  +--rw range-bitmap?          binary

   /* LLCL entry underlay TE path(s) */
   |     +--rw underlay! {te-topology-hierarchy}?
   |     |  +--rw primary-path
   |     |  +--rw backup-path* [index]
   /* LLCL entry protection type */
   |     |  +--rw protection-type?   uint16
   /* LLCL entry supporting TE tunnels */
   |     |  +--rw tunnels
   /* LLCL entry bandwidth parameters */
   |     +--rw max-lsp-bandwidth* [priority]

   ..................................................................
```

```
/* LLCL entry metrics  (vector of costs) */
|     +--rw te-default-metric?         uint32
|     +--rw te-delay-metric?           uint32
|     +--rw te-srlgs
|     |  +--rw value*   te-types:srlg
|     +--rw te-nsrlgs {nsrlg}?

..................................................................
/* LLCL entry ID */
|     |  +--rw id*   uint32
```

- o TTP adaptation client layer list - a list of client layers that could be directly adopted by the TTP. This list is necessary to describe complex multi-layer (more than two layer) client-server layer hierarchies and, in particular, to identify the position of the TTP in said hierarchies.

```
/* TTP adaptation client layer list */
         |  +--rw client-layer-adaptation
         |  |  +--rw switching-capability* [switching-capability
encoding]
         /* Client layer ID */
         |  |     +--rw switching-capability    identityref
         |  |     +--rw encoding                identityref
         /* Adaptation bandwidth available for the client layer */
         |  |     +--rw bandwidth?              te-bandwidth
```

Figure 4. TE Node Connectivity Matrix

- TE node basic connectivity matrix - a TE node attribute describing the TE node’s switching capabilities/limitations in the form of permissible switching combinations of the TE node’s LTP/label pairs (see Figure 4, upper left). From the point of view of a potential TE path arriving at the TE node at a given inbound LTP/label, the node’s basic connectivity matrix describes permissible outbound LTP/label pairs for the TE path to leave the TE node.

- TE node detailed connectivity matrix - TE node basic connectivity matrix extended to provide a set of costs (such as intra-node summary TE metric, delay, SRLGs, etc.) associated with each connectivity matrix entry (see Figure 4, upper right).

```
/* TE node connectivity matrix */
         |  +--rw connectivity-matrix* [id]
         |     +--rw id                            uint32
         |     +--rw from  /* left LTP */
```

```
               |     |  +--rw tp-ref?   leafref
               |     +--rw to    /* right LTP */
               |     |  +--rw tp-ref?   leafref
               |     +--rw is-allowed?               boolean

               /* Connectivity matrix entry label range */
               |     +--rw label-restriction* [inclusive-exclusive
      label-start]
               |     |  +--rw inclusive-exclusive    enumeration
               |     |  +--rw label-start            rt-
      types:generalized-label
               |     |  +--rw label-end?             rt-
      types:generalized-label
               |     |  +--rw range-bitmap?          binary

              /* Connectivity matrix entry underlay TE path(s) */
               |     +--rw underlay! {te-topology-hierarchy}?
               |     |  +--rw primary-path
               |     |  +--rw backup-path* [index]
               /* Connectivity matrix entry protection type */
               |     |  +--rw protection-type?   uint16
               /* Connectivity matrix entry supporting TE tunnels */
               |     |  +--rw tunnels
               /* Connectivity matrix entry bandwidth parameters */
               |     +--rw max-lsp-bandwidth* [priority]

      ..................................................................
               /* Connectivity matrix entry metrics (vector of costs) */
               |     +--rw te-default-metric?        uint32
               |     +--rw te-delay-metric?          uint32
               |     +--rw te-srlgs
               |     |  +--rw value*   te-types:srlg
               |     +--rw te-nsrlgs {nsrlg}?

      ..................................................................
               /* Connectivity matrix entry ID */
               |     |  +--rw id*   uint32
```

Figure 5. TE Path

- TE path - an ordered list of TE node/link IDs (each possibly augmented with labels) that interconnects over a TE topology a pair of TTPs and could be used by a connection (see Figure 5). A TE path could, for example, be a product of a successful path computation performed for a given TE tunnel

---

```
   /* TE path */

         /* TE topology the path is defined in */
     |     |  |  +--rw network-ref?    leafref
         /* Path type (IRO, XRO, ERO, RRO) */
     |     |  |  +--rw path-type?    identityref

         /* TE path elements */
     |     |  |  +--rw path-element* [path-element-id]
     |     |  |     +--rw path-element-id    uint32
     |     |  |     +--rw index?             uint32
     |     |  |     +--rw (type)?
          /* Numbered TE link path element */
     |     |  |        +--:(ip-address)
     |     |  |        |  +--rw ip-address-hop
     |     |  |        |     +--rw address?    inet:ip-address
```

```
       |     |  |        |     +--rw hop-type?   te-hop-type
             /* AS number path element */
       |     |  |        +--:(as-number)
       |     |  |        |  +--rw as-number-hop
       |     |  |        |     +--rw as-number?   binary
       |     |  |        |     +--rw hop-type?    te-hop-type
             /* Unnumbered TE link path element */
       |     |  |        +--:(unnumbered-link)
       |     |  |        |  +--rw unnumbered-hop
       |     |  |        |     +--rw te-node-id?      inet:ip-address
       |     |  |        |     +--rw tp-id?   uint32
       |     |  |        |     +--rw hop-type?       te-hop-type
             /* Label path element */
       |     |  |        +--:(label)
       |     |  |        |  +--rw label-hop
       |     |  |        |     +--rw value?   rt-types:generalized-label
       |     |  |        |     +--rw direction?       boolean
       |     |  |        +--:(sid)
       |     |  |           +--rw sid-hop
       |     |  |              +--rw sid?   rt-types:generalized-label
```

---

o TE path segment - a contiguous fragment of a TE path

Figure 6. TE Inter-Layer Lock

- TE inter-layer lock - a modeling concept describing client-server layer adaptation relationships important for multi-layer traffic engineering. It is an association of M client layer LTPs and N server layer TTPs, within which data arriving at any of the client layer LTPs could be adopted onto any of the server layer TTPs. A TE inter-layer lock is identified by inter-layer lock ID, which is unique across all TE topologies provided by the same provider. The client layer LTPs and the server layer TTPs associated by a given TE inter-layer lock share the same inter-layer lock ID value.

  In Figure 6 a TE inter-layer lock IL_1 associates six client layer LTPs (C_LTP_1 - C_LTP_6) with two server layer TTPs (S_TTP_1 and S_TTP_2). As mentioned, they all have the same attribute -inter-layer lock ID: IL_1, which is the only parameter/value indicating the association. A given LTP may have zero, one or more inter-layer lock IDs. In the case of multiple inter-layer lock IDs, this implies that the data arriving at the LTP can be adopted onto any of TTPs associated with all specified inter-layer locks. For example, C_LTP_1 may be attributed with two inter-layer locks- IL_1 and IL_2. This would mean that C_LTP_1 for adaptation purposes can use not just TTPs associated with inter-layer lock IL_1 (i.e. S_TTP_1 and S_TTP_2 in the Figure), but any of TTPs associated with inter-layer lock IL_2. Likewise, a given TTP may have one or more inter-layer locks, meaning that it can offer the adaptation service to any client layer LTP having an inter-layer lock matching one of its own.

  LTPs and TTPs associated within the same TE inter-layer lock may be hosted by the same (hybrid, multi-layer) TE node or by multiple TE nodes defined in the same or separate TE topologies. The latter case is especially important because TE topologies of different layer networks could be modeled by separate augmentations of the basic (common to all layers) TE topology model.

  ```
          |  +--rw inter-layer-lock-id?         uint32
  ```

- Transitional link - an alternative method of modeling of client-server adaptation relationship. Transitional link is a bi-directional link connecting an LTP in a client layer to an LTP in a server layer, which is associated (via TTP’s LLCL) with a server layer TTP capable of adopting of the client layer data onto a TE tunnel terminated by the TTP. Important attributes pf a transitional link are loca;/remote LTP IDs, TE metric and available adaptation bandwidth.

Figure 7. Native and Abstract TE Topologies

- o Native TE topology - a TE topology as it is known (to full extent and unmodified) to the TE topology provider (see lower part of Figure 7.). A native TE topology might be discovered via various routing protocols and/or subscribe/publish techniques. For example, a first-level TE topology provider (such as a T-SDN Domain Controller, DC) may auto-discover its native TE topology(ies) by participating in the domain’s OSPF-TE protocol instance; while a second-level TE topology provider (such as a Hierarchical T-SDN Controller. HC) normally builds its native TE topology(ies) based on TE topologies exposed by each of the subordinate, first- level TE topology providers.

- o Underlay TE topology - a TE topology that serves as a base for constructing overlay TE topologies.

- o Overlay TE topology - a TE topology constructed based on one or more underlay TE topologies. Each TE node of the overlay TE topology represents a separate underlay TE topology (that could be mapped onto an arbitrary segment of a native TE topology). Each TE link of the overlay TE topology represents, generally speaking, an arbitrary TE path in one of the underlay TE topologies. The overlay TE topology and the supporting underlay TE topologies may represent separate layer networks (e.g. OTN/ODUk and WDM/OCh respectively) or the same layer network.

- o Abstract TE topology - an overlay TE topology created by a provider to describe its network in some abstract way. An abstract TE topology contains at least one abstract TE topology element, such as TE node or TE link. An abstract TE topology is built based on contents of one or more of the provider’s native TE topologies (serving as underlay(s)), the provider’s policies and the client’s preferences (see upper part of Figure 7).

- o Customized TE topology - a TE topology tailored for a given provider’s client. A customized TE topology is usually but not always an abstract TE topology. For example, a given abstract TE topology could be exposed to a group or all provider’s clients (in which case the abstract TE topology is not a customized TE topology). Likewise, a given naive TE topology could be customized for a given client (for example, by removing high delay TE links the client does not care about). So customized TE topology is not an abstract TE topology, because it does not contain abstract TE topology elements

- o TE inter-domain plug - a TE link attribute meaningful for open-ended inter-domain/access TE links. It contains a network-wide unique value (inter-domain plug ID) that identifies in the network a connectivity supporting the inter-domain/access TE link in question. It is expected that a given pair of neighboring domain TE topologies (provided by separate providers) will have each at least one open-ended inter-domain/access TE link with a TE inter-domain plug matching to one provided by its neighbor, thus allowing for a client of both domains to identify adjacent nodes in the separate neighboring TE topologies and resolve the open-ended inter-domain/access TE links by connecting them regardless of the links respective local/remote node ID/link ID attributes. Inter-domain plug IDs may be assigned and managed by a central network authority. Alternatively, inter-domain plug IDs could be dynamically auto-discovered (e.g. via LMP protocol).

```
        +--rw external-domain
           |  +--rw network-ref?           leafref
           |  +--rw remote-te-node-id?     te-types:te-node-id
           |  +--rw remote-te-link-tp-id?  te-types:te-tp-id
           |  +--rw plug-id?               uint32
```

1.3. Abstract TE Topology Calculation, Configuration and Maintenance

The TE Topology Model does not prescribe what and how abstract TE topologies are computed, configured, manipulated and supported by a TE network (e.g. transport network) provider. However, it is assumed that:

o All TE topologies, native or abstract, conveyed to the same or different clients, are largely independent one from another. This implies that each TE topology, generally speaking, has an independent name space for TE node and link IDs, SRLGs, etc. (possibly overlapping with the name spaces of other TE topologies);

o All abstract TE topologies are bound to the respective underlay native or abstract TE topologies only by the overlay/underlay relationships defined by the TE Topology Model, but, otherwise, the abstract TE topologies are decoupled from their respective underlay TE topologies.

It is envisioned that an original set of abstract TE topologies is produced by a TE network provider for each of its clients based on the provider’s local configurations and/or policies, as well as the client-specific profiles. The original set of abstract TE topologies offered to a client may be accepted by the client as-is. Alternatively, the client may choose to negotiate/re-configure the abstract TE topologies, so that the latter optimally satisfy the client’s needs. In particular, for each of the abstract TE topologies the client may request adding/removing TE nodes, TE links, TTPs and/or modifying re-configurable parameters of the existing components. The client may also request different optimization criteria as compared to those used for the original abstract TE topology optimization, or/and specify various topology-level constraints. The provider may accept or reject all or some abstract TE topology re-configuration requests. Hence, the abstract TE topology negotiation process may take multiple iterations before the provider and each of its clients agree upon a set of abstract TE topologies and their contents. Furthermore, the negotiation process could be repeated over time to produce new abstract TE topologies optimal to best suit evolving circumstances.

Figure 8. Native Transport Network Domain TE Topology as an Underlay for Abstract TE Topologies

Let’s assume that a native transport network domain TE topology to be as depicted in Figure 8. The popular types of abstract TE topologies based on this native TE topology as an underlay are described in the following sections.

1.3.1. Single-Node Abstract TE Topology

Figure 9. Blocking/Asymmetrical TE Node with Basic Connectivity Matrix Attribute

In Figure 9, the transport network domain is presented to a client as a one-node abstract TE topology, where the single TE node (AN1) represents the entire domain and terminates all of the inter-domain/access TE links connecting the domain to its adjacent domains (i.e. TE links L1...L8). Because AN1 represents the entire domain the node's Underlay TE Topology attribute matches the ID of one of the domain's native TE topologies (e.g. one presented in Figure 8). [Note: all or some of the underlay TE topologies a given abstract TE topology depends on could be catered to the client by the provider along with the abstract TE topology in question or upon separate request(s) issued by the client.]

One important caveat about abstract TE node AN1 is that it should be considered as an asymmetrical/blocking switch, because, generally speaking, it is not guaranteed that a suitable TE path exists between any given pair of inter-domain TE links into/out of the domain. This means from the TE Topology model point of view that there are certain limitations as to how AN1's LTPs could be interconnected inside/across the TE node. The model allows for asymmetrical/blocking switches by specifying for the associated TE nodes a non-empty basic connectivity matrix attribute describing permissible inbound-outbound TE link/label switching combinations. It is assumed that the provider's path computer can compute a set of optimal TE paths, connecting inbound TE link/label_x <=> outbound TE link/label_y combinations inside the abstract TE node over the TE node's underlay TE topology. Based on the results of such computations, AN1's connectivity matrix can be (re-)generated and (re-)conveyed to the abstract TE topology client.

A richer version of the basic connectivity matrix is the detailed connectivity matrix. The latter not only describes permissible inbound TE link/label_x <=> TE link/label TE link/label_y switching combinations, but also provides connectivity matrix entry specific vectors of various costs/metrics (in terms of delay, bandwidth, intra-node SRLGs and summary TE metrics) that a potential TE path will accrue, should a given connectivity matrix entry be selected by the path for crossing the TE node (see Figure 10).

Figure 10. Blocking/Asymmetrical TE Node with Detailed Connectivity Matrix Attribute

1.3.2. Full Mesh Link Abstract TE Topology

Figure 11. Full Mesh Link Abstract TE Topology

In Figure 11, the transport network domain is abstracted in the following way.

o Each of the underlay native TE topology border TE nodes (i.e., the TE nodes terminating at least one inter-domain/access TE link, such as TE nodes S3 or S11 in Figure 8) is represented in the abstract TE topology as a separate abstract TE node, matching one-for-one to the respective border TE node of the underlay TE topology. For example, S3’ of the abstract TE topology represents S3 of the underlay TE topology in Figure 8. [Note that such a relationship is modeled via Supporting Node attribute of TE node S3’ specifying the ID of S3, as well as the ID of the TE topology where S3 is defined (i.e. TE topology in Figure 8)]. Likewise, S9’ represents S9, S11’ represents S11 and so forth;

- TE nodes S3’, S5’, S8’, S9’ and S11’ are interconnected via a full mesh of abstract TE links. It is assumed that the provider’s path computer can compute a set of optimal TE paths over one or more of underlay TE topologies (such as presented in Figure 8)- one for each of said abstract TE links; and the provider can set up the TE tunnels in the network supporting each of the abstract TE links, either during the abstract TE topology configuration (in the case of committed/pre-established abstract TE links), or at the time the first client’s connection is placed on the abstract TE link in question (the case of uncommitted abstract TE links). [Note that so (re-)computed TE paths, as well as the IDs of respective underlay TE topologies used for their computation are normally catered to the client in the Underlay TE path attribute of the associated abstract TE links]

The configuration parameters of each of the abstract TE links (such as layer ID, bandwidth and protection requirements, preferred TE paths across the underlay TE topology for the primary and backup connections, etc.) are expected to be found in the abstract TE topology profiles/templates locally configured with the provider or pushed to the provider by the client via the policy NBI. Each of the abstract TE links may be later re-configured or removed by direct configuration requests issued by the client via TE Topology NBI. Likewise, additional abstract TE links may be requested by the client at any time.

Some possible variants/flavors of the Full Mesh Link Abstract TE Topology described above are:

- Partial Mesh Link Abstract TE Topology (where some of the abstract TE links from the full mesh are missing);

- Double Mesh Link Abstract TE Topology (where each pair of abstract TE nodes is connected via two diverse abstract TE links).

1.3.3. Star-n-Spokes Abstract TE Topology

Figure 12. Star-n-Spoke Abstract TE Topology

The Full Mesh Link Abstract TE Topology suffers from the n-squared problem; that is, the number of required abstract TE links is proportional to square of the number of native TE topology border TE nodes. This problem can be mitigated (i.e., the number of required abstract TE links may be significantly reduced) by adding, to the abstract TE topology, an additional abstract TE node (the star) representing one or several interconnected non-border TE nodes from the native TE topology. Abstract TE links in the Star-n-Spokes Topology connect the star with all other TE nodes of the topology (the spokes). For example, abstract TE node AN1 in Figure 12 could represent collectively TE nodes S7, S10 and S4 of the native TE topology (see Figure 8) with abstract TE links connecting AN1 with all other TE nodes in the Star-n-Spokes Abstract TE Topology in Figure 12.

In order to introduce a composite abstract TE node, (e.g. AN1 in Figure 12) representing in a given abstract TE topology an arbitrary segment of another TE topology (e.g. TE nodes S7, S12 and S4 of the TE topology in Figure 8) the TE topology provider is expected to perform the following operations:

- o Copy the TE topology segment to be represented by the abstract TE node (i.e. TE nodes S7, S10 and S4 in Figure 8, as well as the TE links interconnecting them) into a separate auxiliary TE topology (with a separate TE topology ID);

- Set for each TE node and TE link of the auxiliary TE topology the Supporting Node/Link attribute matching the original TE topology ID, as well as the ID of the respective original TE node/link of the original TE topology. For example, if S7" of the auxiliary TE topology is a copy of S7 of the original TE topology, the Supporting Node attribute of S7" will specify the ID of the original TE topology (presented in figure 8) and the ID of S7;

- Set for the abstract TE node AN1 the Underlay TE Topology attribute matching the auxiliary TE Topology ID

Furthermore, the Star-n-Spokes Abstract TE topology provider is expected to:

- Compute/provision TE paths/tunnels supporting each of the abstract TE links in Figure 12 (i.e. abstract TE links connecting the spokes to the star, AN1) as described in 1.3.2;

- Generate the AN1’s Basic/Detailed Connectivity Matrix attribute based on intra-node path computations performed on the AN1’s underlay (i.e. auxiliary) TE topology and describing permissible inbound TE link/label_x. outbound TE link/label_y switching combinations as described in 1.3.1

1.3.4. Arbitrary Abstract TE Topology

Figure 13. Arbitrary Abstract TE Topology

To achieve an optimal tradeoff between the number of components, the amount of information exposed by a transport network provider and the amount of path computations required to keep said information up-to-date, the provider may present the TE network domain as an arbitrary abstract TE topology comprised of any number of abstract TE nodes interconnected by abstract TE links (see Figure 13). Each of the abstract TE nodes can represent a single or several interconnected TE nodes from the domain’s underlay (native or lower level abstract) TE topology, or a fraction of an underlay TE node. [Note that each of the abstract TE nodes of the TE topology in Figure 13 is expected to be introduced and maintained by the provider following the instructions as described in 1.3.3; likewise, each of the abstract TE links of the topology is expected to be computed, provisioned and maintained as described in 1.3.2]

1.3.5. Customized Abstract TE Topologies

Figure 14. Customized Abstract TE Topology(ies)

A transport network/domain provider may serve more than one client. In such a case, the provider "slices" the network/domain resources and exposes a slice for each of the clients in the form of a customized abstract TE topology. In Figure 14, the provider serves two clients (Blue and Red). Client Blue is provided with the Blue abstract TE topology supported by the blue TE tunnels or paths in the underlay (native) TE topology (depicted in the Figure with blue broken lines). Likewise, client Red is provided with the Red abstract TE topology supported by the red TE tunnels or paths in the underlay TE topology.

1.3.6. Hierarchical Abstract TE Topologies

Figure 15. Hierarchy of Abstract TE Topologies

As previously mentioned, an underlay TE topology for a given abstract TE topology component does not have to be one of the domain’s native TE topologies - another (lower level) domain’s abstract TTE topology can be used instead. This means that abstract TE topologies are hierarchical in nature.

Figure 15 provides an example of abstract TE topology hierarchy. In this Figure the blue topology is a top level abstract TE topology catered to by the provider to one of the domain’s clients. One of the TE links of the blue topology - link EF - is supported by a TE path E’-M-P-Q-N-F’ computed in the underlay TE topology (red topology), which happens to be domain’s (lower level) abstract TE topology.. Furthermore, as shown, the TE link PQ - one of the TE links comprising the E’-M-P-Q-N-F’ path - is supported by its own underlay

TE path, P’-X-Q’ - computed on one of the domain’s native TE topologies.

Importantly, each TE link and TE node of a given abstract TE topology has, generally speaking, its individual stack/hierarchy of underlay TE topologies.

1.4. Merging TE Topologies Provided By Multiple Providers

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

In particular, the process of merging TE topologies includes:

- Identifying neighboring TE domains and locking their TE topologies horizontally by connecting their inter-domain open-ended TE links;

- Renaming TE node, link, and SRLG IDs into ones allocated from a separate name space; this is necessary because all TE topologies are considered to be, generally speaking, independent with a possibility of clashes among TE node, link or SRLG IDs. Original TE node/link IDs along with the original TE topology ID are stored in the Source attribute of the respective TE nodes/links of the merged TE topology;

- Locking, TE topologies associated with different layer networks vertically according to provided TE inter-layer locks; this is to facilitate inter-layer path computations across multiple TE topologies provided by the same topology provider.

Figure 16. Merging Domain TE Topologies

Figure 16 illustrates the process of merging, by the client, of TE topologies provided by the client’s providers.

In the Figure, each of the two providers caters to the client a TE topology (abstract or native), describing the network domain under the respective provider’s control. The client, by consulting the attributes of the open-ended inter-domain/access TE links - such as TE inter-domain plugs or remote TE node/link IDs - is able to determine that:

1. the two domains are adjacent and are interconnected via three inter-domain TE links, and;

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

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

As mentioned, one way to interconnect the open-ended inter-domain/access TE links of neighboring domains is to mandate the providers to specify remote nodeID/linkID attributes in the provided inter-domain/access TE links. This, however, may prove to be not flexible. For example, the providers may not be aware of the respective remote nodeID/linked values. More importantly, this option does not allow for the client to mix-n-match multiple (more than one) TE topologies catered by the same providers (see the next section). Another, more flexible, option to resolve the open-ended inter-domain/access TE links is by decorating them with the TE inter-domain plug attribute. The attribute specifies inter-domain plug ID - a network-wide unique value that identifies on the network connectivity supporting a given inter-domain/access TE link. Instead of specifying remote node ID/link ID, an inter-domain/access TE link may provide a non-zero inert-domain plug ID. It is expected that two neighboring domain TE topologies (provided by separate providers) will have each at least one open-ended inter-domain/access TE link with a TE inter-domain plug matching to one provided by its neighbor. For example, the inter-domain TE link originating from node S5 of the Domain 1 TE topology (Figure 8) and the inter-domain TE link coming from node S3 of Domain2 TE topology may specify matching TE inter-domain plugs (i.e. carrying the same inter-domain plug ID). This would allow for the client to identify adjacent nodes in the separate neighboring TE topologies and resolve the inter-domain/access TE links connecting them regardless of their respective nodeIDs/linkIDs (which, as mentioned, could be allocated from independent name spaces).

Inter-domain plug IDs may be assigned and managed by a central network authority. Alternatively, inter-domain plug IDs could be dynamically auto-discovered (e.g. via LMP protocol).

Furthermore, the client renames the TE nodes, links and SRLGs offered in the abstract TE topologies by assigning to them IDs allocated from a separate name space managed by the client. Such renaming is necessary, because the two abstract TE topologies may have their own name spaces, generally speaking, independent one from another; hence, ID overlaps/clashes are possible. For example, both TE topologies have TE nodes named S7, which, after renaming, appear in the merged TE topology as S17 and S27 respectively. IDs of the original (i.e. abstract TE topology) TE nodes/links along with the ID of the abstract TE topology they belong to are stored in the Source attribute of the respective TE nodes/links of the merged TE topology. For example, the Source attribute of S27 will contain S7 and the TE topology ID of the abstract TE topology describing domain 2.

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

1.4.1. Dealing With Multiple Abstract TE Topologies Provided By The Same Provider

Figure 17. Multiple Abstract TE Topologies Provided By TE Topology Providers

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

It should be up to the client to decide how to mix-and-match multiple abstract TE topologies provided by each of the providers, as well as how to merge them into the client’s native TE topologies. The client also decides how many such merged TE topologies it needs to produce and maintain. For example, in addition to the merged TE topology depicted on the upper part of Figure 16, the client may merge the abstract TE topologies received from the two providers, as shown in Figure 17, into the client’s additional native TE topologies, as shown in Figure 18.

[Note: allowing for the client mix-n-matching of multiple TE topologies assumes that TE inter-domain plugs (rather than remote nodeID/linked) option is used for identifying neighboring domains and inter-domain/access TE link resolution.]

Figure 18. Multiple Native (Merged) Client’s TE Topologies

It is important to keep in mind that each of the three native (merged) TE topologies could be used by the client for computing TE paths for any of the multi-domain connections. The choice as to which topology to use for a given connection depends on the connection/tunnel parameters/requirements and the topology's style and optimization criteria.

1.5. Configuring Abstract TE Topologies

When a client receives one or more abstract TE topologies from one of its providers, it may accept the topologies as-is and merge then into one or more of its own native TE topologies. Alternatively, the client may choose to request a re-configuration of one, some or all abstract TE topologies provided by the providers. Specifically, with respect to a given abstract TE topology, some of its TE nodes/links may be requested to be removed, while additional ones may be requested to be added. It is also possible that existing TE nodes/links may be asked to be re-configured. For example, a set of TE links may be requested to be disjoint from each other by configuring the same Non Sharing Risk Link Group (NSRLG) attribute for all links from the set. Such a configuration would force the provider to place TE tunnels supporting the TE links from the set onto sufficiently disjoint TE paths computed in the tunnels underlay TE topology. Furthermore, the topology-wide optimization criteria may be requested to be changed. For example, underlay TE paths supporting the abstract TE links, currently optimized to be shortest (least-cost) paths, may be requested to be re-optimized based on the minimal delay criteria. Additionally, the client may request the providers to configure entirely new abstract TE topologies and/or to remove existing ones. Furthermore, future periodic or one time additions, removals and/or re-configurations of abstract TE topology elements and/or their attributes could be (re-)scheduled by the client ahead of time.

It is the responsibility of the client to implement the logic behind the above-described abstract TE topology negotiation. It is expected that the logic is influenced by the client's local configuration/templates, policies conveyed by client's clients, input from the network planning process, telemetry processor, analytics systems and/or direct human operator commands. Figure 19 exemplifies the abstract TE topology negotiation process. As shown in the Figure, the original abstract TE topology exposed by a provider was requested to be re-configured. Specifically, one of the abstract TE links was asked to be removed, while three new ones were asked to be added to the abstract TE topology.

Figure 19. Provider. Client Abstract TE Topology Negotiation

1.6. TE Tunnel Model

The TE Tunnel Model is written in YANG modeling language. It is defined and developed by the IETF TEAS WG and is documented as "YANG Data Model for Traffic Engineering Tunnels and Interfaces" [I-D.ietf-teas-yang-te]. Among other things the model describes a TE network provider's TE Tunnel data store as it is seen and influenced by a client.

The TE Tunnel Model allows for the provider to convey to each of its clients:

- information on TE tunnels provided to the client that are fully contained within the controlled network domain,

- information on multi-domain TE tunnel segments across the network domain controlled by the provider;

- information on connections/LSPs, supporting TE tunnels and TE tunnel segments;

- updates in response to changes to the client's active TE tunnels/segments and the connections supporting them,

- updates in response to the TE tunnel/segment telemetry/state information the client has expressed an interest in.

The TE Tunnel Model allows for a TE network client to:

- Issue configuration requests to set up, tear down, replace, modify and manipulate end-to-end TE tunnels, as well as segments of multi-domain TE tunnels across the network controlled by the provider;

- Request and obtain information on active TE tunnels/segments and connections supporting them;

- Subscribe to and configure with the provider triggers, pace and contents of the TE tunnel/segment change update notifications;

- Subscribe to and configure with the provider triggers, pace and contents of the TE tunnel/segment event notifications, such as detected alarms, faults, protection/restoration actions, etc..

- Subscribe to and configure with the provider triggers, pace and contents of TE tunnel/segment telemetry (e.g. statistics counters) update notifications.

1.7. TE Tunnel/Transport Service Modeling Constructs

Figure 20. TE tunnel

- o TE tunnel - a connection-oriented service provided by a layer network of delivery of a client’s data between source and destination tunnel termination points. A TE tunnel in a server layer network may support a link in a client layer network (e.g. OCh layer TE tunnel supporting ODU4 link). In Figure 20, a TE tunnel interconnects tunnel termination points resident on switches C-R2 and C-R3. A TE tunnel is realized via (supported by, mapped onto) one or more layer network connections/LSPs

```
   /* TE tunnel */
    |  +--rw tunnel* [name]
         |  |  +--rw name                   leafref
```

```
            |  |  +--rw identifier?            leafref
      /* TE tunnel configuration parameters */
            |  |  +--rw config
            |  |  |  +--rw name?                   string
            |  |  |  +--rw type?                   identityref
            |  |  |  +--rw identifier?             uint16
            |  |  |  +--rw description?            string
            |  |  |  +--rw switchcap?              identityref
            |  |  |  +--rw encoding?               identityref
            |  |  |  +--rw protection-type?        identityref
            |  |  |  +--rw admin-status?           identityref
            |  |  |  +--rw preference?             uint8
            |  |  |  +--rw reoptimize-timer?       uint16
            |  |  |  +--rw source?                 inet:ip-address
            |  |  |  +--rw destination?            inet:ip-address
            |  |  |  +--rw src-tp-id?              binary
            |  |  |  +--rw dst-tp-id?              binary
            |  |  |  +--rw topology-id?            te-types:te-topology-
      id
            |  |  |  +--rw ignore-overload?        boolean
            |  |  |  +--rw bandwidth-generic?      te-types:te-bandwidth
            |  |  |  +--rw disjointness?           te-types:te-path-
      disjointness
            |  |  |  +--rw setup-priority?         uint8
            |  |  |  +--rw hold-priority?          uint8
            |  |  |  +--rw signaling-type?         identityref
      /* Hierarchy TE tunnel parameters */
            |  |  |  +--rw hierarchical-link-id
            |  |  |  |  +--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-id?        te-types:te-
      topology-id
      /* Bidirectional TE tunnel parameters */
            |  |  |  +--rw bidirectional
            |  |  |     +--rw association
            |  |  |        +--rw id?              uint16
            |  |  |        +--rw source?          inet:ip-address
            |  |  |        +--rw global-source?   inet:ip-address
            |  |  |        +--rw type?            identityref
            |  |  |        +--rw provisioing?     identityref
      /* TE tunnel state */
            |  |  +--ro state
            |  |  |  +--ro name?                   string
            |  |  |  +--ro type?                   identityref
            |  |  |  +--ro identifier?             uint16
      ..............................................................
```

```
            |  |  |  +--ro oper-status?            identityref
      /* TE tunnel primary path and LSP container */
            |  |  +--rw p2p-primary-paths
            |  |  |  +--rw p2p-primary-path* [name]
            |  |  |     +--rw name
                     /* Configuration */
      leafref
            |  |  |     +--rw config
            |  |  |     |  +--rw name?                      string
            |  |  |     |  +--rw preference?                uint8
            |  |  |     |  +--rw path-setup-protocol?       identityref
            |  |  |     |  +--rw path-computation-method?   identityref
            |  |  |     |  +--rw path-computation-server?   inet:ip-
      address
            |  |  |     |  +--rw compute-only?              empty
            |  |  |     |  +--rw use-cspf?                  boolean
            |  |  |     |  +--rw verbatim?                  empty
            |  |  |     |  +--rw lockdown?                  empty
            |  |  |     |  +--rw named-explicit-path?       leafref
            |  |  |     |  +--rw named-path-constraint?     leafref {te-
      types:named-path-constraints}?
                      /* state */
            |  |  |     +--ro state
            |  |  |     |  +--ro name?                       string
            |  |  |     |  +--ro preference?                 uint8
            |  |  |     |  +--ro path-setup-protocol?        identityref
            |  |  |     |  +--ro path-computation-method?    identityref
            |  |  |     |  +--ro path-computation-server?    inet:ip-
      address
            |  |  |     |  +--ro compute-only?               empty
            |  |  |     |  +--ro use-cspf?                   boolean
            |  |  |     |  +--ro verbatim?                   empty
            |  |  |     |  +--ro lockdown?                   empty
            |  |  |     |  +--ro named-explicit-path?        leafref
            |  |  |     |  +--ro named-path-constraint?      leafref
      {te-types:named-path-constraints}?
                        /* Computed path */
                        /* Computed path properties/metrics /
            |  |  |     |  +--ro computed-path-properties
            |  |  |     |  |  +--ro path-metric* [metric-type]
            |  |  |     |  |  |  +--ro metric-type           identityref
            |  |  |     |  |  |  +--ro accumulative-value?   uint64
                        /* Computed path affinities */
            |  |  |     |  |  +--ro path-affinities
            |  |  |     |  |  |  +--ro constraints* [usage]
            |  |  |     |  |  |     |  +--ro usage?
      identityref
```

```
            |  |  |     |  |  |     |  +--ro (style)?
            |  |  |     |  |  |     |     +--:(value)
            |  |  |     |  |  |     |     |  +--ro value?            te-
      types:admin-groups
            |  |  |     |  |  |     |     +--:(named)
            |  |  |     |  |  |     |        +--ro affinity-names*
      [name]
            |  |  |     |  |  |     |           +--ro name    string
                        /* Computed path SRLGs */
            |  |  |     |  |  +--ro path-srlgs
            |  |  |     |  |  |  +--ro (style)?
            |  |  |     |  |  |     +--:(values)
            |  |  |     |  |  |     |  +--ro usage?         identityref
            |  |  |     |  |  |     |  +--ro values*        te-
      types:srlg
            |  |  |     |  |  |     +--:(named)
            |  |  |     |  |  |        +--ro constraints* [usage]
            |  |  |     |  |  |           +--ro usage
      identityref
            |  |  |     |  |  |           +--ro constraint
            |  |  |     |  |  |              +--ro srlg-names* [name]
            |  |  |     |  |  |                 +--ro name    string
                        /* Computed path sub-objects */
            |  |  |     |  |  +--ro path-computed-route-objects
      ..............................................................
                        /* LSP (provisioned path) */
            |  |  |     |     +--ro lsp* [source destination tunnel-id
      lsp-id extended-tunnel-id type]
                        /* LSP parameters */
            |  |  |     |        +--ro source                leafref
            |  |  |     |        +--ro destination           leafref
            |  |  |     |        +--ro tunnel-id             leafref
            |  |  |     |        +--ro lsp-id                leafref
            |  |  |     |        +--ro extended-tunnel-id    leafref
            |  |  |     |        +--ro type                  leafref
            |  |  |     |        +--ro signaling-type?       identityref
            |  |  |     +--rw candidate-p2p-secondary-paths
            |  |  |        +--rw candidate-p2p-secondary-path*
      [secondary-path]
            |  |  |           +--rw secondary-path    leafref
            |  |  |           +--rw config
            |  |  |           |  +--rw secondary-path?        leafref
            |  |  |           |  +--rw priority?              uint16
            |  |  |           |  +--rw path-setup-protocol?
      identityref
            |  |  |           +--ro state
            |  |  |              +--ro secondary-path?        leafref
```

```
             |  |  |              +--ro priority?              uint16
             |  |  |              +--ro path-setup-protocol?
      identityref
             |  |  |              +--ro active?                boolean

      /* TE tunnel secondary path and LSP container */

             |  |  +--rw p2p-secondary-paths
             |  |  |  +--rw p2p-secondary-path* [name]
      .......................................................
             |  |  |     +--rw name      leafref
             |  |  |     +--rw config (same as for primary path )
      ......................................................
             |  |  |     +--ro state  (same as for primary, except for
      disjointedness_state )
          |        |  +--ro disjointness_state?        te-types:te-path-
      disjointness......................................................
             |  |  |        +--ro computed-path-properties (same as for
      primary path)
      ...........................................................
           |  |  |        |  +--ro path-affinities (same as for primary
      path)
      ...........................................................
             |  |  |        |  +--ro path-srlgs    (same as for primary
      path)
      ...........................................................
             |  |  |        |  +--ro path-computed-route-objects
      ..........................................................
                            /* LSP (provisioned path) */
             |  |  |        +--ro lsp  (same as for the primary LSP)
      .........................................................
```

- o Tunnel termination point (TTP) - a physical device inside a given node/switch realizing a TE tunnel termination function in a given layer network, as well as the TE tunnel's adaptation function provided for client layer network(s). One example of tunnel termination point is an OCh layer transponder. [Note: Tunnel termination points are not to be confused with TE tunnel termination points, which are TE representations of physical tunnel termination points. Similar to physical switches and links of the network, such as depicted in Figure 20, being represented on a TE topology describing the network as TE nodes and TE links, (physical) tunnel termination points (TTPs) are represented as TE tunnel termination points (TE TTPs, see 1.2) hosted by the TE nodes. For example, a provisioned connection/LSP starts on a source TTP, goes through a chain of physical links and stops on a destination TTP. In contrast, TE path (e.g. result of a path computation) starts on a source TE TTP, goes through a chain of TE links and stops on a destination TE TTP.]

```
         |  |  |  +--rw source?               inet:ip-address
         |  |  |  +--rw destination?          inet:ip-address
         |  |  |  +--rw src-tp-id?            binary
         |  |  |  +--rw dst-tp-id?            binary
```

- o TE tunnel hand-off point - an access link or inter-domain link by which a multi-domain TE tunnel enters or exits a given network domain, in conjunction with a layer network resource (such as a wavelength channel or ODUk container) allocated on the access/inter-domain link for the TE tunnel.

- o TE tunnel segment - a part of a multi-domain TE tunnel that spans a given network domain and is directly and fully controlled by the domain's controller, DC. TE tunnel segment is a fragment of a multi-domain TE tunnel between

  1. the source tunnel termination point and the TE tunnel hand-off point outbound from the TE tunnel's first domain (head TE tunnel segment);

  2. inbound and outbound TE tunnel hand-off points into/from a given domain (transit TE tunnel segment);

3. inbound TE tunnel hand-off point into the TE tunnel's last domain and the destination tunnel termination point (tail TE tunnel segment);

- Transport service - the same as TE tunnel segment

- Hierarchy TE tunnel - a server layer TE tunnel that supports a dynamically created TE link in the client layer network topology (e.g. see 1.2)

```
/* Hierarchy TE tunnel parameters */
      |  |  |  +--rw hierarchical-link-id
      |  |  |  |  +--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-id?        te-types:te-
topology-id
```

- Hierarchy transport service - the first or the last segment of a multi-domain hierarchy TE tunnel

- Dependency TE tunnel - a hierarchical TE tunnel provisioned or to be provisioned in an immediayely adjacent server layer a given client layer TE tunnel depends on (i.e. carried or to be carried within)

- Potential TE tunnel/segment - a TE tunnel/segment configured in COMPUTE_ONLY mode. For such a TE tunnel/segment TE paths to be taken by supporting connection(s) is/are computed and monitored, but the connection(s) are not provisioned

```
      |  |  |     |  +--rw path-computation-method?   identityref
      |  |  |     |  +--rw path-computation-server?   inet:ip-
address
      |  |  |     |  +--rw compute-only?             empty
      |  |  |     |  +--rw use-cspf?                 Boolean
```

Figure 20a. TE Tunnel Connections/LSPs

o Layer network connection/connection/LSP - a layer network path supporting a TE tunnel by realizing its implied forwarding function. Said path is provisioned in a given layer network’s data plane over a chain of links and cross-connected over switches terminating the links. It interconnects the supported TE tunnel’s source and destination termination points (in the case of end-to-end connection) or TE tunnel’s hand-off points (in the case of transport service connection) or the TE tunnel’s two split-merge points (in the case of segment protection connection.

   Example: ODU2 connection supporting an ODU2 TE tunnel.

```
                        /* LSP (provisioned path) */
            |  |  |     |     +--ro lsp* [source destination tunnel-id
      lsp-id extended-tunnel-id type]
                        /* LSP parameters */
            |  |  |     |        +--ro source                leafref
            |  |  |     |        +--ro destination           leafref
            |  |  |     |        +--ro tunnel-id             leafref
            |  |  |     |        +--ro lsp-id                leafref
            |  |  |     |        +--ro extended-tunnel-id    leafref
            |  |  |     |        +--ro type                  leafref
            |  |  |     |        +--ro signaling-type?       identityref
      ..................................................................
            |  |  |              +--ro priority?              uint16
            |  |  |              +--ro path-setup-protocol?
      identityref
            |  |  |              +--ro active?                Boolean
```

- Working connection - the primary connection of the supported TE tunnel or transport service (see Figure 20a).

- End-to-end protection connection - a secondary end-to-end connection of the supported TE tunnel (e.g. end-to-end 1+1 protection connection, see Figure 20a).

- Segment protection connection - a secondary connection of the supported transport service protecting the service over a given network domain (e.g. 1+1 segment protection connection, see Figure 20a)

- Restored connection - a connection after successful network failure restorationrestoration procedures

- Current connection - the same as restored connection

- Nominal connection - a connection as (re-)provisioned upon a client configuration request (i.e. a connection before any automatic network failure restoration re-configurations are carroed out, also a connection after restoration reversion procedures are successfully completed)

- Unprotected TE tunnel/transport service - TE tunnel/transport service supported by a single (working/primary) connection/LSP

- Protected TE tunnel/transport service - TE tunnel/transport service supported by one working connection/LSP and at least one protection/secondary connection/LSP

- Restorable TE tunnel/transport service - TE tunnel/transport service with pre-configured automatic network failure restoration capabilities

- TE tunnel/transport service automatic protection switchover - a process of switching of carrying user payload from the tunnel’s/service’s affected by a network failure working connection onto one of the tunnel’s/service’s healthy protection connection

- TE tunnel/transport service automatic protection reversion - a process of switching of carrying user payload from the tunnel’s/service’s protection connection back onto the tunnel’s/service’s working connection after the latter was repaired from network failure

- TE tunnel/transport service protection Hold-off time - a configured period of time to expire between the moment of detecting of the first network failure affecting the tunnel’s/service’s working connection and the begining of the tunnel’s/service’s automatic protection switchover procedures

- TE tunnel/transport service protection WTR time - a configured period of time to expire between the moment of repairing the last network failure affecting the tunnel’s/service’s working connection and the begining of the tunnel’s/service’s automatic protection reversion procedures

- TE tunnel/transport service automatic network failure restoration - a process of replacing of the tunnel’s/service’s connection(s) affected by one or more network failures away from the point(s) of failue

- TE tunnel/transport service restoration reversion- a process of replacing of the tunnel’s/service’s connection(s) back onto the nominal connection paths after all network failures affecting the tunnel’s/service’s nominal connection(s) are repaired

- TE tunnel/transport service restoration Hold-off time - a configured period of time to expire between the moment of detecting of the first network failure affecting the tunnel’s/service’s nominal or current connection and the beginning of the automatic connection restoration procedures

- TE tunnel/transport service restoration WTR time - a configured period of time to expire between the moment of repairing the last network failure affecting the tunnel’s/service’s nominal connection and the begining of the connection automatic restoration reversion procedures

- Configured restoration path - a TE path specified by the client to be used during the automatic network failure restoration operation on one of the TE tunnel’s/transport service’s nominal or current connections

- Pre-computed restoration path - a configured restoration path to be validated by a path computer during the TE tunnel/transport service setup or client triggered modification

- Pre-provisioned restoration path - a pre-computed restoration path to be pre-provisioned/pre-signaled in the network (with all associated network resources allocated but not necessarily bound into cross-connects) during the TE tunnel/transport service setup or client triggered modification

- Connection configured path - a TE path (see 1.2) over a TE topology describing a layer network/domain that specifies (loosely or strictly) the client’s requirements with respect to an ordered list of network nodes, links and resources on the links a given connection should go through

---

```
      |  |        +--rw explicit-route-object* [index]
            |  |           +--rw index                   leafref
            |  |           +--rw explicit-route-usage?   identityref
      (INCLUDE/EXCLUDE)
            |  |           |  +--rw index?            uint32
            |  |           |  +--rw (type)?
            |  |           |     +--:(numbered)
            |  |           |     |  +--rw numbered-hop
            |  |           |     |     +--rw address?    te-types:te-tp-
      id
            |  |           |     |     +--rw hop-type?   te-hop-type
            |  |           |     +--:(as-number)
            |  |           |     |  +--rw as-number-hop
            |  |           |     |     +--rw as-number?   binary
            |  |           |     |     +--rw hop-type?    te-hop-type
            |  |           |     +--:(unnumbered)
            |  |           |     |  +--rw unnumbered-hop
            |  |           |     |     +--rw node-id?      te-types:te-
      node-id
            |  |           |     |     +--rw link-tp-id?   te-types:te-
      tp-id
            |  |           |     |     +--rw hop-type?     te-hop-type
            |  |           |     +--:(label)
            |  |           |     |  +--rw label-hop
            |  |           |     |     +--rw value?   rt-
      types:generalized-label
            |  |           |     +--:(sid)
            |  |           |        +--rw sid-hop
            |  |           |           +--rw sid?   rt-
      types:generalized-label
```

o Connection exclusion path - a TE path over a TE topology describing a layer network/domain that specifies the client’s requirements with respect to an unordered list of network nodes, links and resources on the links to be avoided by a given connection

```
|        |  +--rw route-object-exclude-always* [index]
      |        |  |  +--rw index     leafref
      |        |  |  |  +--rw index?            uint32
      |        |  |  |  +--rw (type)?
      |        |  |  |     +--:(numbered)
      |        |  |  |     |  +--rw numbered-hop
      |        |  |  |     |     +--rw address?    te-types:te-tp-
id
      |        |  |  |     +--:(as-number)
      |        |  |  |     |  +--rw as-number-hop
      |        |  |  |     |     +--rw as-number?   binary
      |        |  |  |     +--:(unnumbered)
      |        |  |  |     |  +--rw unnumbered-hop
      |        |  |  |     |     +--rw node-id?      te-types:te-
node-id
      |        |  |  |     |     +--rw link-tp-id?   te-types:te-
tp-id
      |        |  |  |     +--:(label)
      |        |  |  |     |  +--rw label-hop
      |        |  |  |     |     +--rw value?   rt-
types:generalized-label
```

```
            |        |  |  |     +--:(sid)
            |        |  |  |        +--rw sid-hop
            |        |  |  |           +--rw sid?   rt-
      types:generalized-label
```

o Connection computed path - a TE path over a TE topology describing a layer network/domain as computed (subject to all configured constraints and optimization criteria) for a given connection to take. Computed connection path could be thought as the TE path intended to be taken by the connection

```
       /* Computed path */
                        /* Computed path properties/metrics /
            |  |  |     |  +--ro computed-path-properties
            |  |  |     |  |  +--ro path-metric* [metric-type]
            |  |  |     |  |  |  +--ro metric-type           identityref
            |  |  |     |  |  |  +--ro accumulative-value?   uint64
                        /* Computed path affinities */
            |  |  |     |  |  +--ro path-affinities
            |  |  |     |  |  |  +--ro constraints* [usage]
            |  |  |     |  |  |     |  +--ro usage?
      identityref
            |  |  |     |  |  |     |  +--ro (style)?
            |  |  |     |  |  |     |     +--:(value)
            |  |  |     |  |  |     |     |  +--ro value?            te-
      types:admin-groups
            |  |  |     |  |  |     |     +--:(named)
            |  |  |     |  |  |     |        +--ro affinity-names*
      [name]
            |  |  |     |  |  |     |           +--ro name    string
                        /* Computed path SRLGs */
            |  |  |     |  |  +--ro path-srlgs
            |  |  |     |  |  |  +--ro (style)?
            |  |  |     |  |  |     +--:(values)
            |  |  |     |  |  |     |  +--ro usage?         identityref
            |  |  |     |  |  |     |  +--ro values*        te-
      types:srlg
            |  |  |     |  |  |     +--:(named)
            |  |  |     |  |  |        +--ro constraints* [usage]
            |  |  |     |  |  |           +--ro usage
      identityref
```

```
            |  |  |     |  |  |           +--ro constraint
            |  |  |     |  |  |              +--ro srlg-names* [name]
            |  |  |     |  |  |                 +--ro name    string
                        /* Computed path sub-objects */
            |  |  |     |  |  +--ro path-computed-route-objects
      ..............................................................
```

- o Connection actual path - an active connection’s path as provisioned in the layer network’s data plane in the form of a TE path over a TE topology describing the layer network/domain

1.8. Transport Service Mapping

Figure 21. Transport Service Mapping

Let’s assume that a provider has exposed to a client its network domain in the form of an abstract TE topology, as shown on the left side of Figure 21. From then on, the provider should be prepared to receive from the client, a request to set up or manipulate a transport service with TE path(s) computed for the service connection(s) based on and expressed in terms of the provided abstract TE topology (as, for example, displayed in red broken line on the right side of Figure 21). When this happens, the provider is expected to set up the TE tunnels supporting all yet uncommitted abstract TE links (e. g, TE link S3’-S8’ in the Figure).

Furthermore, it is the responsibility of the provider to:

- o Perform all the necessary abstract-to-native translations for the specified TE paths (i.e. the transport service connection configured paths);

- Provision working and protection connections supporting the transport service; as well as replace/modify/delete them in accordance with subsequent client’s configuration requests;

- Perform all the requested recovery operations upon detecting network failures affecting the transport service;

- Notify the client about all parameter changes, events and other telemetry information the client has expressed an interest in, with respect to the transport service in question.

1.9. Multi-Domain Transport Service Coordination

A client of multiple TE network domains may need to orchestrate/coordinate its transport service setup/manipulation across some or all the domains. One example of such a client is a Hierarchical T-SDN Controller, HC, managing a connected multi-domain transport network where each of the domains is controlled by a separate Domain T-SDN Controller, DC. Said DCs are expected to expose TE Topology and TE Tunnel North Bound Interfaces, NBIs,, supported respectively by IETF TE Topology and TE Tunnel models (and their network layer specific augmentations). HC is assumed to establish client-provider relationship with each of the DCs and make use of said NBIs to extract from the domains various information (such as TE topologies and telemetry), as well as to convey instructions to coordinate across multiple domains its transport services set up and manipulation.

Figure 22. Two-Domain Transport Network

Let’s consider, for example, a two-domain transport network as represented in Figure 22. Suppose that HC is requested to set up an unprotected transport service to provide connectivity between customer network elements C-R1 and C-R6. It is assumed that by the time the request has arrived, the two DCs have already provided abstract TE topologies describing their respective domains, and that HC has merged the provided TE topologies into one that homogeneously describes the entire transport network (as shown in Figure 23).

Figure 23. Two-Domain Transport Network (Abstracted View)

Consider that HC, using the merged TE topology, selected a TE path to be taken by the requested transport service connection as shown on the upper part of Figure 24.

The multi-domain transport service set up coordination includes:

o Splitting selected for the transport service TE path(s) into segments - one set of segments per each domain involved in the service setup;

o Issuing a configuration request to each of the involved DCs to set up the transport service across the respective domain. Note that the connection configured paths are required to be expressed in terms of respective abstract TE topologies as exposed to HC by DCs (see lower part of Figure 24).

- o Waiting for the set up complete confirmation from each of the involved DCs. In case one of the DCs reports a failure, HC is responsible to carry out the cleanup/rollback procedures by requesting all involved DCs to tear down the successfully created segments

Figure 24. Transport Service Placement Based on Abstract TE Topology

While processing the received from HC configuration request to set up the transport service, each DC is expected to carry out the transport service mapping procedures (as described in 1.8) resulting in the set up of all the necessary underlay TE tunnels, as well as one or more connections supporting the transport service. As a result, the requested transport service will be provisioned as shown in Figure 25.

The multi-domain transport service tear down coordination entails issuing to each of the involved DCs a configuration request to delete the transport service in the controlled by the DC domain. DCs are expected in this case to release all network resources allocated for the transport service.

The multi-domain transport service modify coordination implies issuing to each of the involved DCs a configuration request to replace the transport service connections according to the newly provided paths and/or modify the connection parameters according to the newly provided configuration.

Figure 25. Multi-domain transport service is provisioned

2. Use Cases

2.1. Use Case 1. Transport service control on a single layer multi-domain transport network

Configuration (Figure 26):

- Three-domain multi-vendor ODUk/Och transport network;

- The domains are interconnected via ODUk inter-domain links;

- Each of the domains is comprised of ODUk/Och network elements (switches) from a separate vendor and is controlled by a single (vendor specific) T-SDN Domain Controller (DC);

- All DCs expose IETF TE Topology and TE Tunnel model based NBIs;

- The transport network as a whole is controlled by a single hierarchical T-SDN controller (HC);

- HC makes use of the NBIs to set up client-provider relationship with each of the DCs and controls via the DCs their respective network domains;

- Three customer IP/MPLS sites are connected to the transport network via ODUk access links;

- The customer IP/MPLS routers and the router transport ports connecting the routers to the transport network are managed autonomously and independently from the transport network.

Figure 26 Three-domain ODUk/Och transport network with ODUk access and inter-domain links

Objective: Set up/manipulate/delete a shortest delay unprotected or protected transport service to provide connectivity between customer network elements C-R2 and C-R5

1) TE Topology discovery

All DCs provide to HC respective domain ODUk layer abstract TE topologies. Let’s assume that each such topology is a single-node TE topology (as described in 1.3.1, abstract TE topology of this type represents the entire domain as a single asymmetrical/blocking TE node). Let’s further assume that the abstract TE nodes representing the domains are attributed with detailed connectivity matrices optimized according to the shortest delay criterion. [Note: single-node abstract TE topologies are assumed for simplicity sake. Alternatively, any DC could have provided an abstract TE topology of any type described in 1.3].

HC merges the provided TE topologies into its own native TE topology (the TE topology merging procedures are discussed in 1.4). The merged TE topology, as well as the TE topologies provided by DCs, are depicted in Figure 27. The merged TE topology homogeneously describes the entire transport network and hence is suitable for path computations across the network. Note that the dotted lines in the Figure connecting the topology access TE links with customer devices illustrate that HC in this use case has neither control nor information on the customer devices/ports and, therefore, can only provide a connectivity between the requested transport service ingress and egress access links (on assumption that the customer transport ports are provisioned independently)

Figure 27. Three-domain single layer transport network abstract TE topology

2) Transport service path computation

Using the merged TE topology (Figure 27, upper part) HC selects one or more optimal and sufficiently disjoint from each other TE path(s) for the requested transport service connection(s). Resulting TE paths for the requested end-to-end protected transport service, for example, could be as marked on the upper part of Figure 28.

It is important to keep in mind that HC’s path computer is capable of performing the necessary path selection only as long as the merged TE topology provides the necessary TE visibility for the path selection, both intra-domain (e.g. by virtue of provided by the abstract TE nodes detailed connectivity matrices) and inter-domain (because of provided inter-domain TE link attributes). In case one or more DCs is/are not capable of or willing to provide the detailed connectivity matrices (that is, DCs expose the respective domains as black boxes - unconstrained TE nodes terminating the inter-domain TE links), HC will not be able to select the end-to-end TE path(s) for the requested transport service on its own. In such a case HC may opt for making use of the Path Computation NBI, exposed by the DCs to explore/evaluate intra-domain TE path availability in real time. IETF TE Tunnel model supports the Path Computation NBI by allowing for the configuration of transport services in COMPUTE_ONLY mode. In this mode the provider is expected to compute TE paths for a requested transport service connections and return the paths in the request’s response without triggering the connection provisioning in the network.

Consider, for example, the case when none of the DCs has provided the detailed connectivity matrix attribute for the abstract TE nodes representing the respective domain. In such a case HC may:

1. Request the ingress domain DC (i.e. DC1) to compute intra-domain TE paths connecting the ingress access TE link (i.e. the link facing C-R2) with each of the inter-domain TE links (i.e. links connecting Domain 1 to Domain 2 and Domain 3 respectively);

2. Grow the TE paths returned by DC1 in (1) over the respective outbound inter-domain TE links;

3. Request the neighboring DC(s) (e.g. DC3) to compute all intra-domain TE paths connecting across the domain all inbound into the domain inter-domain TE links reached by the path growing process in (2) with all other (outbound) domain’s inter-domain TE links;

4. Augment the TE paths produced in step (2) with the TE paths determined in step (3);

5. Repeat steps (2), (3) and (4) until the resulting TE paths reach the egress domain (i.e. Domain 2);

6. Request the egress domain DC (i.e. DC2) to grow each of the TE paths across the domain to connect them to the egress access TE link (i.e. the link facing C-R5);

7. Select one (or more) most optimal and sufficiently disjoint from each other TE path(s) from the list produced in step (6).

[Note: The transport service path selection method based on Path Computation NBIs exposed by DCs does not scale well and the more domains comprise the network and the more inter-domain links interconnect them, the worse the method works. Realistically, this approach will not work sufficiently well for the networks with more than 3 domains]

Figure 28. TE paths computed for the protected transport service

3) Transport service setup coordination

HC carries out the multi-domain transport service setup coordination as described in 1.9. In particular, HC splits the computed TE path(s) into 3 sets of TE path segments - one set per domain (as shown on the lower part of Figure 28), and issues a TE tunnel configuration request to each of the DCs to set up the requested transport service across the domain under the DC’s control. The primary (and secondary) connection explicit path(s) is/are specified in the requests in terms of respective domain abstract TE topologies.

While processing the configuration request, each DC performs the transport service mapping (as described in 1.8). In particular, the DC translates the specified explicit path(s) from abstract into native TE topology terms, sets up supporting underlay TE tunnels (e.g. Och TE tunnels), and, then, allocates required ODUk containers on the selected links and provisions the ODUk cross-connects on the switches terminating the links.

If the setup is successfully completed in all three domains, the transport service connection(s) will be provisioned as depicted in Figure 29. If one of the DCs fails to set up its part, all successfully provisioned segments will be asked by HC to be released.

4) Transport service teardown coordination

HC issues to each of DCs a configuration request to release the transport service over the controlled domain, as well as the server layer TE tunnels supporting dynamically created links.

Figure 29. Transport service is provisioned

2.2. Use Case 2. End-to-end TE tunnel control on a single layer multi-domain transport network

Configuration (Figure 26): the same as in use case 1, except that HC in this use case controls customer devices/ports by extracting information from and pushing configuration to the customer site SDN controller(s) managing the customer devices directly.

Objective: Set up//delete an unprotected shortest delay TE tunnel interconnecting end-to-end C-R2 and C-R5

1) TE Topology discovery

As in use case 1 all DCs provide to HC domain ODUk layer abstract TE topologies. Additionally in this use the three customer site controllers expose the TE Topology and Tunnel model based NBIs to HC. Using the TE Topology NBI each customer controller provides to HC the respective customer site domain abstract TE topology. Customer site abstract TE topologies contain abstract TE nodes representing the devices which are directly connected to the transport network. Said abstract TE nodes host TE tunnel termination points, TTPs, representing the ports over which the customer devices are connected to the transport network, and terminate access TE links the TTPs are accessible from (see Figure 30).

Figure 30. Abstract TE topologies provided by all network domains and customer sites

HC merges the provided topologies into its own native TE Topology (the TE topology merging procedures are discussed in 1.4). The merged TE topology is depicted in Figure 31. It homogeneously describes end-to-end not only the entire transport network, but also the customer sites connected to the network and hence is suitable for TE tunnel end to end path computations.

Figure 31. Abstract TE topology describing transport network and connected to it customer sites

2) TE tunnel path computation

Using the merged TE topology (Figure 31) HC selects an optimal TE path for the requested TE tunnel connecting end-to-end the specified TE tunnel termination points, TTPs. The resulting TE path, for example, could be as marked on the upper part of Figure 32.

Figure 32. TE path computed for the TE tunnel

3) TE tunnel setup coordination

HC carries out the multi-domain TE tunnel setup coordination as described for use case 1, except that in this use case HC additionally initiates and controls the setup of the TE tunnel's head and tail segments on the respective customer sites. Note that the customer site controllers behave exactly as transport network domain DCs. In particular, they receive issued by HC configuration requests to set up the TE tunnel's head and tail segments respectively. While processing the requests the customer site controllers perform the necessary provisioning of the TE tunnel's source and destination termination points, as well as of the local sides of the selected access links. If all segments are successfully provisioned on customer sites and network domains, the TE tunnel connection will be provisioned as marked in Figure 33.

4) TE tunnel teardown coordination

HC issues to each of DCs and customer site controllers a configuration request to release respective segments of the TE tunnel, as well as the server layer TE tunnels supporting dynamically created links.

Figure 33. TE tunnel is provisioned

2.3. Use Case 3. Transport service control on a ODUk/Och multi-domain transport network with Ethernet access links

Configuration (Figure 34): the same as in use case 1, except that all access links in this use case are Ethernet layer links (depicted as blue lines in the Figure), while all inter-domain links remain to be ODUk layer links.

Figure 34. Three-domain ODUk/Och transport network with Ethernet layer access links

Objective: Set up//delete an unprotected shortest delay transport service supporting connectivity between C-R2 and C-R5

1) TE Topology discovery

In order to make possible for the necessary in this use case multi-layer path computation, each DC exposes to HC two (ODUk layer and Ethernet layer) abstract TE topologies, Additionally, the lower layer (ODUk) TE nodes announce hosted by them TE tunnel termination points, TTPs, capable of adopting the payload carried over the Ethernet layer access links, From the TE Topology model point of view this means that said TTPs are attributed with TE inter-layer locks matching ones attributed to Ethernet TE links (i.e. TE links provided within Ethernet layer abstract TE topologies).

Ethernet and ODUk layer single node abstract TE topologies catered to HC by each of the DCs are presented in Figure 35.

HC merges the provided TE topologies into its own native TE Topology (the merging procedures are described in 1.4). Importantly in this case HC locks the provided TE topologies not only horizontally, but vertically as well, thus producing a two-layer TE topology homogenously describing both layers of the entire transport network, as well as the client-server layer adaptation relationships between the two layers. This makes the merged TE topology suitable for multi-layer/inter-layer multi-domain transport service path computations. The merged TE topology is presented in Figure 36.

Figure 35. ODUk and Ethernet layer abstract TE topologies exposed by DCs

Figure 36. Two-layer three-domain transport network abstract TE topology

2) Transport service path computation

Using the merged TE topology (Figure 36) HC selects an optimal TE path for the requested transport service.

Note that if HC’s path computer considered only Ethernet layer TE nodes and links, the path computation would .fail. This is because the Ethernet layer TE nodes (i.e. D1-e, D2-e and D3-e in the Figure) are disconnected from each other. However, the inter-layer associations (in the form of the TE inter-layer locks) make possible for the path computer to select TE path(s) in the lower (ODUk) layer that can be used to set up hierarchy TE tunnel(s) supporting additional dynamic TE link(s) in the upper (Ethernet) layer in order for the requested transport service path computation to succeed.

Let’s sssume that the resulting TE path is as marked in Figure 37. The red line in the Figure marks the TE path selected for the ODUk layer hierarchy TE tunnel supporting the required Ethernet layer dynamic TE link.

Figure 37. Multi-layer TE path computed for the transport service

3) Transport service setup coordination

HC sets up the requested Ethernet layer transport service in two stages. First, it coordinates the end-to-end setup of the ODUk layer hierarchy TE tunnel between the selected TTPs. If this operation succeeds, a new Ethernet layer dynamic TE link (blue line connecting TE nodes D1-e and D2-e in Figure 38) is automatically added to the merged abstract TE topology. Importantly, as a part of the hierarchy transport service setup both DC1 and DC 2 add a new open-ended Ethernet layer inter-domain dynamic TE link to their respective abstract TE topologies. Second, HC coordinates the setup of the requested (Ethernet layer) transport service. The required TE path for the second stage is marked as fat blue line in the Figure. Note that DC3 controlling domain 3 is only involved in the first stage, but is oblivious to the second stage.

Figure 38. A new Ethernet layer TE link supported by ODUk layer TE tunnel is added to the provided and merged abstract TE topologies

IF all involved DCs confirm successful setup completion, the requested transport service, as well as the supporting server layer hierarchy TE tunnel, will be provisioned as depicted in Figure 39. If one of the DCs fails to set up its segment in either of the layers, all successfully provisioned segments will be requested by HC to be released.

Figure 39. Ethernet transport service and supporting ODUk TE tunnel are provisioned

4) Transport service teardown coordination

First, HC issues to DC1 and DC2 a configuration request to release the Ethernet layer transport service in the respective domains. After that, all three DCs are requested to release the segments of the supporting ODUk layer hierarchy TE tunnel. While processing the request DC1 and DC2 also remove the dynamic Ethernet layer TE links supported by the respective hierarchy TE tunnel’s segments, thus the network’s abstract TE topologies are reverted back to the state as shown in Figures 35 and 36.

2.4. Use Case 4. Transport service control on a ODUk/Och multi-domain transport network with multi-function access links

Configuration (Figure 40): the same as in use case 3, except that all access links in this use case are multi-function links (depicted in the Figure as blue compound lines). Let’s assume that, depending on configuration, the multi-function access links in this use case can carry either Ethernet or SDH/STM16 layer payload.

Objective: Set up//delete an unprotected shortest delay SDH/STM16 layer transport service interconnecting C-R2 and C-R5

Figure 40. Three-domain ODUk/Och transport network with multi-function access links

1) TE Topology discovery

The TE Topology model considers multi-function links as parallel mutually exclusive TE links each belonging to a separate layer network. For this use case each DC exposes to HC three (ODUk-, Ethernet- and SDH/STM16-layer) abstract TE topologies (generally speaking, one abstract TE topology per each layer network supported by at least one access or inter-domain link). Like in use case 3, the lower layer (ODUk) TE nodes announce hosted by them TE tunnel termination points, TTPs, capable in this case of adopting Ethernet, SDH/STM16 or both layer payloads, The TTPs are attributed with TE inter-layer locks matching ones specified for Ethernet and/or SDH/STM16 TE links.

Ethernet, SDH/STM16 and ODUk layer single-node abstract TE topologies catered to HC by each of the DCs are presented in Figure 41.

HC merges the provided topologies into its own native TE Topology (the merging procedures are described in 1.4). As in use case 3 HC locks the provided TE topologies not only horizontally (i.e. between domains), but vertically (between layers) as well, thus producing a three-layer TE topology homogenously describing the three layers of the entire transport network, as well as the client-server layer adaptation relationships between the layers. This makes the merged TE topology suitable for multi-layer/inter-layer multi-domain transport service path computations. The merged TE topology is presented in Figure 42.

Figure 41. ODUk, Ethernet and SDH/STM16 layer abstract TE topologies exposed by DCs

Figure 42. Three-layer three-domain transport network abstract TE topology

2) Transport service path computation

Using the merged TE topology (Figure 42) HC’s path computer selects a TE path for the requested transport service. For example, for the SDH/STM16 layer unprotected transport service the resulting TE path could be determined as marked in Figure 43.

Figure 43. Multi-layer TE path computed for SDH/STM16 layer transport service

3) Transport service setup coordination

Same as in use case 3.

4) Transport service teardown coordination

Same as in use case 3.

2.5. Use Case 5. Real time updates of IP/MPLS layer TE link attributes that depend on supporting transport connectivity (e.g. transport SRLGs, propagation delay, etc.)

Configuration (Figure 26): the same as in use case 1,

Objective: A transport service interconnecting transport ports of two IP routers across a transport network is likely to serve a link in IP/MPLS layer network, which is usually controlled by a client of the transport network, such as IP/MPLS Controller. Performance of TE applications (e.g. path computer) running on the IP/MPLS Controller depends on the accuracy of IP/MPLS layer TE link attributes. Some of these attributes can change over time and are known real-time only to a transport network controller, such as HC. Examples of said attributes are transport SRLGs, propagation delay metric, protection capacities and status, etc. The objective of this use case is to ensure up-to-date state of said attributes in the IP/MPLS Controller’s internal TED via necessary updates provided in a timely manner by the controller (e.g. HC) managing transport connectivity supporting IP/MPLS layer links.

Realization:

o HC exposes and supports IETF TE Topology and TE Tunnel model based NBIs (the same NBIs that are exposed by DCs serving HC);

o IP/MPLS Controller makes use of the exposed NBIs to set up the respective client-provider relationships with HC;

o IP/MPLS Controller uses the TE Tunnel NBI to configure with HC a transport service interconnecting transport ports of a pair of IP routers desired to be adjacent in the IP/MPLS layer network. The TE Tunnel model allows for specifying in the transport service configuration request the TE topology and link IDs of the IP/MPLS TE link the requested transport service will be serving;

o IP/MPLS Controller uses the TE Topology NBI to subscribe with HC on the IP/MPLS TE link notifications with respect to changes in the TE link’s attributes, such as SRLGs, propagation delay, protection capabilities/status, etc.;

o HC uses the TE Topology NBI to convey the requested notifications when HC learns the attributes IP/MPLS has expressed interest in or detects any changes since previous notifications (for example, due to network failure restoration/reversion procedures happened to the transport connectivity that supports the failure affected IP/MPLS links)

2.6. Use Case 6. Virtual Network Service

Configuration (Figure 26): the same as in use case 1,

Objective: Set up two Virtual Networks for the client, with Virtual Network 1 interconnecting customer IP routers C-R1, C-R7 and C-R4 over a single-node abstract TE topology, and Virtual Network 2 interconnecting customer IP routers C-R2, C-R3, C-R8, C-R5 and C-R6 over a full mesh link abstract TE topology as depicted in Figure 44.

[Note: A client of a transport network may want to limit the transport network connectivity of a particular type and quality within distinct subsets of its network elements interconnected across the transport network. Furthermore, a given transport network may serve more than one client. In this case some or all clients may want to ensure the availability of transport network resources in case dynamic (re-)connecting of their network elements across the transport network is envisioned. In all such cases a client may want to set up one or more Virtual Networks over provided transport network]

1) Virtual Network setup

From the client’s point of view a Virtual Network setup includes the following procedures:

- Identifying the Virtual Network membership - a subset of the client’s network elements/ports to be interconnected over the abstract TE topology configured for the Virtual Network. Note that from the transport network provider’s point of view this effectively determines the list of abstract TE topology’s open-ended access TE links;

- Deciding on the Virtual Network’s abstract TE topology type (e.g. single-node vs. link mesh), optimization criterion (e.g. shortest delay vs. smallest cost), bandwidth, link disjointedness, adaptation capabilities and other requirements/constraints, as well as, whether the TE tunnels supporting the abstract TE topology need to be pre-established or established on demand (i.e. when respective abstract TE topology elements are selected for a client transport service);

- Using the IETF TE Topology model based NBI exposed by the transport network controller (i.e. HC), configure the Virtual Network’s abstract TE topology. Let’s assume that in this use case the abstract TE topology for Virtual Network 1 is configured as a single-node abstract TE topology (see section 1.3.1) with the abstract TE node’s detailed connectivity matrix optimized according to the shortest delay criteria. Likewise, the abstract TE topology for Virtual Network 2 is configured as a full-mesh link abstract TE topology (see section 1.3.2) optimized according to the smallest cost criteria with each of the abstract TE links to be supported by pre-established end-to-end protected TE tunnels.

[Note: Virtual Network’s abstract TE topology (re-)configuration/negotiation process is no different from one that happens, for example, between HC and its providers, DCs, and is described in section 1.5]

Figure 44. Virtual Networks provided for a transport network client

2) Using Virtual Network

Recall that use case 1 was about setting up a transport service interconnecting customer network elements C-R2 and C-R5 across the transport network. With the Virtual Network 2 in place, the client could have used the Virtual Network’s TE topology to select a TE path for the service. The TE Tunnel model based NBI allows for the client to specify the Virtual Network’s TE topology ID, as well, as the selected TE path (for example, as marked in Figure 45) as a configured path attribute in the transport service configuration request to ensure that the intended transport network resources are used for the service.

Figure 45. Transport service TE path is selected on Virtual Network’s TE topology

3. Security Considerations

This document does not define networking protocols and data, hence are not directly responsible for security risks.

4. IANA Considerations

This document has no actions for IANA.

5. References

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-08 (work in progress), March 2017.

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

Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-06 (work in progress), March 2017.

5.2. Informative References

[RFC2702] Awduche, D., "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

6. Acknowledgments

TBD.

Appendix A.   Data Examples

This section contains examples of an instance data in the JSON encoding [RFC7951].

A.1. Use Case 1

In the use case described in Section 2.1. , there are three provider network domains, each of them is represented as an abstract TE topology. The JSON encoded example data configurations for the three domains are:

A.1.1. Domain 1

```
{
  "networks": {
    "network": [
      {
        "network-types": {
          "te-topology": {}
        },
        "network-id": "otn-domain1-abs",
        "provider-id": 201,
        "client-id": 300,
        "te-topology-id": "te-topology:otn-domain1-abs",
        "node": [
          {
            "node-id": "D1",
            "te-node-id": "2.0.1.1",
            "te": {
              "te-node-attributes": {
                "domain-id" : 1,
                "is-abstract": [null],
                "underlay-topology": "domain1-och",
                "connectivity-matrices": {
                  "is-allowed": true,
                  "path-constraints": {
                    "bandwidth-generic": {
                      "te-bandwidth": {
                        "otn": [
                          {
                            "rate-type": "odu1",
                            "counter": 2
```

```
                             }
                           ]
                         }
                       }
                     }
                     "connectivity-matrix": [
                       {
                         "id": 10302,
                         "from": "1-0-3",
                         "to": "1-2-0"
                       },
                       {
                         "id": 10203,
                         "from": "1-0-2",
                         "to": "1-3-0"
                       },
                       {
                         "id": 10311,
                         "from": "1-0-3",
                         "to": "1-11-0"
                       },
                       {
                         "id": 11103,
                         "from": "1-0-11",
                         "to": "1-3-0"
                       },
                       {
                         "id": 10903,
                         "from": "1-0-9",
                         "to": "1-3-0"
                       },
                       {
                         "id": 10309,
                         "from": "1-0-3",
                         "to": "1-9-0"
                       },
                       {
                         "id": 10910,
                         "from": "1-0-9",
                         "to": "1-10-0"
                       },
                       {
                         "id": 11009,
                         "from": "1-0-10",
                         "to": "1-9-0"
                       },
                       {
                         "id": 20910,
                         "from": "1-1-9",
                         "to": "1-10-0"
                       },
                       {
                         "id": 21009,
                         "from": "1-0-10",
                         "to": "1-9-1"
                       },
                       {
                         "id": 20911,
                         "from": "1-1-9",
                         "to": "1-11-0"
                       },
                       {
                         "id": 21109,
                         "from": "1-0-11",
                         "to": "1-9-1"
                       }
                     ]
                   }
                 }
               },
               "termination-point": [
                 {
                   "tp-id": "1-0-3",
                   "te-tp-id": 10003
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                         }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-3-0",
                   "te-tp-id": 10300
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                           "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-9",
                   "te-tp-id": 10009
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-9-0",
                   "te-tp-id": 10900
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-1-9",
                   "te-tp-id": 10109
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-9-1",
                   "te-tp-id": 10901
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-2",
                   "te-tp-id": 10002
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-2-0",
                   "te-tp-id": 10200
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-10",
                   "te-tp-id": 10010
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-10-0",
                   "te-tp-id": 11000
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-11",
                   "te-tp-id": 10011
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-11-0",
                   "te-tp-id": 11100
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-1-11",
                   "te-tp-id": 10111
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-11-1",
                   "te-tp-id": 11101
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 }
               ]
             }
           ]
         }
       ]
     }
   }
```

A.1.2. Domain 2

```
   {
     "networks": {
       "network": [
         {
           "network-types": {
             "te-topology": {}
           },
           "network-id": "otn-domain2-abs",
           "provider-id": 202,
           "client-id": 300,
           "te-topology-id": "te-topology:otn-domain2-abs",
           "node": [
             {
               "node-id": "D2",
               "te-node-id": "2.0.2.2",
               "te": {
                 "te-node-attributes": {
                   "is-abstract": [null],
                   "underlay-topology": "domain2-och",
                   "connectivity-matrices": {
                     "is-allowed": true,
                     "path-constraints": {
                       "bandwidth-generic": {
                         "te-bandwidth": {
                           "otn": [
                             {
                               "rate-type": "odu1",
                               "counter": 2
                             }
                           ]
                         }
                       }
                     }
                     "connectivity-matrix": [
                       {
                         "id": 12125,
                         "from": "1-0-21",
                         "to": "1-25-0"
                       },
                       {
                         "id": 12521,
                         "from": "1-0-25",
                         "to": "1-21-0"
                         },
                       {
                         "id": 12128,
                         "from": "1-0-21",
                         "to": "1-28-0"
                       },
                       {
                         "id": 12821,
                         "from": "1-0-28",
                         "to": "1-21-0"
                       },
                       {
                         "id": 12231,
                         "from": "1-0-22",
                         "to": "1-31-0"
                         },
                       {
                         "id": 13122,
                         "from": "1-0-31",
                         "to": "1-22-0"
                       },
                       {
                         "id": 22228,
                         "from": "1-1-22",
                         "to": "1-28-0"
                       },
                       {
                         "id": 22822,
                         "from": "1-0-28",
                         "to": "1-22-1"
                       },
                       {
                         "id": 12528,
                         "from": "1-0-25",
                         "to": "1-28-0"
                       },
                       {
                         "id": 12825,
                         "from": "1-0-28",
                         "to": "1-25-0"
                       }
                     ]
                   }
                 }
               },
               "termination-point": [
                 {
                   "tp-id": "1-0-21",
                   "te-tp-id": 10021
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-21-0",
                   "te-tp-id": 12100
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-22",
                   "te-tp-id": 10022
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-22-0",
                   "te-tp-id": 12200
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-1-22",
                   "te-tp-id": 10122
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-22-1",
                   "te-tp-id": 12201
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-25",
                   "te-tp-id": 10025
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-25-0",
                   "te-tp-id": 12500
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-1-25",
                   "te-tp-id": 10125
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-25-1",
                   "te-tp-id": 12501
                   "te": {
                     "interface-switching-capability": [
                       {
```

```
                   "switching-capability": "switching-otn",
                   "encoding": "lsp-encoding-oduk"
                 }
               ]
             }
           },
           {
             "tp-id": "1-0-28",
             "te-tp-id": 10028
             "te": {
               "interface-switching-capability": [
                 {
                   "switching-capability": "switching-otn",
                   "encoding": "lsp-encoding-oduk"
                 }
               ]
             }
           },
           {
             "tp-id": "1-28-0",
             "te-tp-id": 12800
             "te": {
               "interface-switching-capability": [
                 {
                   "switching-capability": "switching-otn",
                   "encoding": "lsp-encoding-oduk"
                 }
               ]
             }
           },
           {
             "tp-id": "1-0-31",
             "te-tp-id": 10031
             "te": {
               "interface-switching-capability": [
                 {
                   "switching-capability": "switching-otn",
                   "encoding": "lsp-encoding-oduk"
                 }
               ]
             }
```

```
                 },
                 {
                   "tp-id": "1-31-0",
                   "te-tp-id": 13100
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 }
               ]
             }
           ]
         }
       ]
     }
   }
```

A.1.3. Domain 3

```
   {
     "networks": {
       "network": [
         {
           "network-types": {
             "te-topology": {}
           },
           "network-id": "otn-domain3-abs",
           "provider-id": 203,
           "client-id": 300,
           "te-topology-id": "te-topology:otn-domain3-abs",
           "node": [
             {
               "node-id": "D3",
               "te-node-id": "2.0.3.3",
               "te": {
                 "te-node-attributes": {
                   "is-abstract": [null],
                   "underlay-topology": "domain3-och",
                   "connectivity-matrices": {
                     "is-allowed": true,
                     "path-constraints": {
                       "bandwidth-generic": {
                         "te-bandwidth": {
                           "otn": [
                             {
                               "rate-type": "odu1",
                               "counter": 2
                             }
                           ]
                         }
                       }
                     }
                     "connectivity-matrix": [
                       {
                         "id": 13638,
                         "from": "1-0-38",
                         "to": "1-38-0"
                       },
                       {
                         "id": 13836,
                         "from": "1-0-38",
                         "to": "1-36-0"
                       },
                       {
                         "id": 13639,
                         "from": "1-0-36",
                         "to": "1-39-0"
                       },
                       {
                         "id": 13936,
                         "from": "1-0-39",
                         "to": "1-36-0"
                       },
                       {
                         "id": 23636,
                         "from": "1-0-36",
                         "to": "1-36-1"
                       },
                       {
                         "id": 33636,
                         "from": "1-1-36",
                         "to": "1-36-0"
                       },
                       {
                         "id": 13739,
                         "from": "1-0-37",
                         "to": "1-39-0"
                       },
                       {
                         "id": 13937,
                         "from": "1-0-39",
                         "to": "1-37-0"
                       },
                       {
                         "id": 23737,
                         "from": "1-0-37",
                         "to": "1-37-1"
                       },
                       {
                         "id": 33737,
                         "from": "1-1-37",
                         "to": "1-37-0"
                       }
                     ]
                   }
                 }
               },
               "termination-point": [
                 {
                   "tp-id": "1-0-36",
                   "te-tp-id": 10036
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-36-0",
                   "te-tp-id": 13600
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-37",
                   "te-tp-id": 10037
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-37-0",
                   "te-tp-id": 13700
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-1-37",
                   "te-tp-id": 10137
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-37-1",
                   "te-tp-id": 13701
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-39",
                   "te-tp-id": 10039
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-39-0",
                   "te-tp-id": 13900
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-36",
                   "te-tp-id": 10036
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-36-0",
                   "te-tp-id": 13600
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-0-38",
                   "te-tp-id": 10038
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 },
                 {
                   "tp-id": "1-38-0",
                   "te-tp-id": 13800
                   "te": {
                     "interface-switching-capability": [
                       {
                         "switching-capability": "switching-otn",
                         "encoding": "lsp-encoding-oduk"
                       }
                     ]
                   }
                 }
               ]
             }
           ]
         }
       ]
     }
   }
```

Authors' Addresses

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

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc
Email: tsaad@cisco.com

Xufeng Liu
Jabil
Email: Xufeng_Liu@jabil.com

Network Working Group I. Bryskin
Internet-Draft Huawei Technologies
Intended status: Informational X. Liu
Expires: September 19, 2018 Jabil
J. Guichard
Y. Lee
Huawei Technologies
L. Contreras
Telefonica
D. Ceccarelli
Ericsson
J. Tantsura
Nuage Networks
March 18, 2018

Use Cases for SF Aware Topology Models
draft-bryskin-teas-use-cases-sf-aware-topo-model-03

Abstract

This document describes some use cases that benefit from the network topology models that are service and network function aware.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Normally network connectivity services are discussed as a means to inter-connect various abstract or physical network topological elements, such as ports, link termination points and nodes [I-D.ietf-teas-yang-te-topo] [I-D.ietf-teas-yang-te]. However, the connectivity services, strictly speaking, interconnect not the network topology elements per-se, rather, located on/associated with the various network and service functions [RFC7498] [RFC7665]. In many scenarios it is beneficial to decouple the service/network functions from the network topology elements hosting them, describe them in some unambiguous and identifiable way (so that it would be possible, for example, to auto-discover on the network topology service/network functions with identical or similar functionality and characteristics) and engineer the connectivity between the service/network functions, rather than between their current topological locations. The purpose of this document is to describe some use cases that could benefit from such an approach.

2. Terminology

- Network Function (NF): A functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behaviour [ETSI-NFV-TERM]. Such functions include message router, CDN, session border controller, WAN cceleration, DPI, firewall, NAT, QoE monitor, PE router, BRAS, and radio/fixed access network nodes.

- Network Service: Composition of Network Functions and defined by its functional and behavioural specification. The Network Service contributes to the behaviour of the higher layer service, which is characterized by at least performance, dependability, and security specifications. The end-to-end network service behaviour is the result of the combination of the individual network function behaviours as well as the behaviours of the network infrastructure composition mechanism [ETSI-NFV-TERM].

- Service Function (SF): A function that is responsible for specific treatment of received packets. A service function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). As a logical component, a service function can be realized as a virtual element or be embedded in a physical network element. One or more service functions can be embedded in the same network element. Multiple occurrences of the service function can exist in the same administrative domain. A non-exhaustive list of service functions includes: firewalls, WAN and application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and TCP optimizers. The generic term "L4-L7 services" is often used to describe many service functions [RFC7498].

- Service Function Chain (SFC): A service function chain defines an ordered or partially ordered set of abstract service functions and ordering constraints that must be applied to packets, frames, and/or flows selected as a result of classification. An example of an abstract service function is a firewall. The implied order may not be a linear progression as the architecture allows for SFCs that copy to more than one branch, and also allows for cases where there is flexibility in the order in which service functions need to be applied. The term "service chain" is often used as shorthand for "service function chain" [RFC7498].

- o Connectivity Service: Any service between layer 0 and layer 3 aiming at delivering traffic among two or more end customer edge nodes connected to provider edge nodes. Examples include L3VPN, L2VPN etc.

- o Link Termination Point (LTP): A conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1 [I-D.ietf-teas-yang-te-topo].

- o Tunnel Termination Point (TTP): An element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned with the TE node scope unique ID. Depending on the TE node’s internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node [I-D.ietf-teas-yang-te-topo].

3. Exporting SF/NF Information to Network Clients and Other Network SDN Controllers

In the context of Service Function Chain (SFC) orchestration one existing problem is that there is no way to formally describe a Service or Network Function in a standard way (recognizable/ understood by a third party) as a resource of a network topology node.

One implication of this is that there is no way for the orchestrator to give a network client even a ball-park idea as to which network’s SFs/NFs are available for the client’s use/control and where they are located in the network even in terms of abstract topologies/virtual networks configured and managed specifically for the client. Consequently, the client has no say on how the SFCs provided for the client by the network should be set up and managed (which SFs are to be used and how they should be chained together, optimized, manipulated, protected, etc.).

Likewise, there is no way for the orchestrator to export SF/NF information to other network controllers. The SFC orchestrator may serve, for example, a higher level controller (such as Network Slicing Orchestrator), with the latter wanting at least some level of control as to which SFs/NFs it wants on its SFCs and how the Service Function Paths (SFPs) are to be routed and provisioned, especially, if it uses services of more than one SFC orchestrator.

The issue of exporting of SF/NF information could be addressed by defining a model, in which formally described/recognizable SF/NF instances are presented as topological elements, for example, hosted by TE, L3 or L2 topology nodes (see Figure 1). The model could describe whether, how and at what costs the SFs/NFs hosted by a given node could be chained together, how these intra-node SFCs could be connected to the node’s Service Function Forwarders (SFFs, entities dealing with SFC NSHs and metadata), and how the SFFs could be connected to the node’s Tunnel and Link Termination Points (TTPs and LTPs) to chain the intra-node SFCs across the network topology.

The figure is available in the PDF format.

Figure 1: SF/NF aware TE topology

4. Flat End-to-end SFCs Managed on Multi-domain Networks

SFCs may span multiple administrative domains, each of which controlled by a separate SFC controller. The usual solution for such a scenario is the Hierarchical SFCs (H-SFCs), in which the higher level orchestrator controls only SFs located on domain border nodes. Said higher level SFs are chained together into higher level SFCs via lower level (intra-domain) SFCs provisioned and controlled independently by respective domain controllers. The decision as to which higher level SFCs are connected to which lower level SFCs is driven by packet re-classification every time the packet enters a given domain. Said packet re-classification is a very time-consuming operation. Furthermore, the independent nature of higher and lower level SFC control is prone to configuration errors, which may lead to long lasting loops and congestions. It is highly desirable to be able to set up and manage SFCs spanning multiple domains in a flat way as far as the data plane is concerned (i.e. with a single packet classification at the ingress into the multi-domain network but without re-classifications on domain ingress nodes).

One way to achieve this is to have the domain controllers expose SF/NF- aware topologies, and have the higher level orchestrator operate on the network-wide topology, the product of merging of the topologies catered by the domain controllers. This is similar in spirit to setting up, coordinating and managing the transport connectivity (TE tunnels) on a multi-domain multi-vendor transport network.

5. Managing SFCs with TE Constraints

Some SFCs require per SFC link/element and end-to-end TE constrains (bandwidth, delay/jitter, fate sharing/diversity. etc.). Said constraints could be ensured via carrying SFPs inside overlays that are traffic engineered with the constrains in mind. A good analogy would be orchestrating delay constrained L3 VPNs. One way to support such L3 VPNs is to carry MPLS LSPs interconnecting per-VPN VRFs inside delay constrained TE tunnels interconnecting the PEs hosting the VRFs.

_

Figure 2: L3 VPN with delay constraints

Planning, computing and provisioning of TE overlays to constrain arbitrary SFCs, especially those that span multiple administrative domains with each domain controlled by a separate controller, is a very difficult challenge. Currently it is addressed by pre-provisioning on the network of multiple TE tunnels with various TE characteristics, and "nailing down" SFs/NFs to "strategic" locations (e.g. nodes terminating many of such tunnels) in a hope that an adequate set of tunnels could be found to carry the SFP of a given TE-constrained SFC. Such an approach is especially awkward in the case when some or all of the SFs/NFs are VNFs (i.e. could be instantiated at multiple network locations).

SF/NF-aware TE topology model in combination with TE tunnel model will allow for the network orchestrator (or a client controller) to compute, set up and manipulate the TE overlays in the form of TE tunnel chains (see Figure 3).

Said chains could be duel-optimized compromising on optimal SF/NF locations with optimal TE tunnels interconnecting them. The TE tunnel chains (carrying multiple similarly constrained SFPs) could be adequately constrained both at individual TE tunnel level and at the chain end-to-end level.

Figure 3: SFC with TE constraints

6. SFC Protection and Load Balancing

Currently the combination of TE topology & tunnel models offers to a network controller various capabilities to recover an individual TE tunnel from network failures occurred on one or more network links or transit nodes on the TE paths taken by the TE tunnel’s connection(s). However, there is no simple way to recover a TE tunnel from a failure affecting its source or destination node. SF/NF-aware TE topology model can decouple the association of a given SF/NF with its location on the network topology by presenting multiple, identifiable as mutually substitutable SFs/NFs hosted by different TE topology nodes. So, for example, if it is detected that a given TE tunnel destination node is malfunctioning or has gone out of service, the TE tunnel could be re-routed to terminate on a different node hosting functionally the same SFs/NFs as ones hosted by the failed node (see Figures 6).

This is in line with the ACTN edge migration and function mobility concepts [I-D.ietf-teas-actn-framework]. It is important to note that the described strategy works much better for the stateless SFs/NFs. This is because getting the alternative stateful SFs/NFs into the same respective states as the current (i.e. active, affected by failure) are is a very difficult challenge.

_

Figure 4: SFC recovery: SF2 on node NE1 fails

At the SFC level the SF/NF-aware TE topology model can offer SFC dynamic restoration capabilities against failed/malfunctioning SFs/NFs by identifying and provisioning detours to a TE tunnel chain, so that it starts carrying the SFC's SFPs towards healthy SFs/NFs that are functionally the same as the failed ones. Furthermore, multiple parallel TE tunnel chains could be pre-provisioned for the purpose of SFC load balancing and end-to-end protection. In the latter case said parallel TE tunnel chains could be placed to be sufficiently disjoint from each other.

_

Figure 5: SFC recovery: SFC SF1-SF2-SF6 is recovered after SF2 on node N1 has failed

_

Figure 6: SFC recovery: SFC SF1-SF2-SF6 is recovered after node N1 has failed

7. Network Clock Synchronization

Many current and future network applications (including 5g and IoT applications) require very accurate time services (PTP level, ns resolution). One way to implement the adequate network clock synchronization for such services is via describing network clocks as NFs on an NF-aware TE topology optimized to have best possible delay variation characteristics. Because such a topology will contain delay/delay variation metrics of topology links and node cross-connects, as well as costs in terms of delay/delay variation of connecting clocks to hosting them node link and tunnel termination points, it will be possible to dynamically select and provision bi-directional time-constrained deterministic paths or trees connecting clocks (e.g. grand master and boundary clocks) for the purpose of exchange of clock synchronization information. Note that network clock aware TE topologies separately provided by domain controllers will enable multi-domain network orchestrator to set up and manipulate the clock synchronization paths/trees spanning multiple network domains.

8. Client - Provider Network Slicing Interface

3GPP defines network slice as "a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics" [I-D.defoy-netslices-3gpp-network-slicing] [_3GPP.28.801]. Network slice could be also defined as a logical partition of a provider’s network that is owned and managed by a tenant. SF/NF-aware TE topology model has a potential to support a very important interface between network slicing clients and providers because, on the one hand, the model can describe holistically and hierarchically the client’s requirements and preferences with respect to a network slice functional, topological and traffic engineering aspects, as well as of the degree of resource separation/ sharing between the slices, thus allowing for the client (up to agreed upon extent) to dynamically (re-)configure the slice or (re-)schedule said (re-)configurations in time, while, on the other hand, allowing for the provider to convey to the client the slice’s operational state information and telemetry the client has expressed interest in.

9. Dynamic Assignment of Regenerators for L0 Services

On large optical networks, some of provided to their clients L0 services could not be provisioned as single OCh trails, rather, as chains of such trails interconnected via regenerators, such as 3R regenerators. Current practice of the provisioning of such services requires configuration of explicit paths (EROs) describing identity and location of regenerators to be used. A solution is highly desirable that could:

- Identify such services based, for example, on optical impairment computations;

- Assign adequate for the services regenerators dynamically out of the regenerators that are grouped together in pools and strategically scattered over the network topology nodes;

- Compute and provision supporting the services chains of optical trails interconnected via so selected regenerators, optimizing the chains to use minimal number of regenerators, their optimal locations, as well as optimality of optical paths interconnecting them;

- Ensure recovery of such chains from any failures that could happen on links, nodes or regenerators along the chain path.

NF-aware TE topology model (in this case L1 NF-aware L0 topology model) is just the model that could provide a network controller (or even a client controller operating on abstract NF-aware topologies provided by the network) to realize described above computations and orchestrate the service provisioning and network failure recovery operations (see Figure 7).

Figure 7: Optical tunnel as TE-constrained SFC of 3R regenerators. Red trail (not regenerated) is not optically reachable, but blue trail (twice regenerated) is

10. Dynamic Assignment of OAM Functions for L1 Services

OAM functionality is normally managed by configuring and manipulating TCM/MEP functions on network ports terminating connections or their segments over which OAM operations, such as performance monitoring, are required to be performed. In some layer networks (e.g. Ethernet) said TCMs/MEPs could be configured on any network ports. In others (e.g. OTN/ODUk) the TCMs/MEPs could be configured on some (but not all network ports) due to the fact that the OAM functionality (i.e. recognizing and processing of OAM messages, supporting OAM protocols and FSMs) requires in these layer networks certain support in the data plane, which is not available on all network nodes. This makes TCMs/MEPs good candidates to be modeled as NFs. This also makes TCM/MEP aware topology model a good basis for placing dynamically an ODUk connection to pass through optimal OAM locations without mandating the client to specify said locations explicitly.

Figure 8: Compute/storage resource aware topology

11. SFC Abstraction and Scaling

SF/NF-aware topology may contain information on native SFs/NFs (i.e. SFs/NFs as known to the provider itself) and/or abstract SFs/NFs (i.e. logical/macro SFs/NFs representing one or more SFCs each made of native and/or lower level abstract SFs/NFs). As in the case of abstracting topology nodes, abstracting SFs/NFs is hierarchical in nature - the higher level of SF/NF-aware topology, the "larger" abstract SFs/NFs are, i.e. the larger data plane SFCs they represent. This allows for managing large scale networks with great number of SFs/NFs (such as Data Center interconnects) in a hierarchical, highly scalable manner resulting in control of very large number of flat in the data plane SFCs that span multiple domains.

12. Dynamic Compute/VM/Storage Resource Assignment

In a distributed data center network, virtual machines for compute resources may need to be dynamically re-allocated due to various reasons such as DCI network failure, compute resource load balancing, etc. In many cases, the DCI connectivity for the source and the destination is not predetermined. There may be a pool of sources and a pool of destination data centers associated with re-allocation of compute/VM/storage resources. There is no good mechanism to date to capture this dynamicity nature of compute/VM/storage resource reallocation. Generic Compute/VM/Storage resources can be described and announced as a SF, where a DC hosting these resources can be modeled as an abstract node. Topology interconnecting these abstract nodes (DCs) in general is of multi-domain nature. Thus, SF-aware topology model can facilitate a joint optimization of TE network resources and Compute/VM/Storage resources and solve Compute/VM/ Storage mobility problem within and between DCs (see Figure 8).

13. Application-aware Resource Operations and Management

Application stratum is the functional grouping which encompasses application resources and the control and management of these resources. These application resources are used along with network services to provide an application service to clients/end-users. Application resources are non-network resources critical to achieving the application service functionality. Examples of application resources include: caches, mirrors, application specific servers, content, large data sets, and computing power. Application service is a networked application offered to a variety of clients (e.g., server backup, VM migration, video cache, virtual network on-demand, 5G network slicing, etc.). The application servers that host these application resources can be modeled as an abstract node. There may be a variety of server types depending on the resources they host. Figure 9 shows one example application aware topology for video cache server distribution.

_

Figure 9: Application aware topology

14. IANA Considerations

This document has no actions for IANA.

15. Security Considerations

This document does not define networking protocols and data, hence is not directly responsible for security risks.

16. Acknowledgements

The authors would like to thank Maarten Vissers, Joel Halpern, and Greg Mirsky for their helpful comments and valuable contributions.

17. References

17.1. Normative References

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[ETSI-NFV-TERM] ETSI, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 003 V1.2.1, December 2014.

[I-D.ietf-teas-yang-te-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-15 (work in progress), February 2018.

[I-D.ietf-teas-yang-te] Saad, T., Gandhi, R., Liu, X., Beeram, V., Shah, H., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-14 (work in progress), March 2018.

17.2. Informative References

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <https://www.rfc-editor.org/info/rfc3022>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[I-D.ietf-sfc-hierarchical] Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", draft-ietf-sfc-hierarchical-07 (work in progress), February 2018.

[I-D.defoy-netslices-3gpp-network-slicing]
Foy, X. and A. Rahman, "Network Slicing - 3GPP Use Case", draft-defoy-netslices-3gpp-network-slicing-02 (work in progress), October 2017.

[_3GPP.28.801]
3GPP, "Study on management and orchestration of network slicing for next generation network", 3GPP TR 28.801 V2.0.0, September 2017, <http://www.3gpp.org/ftp/Specs/html-info/28801.htm>.

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-11 (work in progress), October 2017.

Authors' Addresses

Igor Bryskin
Huawei Technologies

EMail: Igor.Bryskin@huawei.com

Xufeng Liu
Jabil

EMail: xufeng.liu.ietf@gmail.com

Jim Guichard
Huawei Technologies

EMail: james.n.guichard@huawei.com

Young Lee
Huawei Technologies

EMail: leeyoung@huawei.com

Luis Miguel Contreras Murillo
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Daniele Ceccarelli
Ericsson

EMail: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Nuage Networks

EMail: jefftant.ietf@gmail.com

Teas Working Group
Internet Draft

Intended status: Informational

Expires: December 21, 2018

Young Lee
Huawei

Sergio Belotti
Nokia

Dhruv Dhody
Huawei

Daniele Ceccarelli
Ericsson

Bin Yeong Yoon
ETRI

June 21, 2018

Information Model for Abstraction and Control of TE Networks (ACTN)

draft-ietf-teas-actn-info-model-10.txt

Abstract

This draft provides an information model for Abstraction and Control of Traffic Engineered Networks (ACTN).

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

This draft provides an information model for Abstraction and Control of Traffic Engineered Networks (ACTN). The information model described in this document covers the interface requirements identified in the ACTN architecture and framework document [ACTN-Frame].

The ACTN reference architecture [ACTN-Frame] identifies a three-tier control hierarchy comprising the following as depicted in Figure 1:

- Customer Network Controllers (CNCs)
- Multi-Domain Service Coordinator (MDSC)
- Provisioning Network Controllers (PNCs).

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

Figure 1: A Three-tier ACTN control hierarchy

The two interfaces with respect to the MDSC, one north of the MDSC and the other south of the MDSC are referred to as CMI (CNC-MDSC Interface) and MPI (MDSC-PNC Interface), respectively. This document models these two interfaces and derivative interfaces thereof (e.g., MDSC to MDSC in a hierarchy of MDSCs) as a single common interface.

1.1. Terminology

The terms "Virtual Network (VN)" and "Virtual Network Service (VNS)" are defined in [ACTN-Frame] and the other key terms such as "abstraction", "abstract topology", "Path", "VN node", and "VN link" are defined in [RFC7926].

2. ACTN Common Interfaces Information Model

This section provides an ACTN common interface information model to describe primitives, objects, their properties (represented as attributes), their relationships, and the resources for the service applications needed in the ACTN context.

The standard interface is described between a client controller and a server controller. A client-server relationship is recursive between a CNC and an MDSC and between an MDSC and a PNC. In the CMI, the client is a CNC while the server is an MDSC. In the MPI, the client is an MDSC and the server is a PNC. There may also be MDSC-MDSC interface(s) that need to be supported. This may arise in a hierarchy of MDSCs in which workloads may need to be partitioned to multiple MDSCs.

Basic primitives (messages) are required between the CNC-MDSC and MDSC-PNC controllers. These primitives can then be used to support different ACTN network control functions like network topology request/query, VN service request, path computation and connection control, VN service policy negotiation, enforcement, routing options, etc.

There are two different types of primitives depending on the type of interface:

- Virtual Network primitives at CMI
- Traffic Engineering primitives at MPI

As well described in [ACTN-Frame], at the CMI level, there is no need for detailed TE information since the basic functionality is to translate customer service information into virtual network service operation.

At the MPI level, MDSC has the main scope for multi-domain coordination and creation of a single e2e abstracted network view which is strictly related to TE information.

As for topology, this document employs two types of topology:

- The first type is referred to as virtual network topology which is associated with a VN. Virtual network topology is a customized topology for view and control by the customer. See Section 3.1 for details.

- The second type is referred to as TE topology which is associated with provider network operation on which we can apply policy to obtain the required level of abstraction to represent the underlying physical network topology.

3. Virtual Network primitives

This section provides a list of main VN primitives related to virtual network which are necessary to satisfy ACTN requirements specified in [ACTN-REQ]

The following VN Action primitives are supported:

- VN Instantiate
- VN Modify
- VN Delete
- VN Update
- VN Path Compute
- VN Query

VN Action is an object describing the main VN primitives.

VN Action can assume one of the mentioned above primitives values.

```
<VN Action> ::= <VN Instantiate> |

                <VN Modify> |

                <VN Delete> |

                <VN Update> |

                <VN Path Compute> |

                <VN Query>
```

All these actions will solely happen at CMI level between Customer Network Controller (CNC) and Multi Domain Service Coordinator (MDSC).

3.1. VN Instantiate

VN Instantiate refers to an action from customers/applications to request the creation of VNs. VN Instantiate is for CNC-to-MDSC communication. Depending on the agreement between client and provider, VN instantiate can imply different VN operations. There are two types of VN instantiation:

VN type 1: VN is viewed as a set of edge-to-edge links (VN members).

VN type 2: VN is viewed as a VN-topology comprising virtual nodes and virtual links.

Please see [ACTN-Frame] for full details regarding the types of VN.

3.2. VN Modify

VN Modify refers to an action issued from customers/applications to modify an existing VN (i.e., an instantiated VN). VN Modify is for CNC-to-MDSC communication.

VN Modify, depending of the type of VN instantiated, can be a modification of the characteristics of VN members (edge-to-edge links) in case of VN type 1, or a modification of an existing virtual topology (e.g., adding/deleting virtual nodes/links) in case of VN type 2.

3.3. VN Delete

VN Delete refers to an action issued from customers/applications to delete an existing VN. VN Delete is for CNC-to-MDSC communication.

3.4. VN Update

VN Update refers to any update to the VN that needs to be updated to the customers. VN Update is MDSC-to-CNC communication. VN Update fulfills a push model at CMI level, to make customers aware of any specific changes in the topology details related to the instantiated VN.

VN Update, depending of the type of VN instantiated, can be an update of VN members (edge-to-edge links) in case of VN type 1, or an update of virtual topology in case of VN type 2.

The connection-related information (e.g., LSPs) update association with VNs will be part of the "translation" function that happens in MDSC to map/translate VN request into TE semantics. This information will be provided in case customer optionally wants to have more detailed TE information associated with the instantiated VN.

3.5. VN Compute

VN Compute consists of Request and Reply. Request refers to an action from customers/applications to request a VN computation.

VN Compute Reply refers to the reply in response to VN Compute Request.

VN Compute Request/Reply is to be differentiated from a VN Instantiate. The purpose of VN Compute is a priori exploration to compute network resources availability and getting a possible VN view in which path details can be specified matching customer/applications constraints. This a priori exploration may not guarantee the availability of the computed network resources at the time of instantiation.

3.6. VN Query

VN Query refers to an inquiry pertaining to a VN that has already been instantiated. VN Query fulfills a pull model that permits getting a topology view.

VN Query Reply refers to the reply in response to VN Query. The topology view returned by VN Query Reply would be consistent with the topology type instantiated for any specific VN.

4. Traffic Engineering (TE) primitives

This section provides a list of the main TE primitives necessary to satisfy ACTN requirements specified in [ACTN-REQ] related to typical TE operations supported at the MPI level.

The TE action primitives defined in this section should be supported at the MPI consistently with the type of topology defined at the CMI.

The following TE action primitives are supported:

- TE Instantiate/Modify/Delete

- TE Topology Update (See Section 4.4. for the description)

- Path Compute

TE Action is an object describing the main TE primitives.

TE Action can assume one of the mentioned above primitives values.

```
<TE Action> ::= <TE Instantiate> |

                <TE Modify> |

                <TE Delete> |

                <TE Topology Update> |
                   <Path Compute> |
```

All these actions will solely happen at MPI level between Multi Domain Service Coordinator (MDSC) and Provisioning Network Controller (PNC).

4.1. TE Instantiate

TE Instantiate refers to an action issued from MDSC to PNC to instantiate new TE tunnels.

4.2. TE Modify

TE Modify refers to an action issued from MDSC to PNC to modify existing TE tunnels.

4.3. TE Delete

TE Delete refers to an action issued from MDSC to PNC to delete existing TE tunnels.

4.4. TE Topology Update (for TE resources)

TE Topology Update is a primitive specifically related to MPI to provide TE resource update between any domain controller towards MDSC regarding the entire content of any "domain controller" actual TE topology or an abstracted filtered view of TE topology depending on negotiated policy.

See [TE-TOPO] for detailed YANG implementation of TE topology update.

```
<TE Topology Update> ::= <TE-topology-list>

<TE-topology-list> ::= <TE-topology> [<TE-topology-list>]

<TE-topology> ::= [<Abstraction>] <TE-Topology-identifier> <Node-
list> <Link-list>
```

```
<Node-list> ::= <Node>[<Node-list>]

<Node> ::= <Node> <TE Termination Point-list>

<TE Termination Point-list> ::= <TE Termination Point> [<TE-
Termination Point-list>]

<Link-list> ::= <Link>[<Link-list>]
```

Where

Abstraction provides information on level of abstraction (as determined a priori).

TE-topology-identifier is an identifier that identifies a specific te-topology, e.g., te-types:te-topology-id [TE-TOPO].

Node-list is detailed information related to a specific node belonging to a te-topology, e.g., te-node-attributes [TE-TOPO].

Link-list is information related to the specific link related belonging to a te-topology, e.g., te-link-attributes [TE-TOPO].

TE Termination Point-list is detailed information associated with the termination points of te-link related to a specific node, e.g., interface-switching-capability [TE-TOPO].

4.5. Path Compute

Path Compute consists of Request and Reply. Request refers to an action from MDSC to PNC to request a path computation.

Path Compute Reply refers to the reply in response to Path Compute Request.

The context of Path Compute is described in [Path-Compute].

5. VN Objects

This section provides a list of objects associated to VN action primitives.

5.1. VN Identifier

VN Identifier is a unique identifier of the VN.

5.2. VN Service Characteristics

VN Service Characteristics describes the customer/application requirements against the VNs to be instantiated.

```
<VN Service Characteristics> ::= <VN Connectivity Type>

                                 <VN Directionality>

                                 (<VN Traffic Matrix>...)

                                 <VN Survivability>
```

Where

```
<VN Connectivity Type> ::= <P2P>|<P2MP>|<MP2MP>|<MP2P>|<Multi-
destination>
```

The Connectivity Type identifies the type of required VN Service. In addition to the classical type of services (e.g. P2P/P2MP etc.), ACTN defines the "multi-destination" service that is a new P2P service where the end points are not fixed. They can be chosen among a list of pre-configured end points or dynamically provided by the CNC.

VN Directionality indicates if a VN is unidirectional or bidirectional. This implies that each VN member that belongs to the VN has the same directionality as the VN.

```
<VN Traffic Matrix> ::= <Bandwidth>

                        [<VN Constraints>]
```

The VN Traffic Matrix represents the traffic matrix parameters for the required service connectivity. Bandwidth is a mandatory parameter and a number of optional constraints can be specified in the VN Constraints (e.g. diversity, cost). They can include objective functions and TE metrics bounds as specified in [RFC5541].

Further details on the VN constraints are specified below:

```
      <VN Constraints> ::= [<Layer Protocol>]
```

```
                        [<Diversity>]

                        ( <Metric> | <VN Objective Function> )
```

Where:

Layer Protocol identifies the layer topology at which the VN service is requested. It could be for example MPLS, ODU, and OCh.

Diversity allows asking for diversity constraints for a VN Instantiate/Modify or a VN Path Compute. For example, a new VN or a path is requested in total diversity from an existing one (e.g. diversity exclusion).

```
<Diversity> ::= (<VN-exclusion> (<VN-id>...)) |

         (<VN-Member-exclusion> (<VN-Member-id>...))
```

Metric can include all the Metrics (cost, delay, delay variation, latency), bandwidth utilization parameters defined and referenced by [RFC3630] and [RFC7471].

As for VN Objective Function See Section 5.4.

VN Survivability describes all attributes related to the VN recovery level and its survivability policy enforced by the customers/applications.

```
<VN Survivability> ::= <VN Recovery Level>

                        [<VN Tunnel Recovery Level>]

                        [<VN Survivability Policy>]
```

Where:

VN Recovery Level is a value representing the requested level of resiliency required against the VN. The following values are defined:

. Unprotected VN
. VN with per tunnel recovery: The recovery level is defined against the tunnels composing the VN and it is specified in the VN Tunnel Recovery Level.

```
<VN Tunnel Recovery Level> ::= <0:1>|<1+1>|<1:1>|<1:N>|<M:N>|
                              <On the fly restoration>
```

The VN Tunnel Recovery Level indicates the type of protection or restoration mechanism applied to the VN. It augments the recovery types defined in [RFC4427].

```
<VN Survivability Policy> ::= [<Local Reroute Allowed>]

                              [<Domain Preference>]

                              [<Push Allowed>]

                              [<Incremental Update>]
```

Where:

Local Reroute Allowed is a delegation policy to the Server to allow or not a local reroute fix upon a failure of the primary LSP.

Domain Preference is only applied on the MPI where the MDSC (client) provides a domain preference to each PNC (server), e.g., when an inter-domain link fails, then PNC can choose the alternative peering with this info.

Push Allowed is a policy that allows a server to trigger an updated VN topology upon failure without an explicit request from the client. Push action can be set as default unless otherwise specified.

Incremental Update is another policy that triggers an incremental update from the server since the last period of update. Incremental update can be set as default unless otherwise specified.

5.3. VN End-Point

VN End-Point Object describes the VN’s customer end-point characteristics.

```
<VN End-Point> ::= (<Access Point Identifier>

                   [<Access Link Capability>]
```

[<Source Indicator>])...

Where:

Access Point Identifier represents a unique identifier of the client end-point. They are used by the customer to ask for the setup of a virtual network instantiation. A VN End-Point is defined against each AP in the network and is shared between customer and provider. Both the customer and the provider will map it against their own physical resources.

Access Link Capability identifies the capabilities of the access link related to the given access point. (e.g., max-bandwidth, bandwidth availability, etc.)

Source Indicator indicates if an end-point is source or not.

5.4. VN Objective Function

The VN Objective Function applies to each VN member (i.e., each E2E tunnel) of a VN.

The VN Objective Function can reuse objective functions defined in [RFC5541] section 4.

For a single path computation, the following objective functions are defined:

- o MCP is the Minimum Cost Path with respect to a specific metric (e.g. shortest path).
- o MLP is the Minimum Load Path, that means find a path composted by te-link least loaded.
- o MBP is the Maximum residual Bandwidth Path.

For a concurrent path computation, the following objective functions are defined:

- o MBC is to Minimize aggregate Bandwidth Consumption.
- o MLL is to Minimize the Load of the most loaded Link.
- o MCC is to Minimize the Cumulative Cost of a set of paths.

5.5. VN Action Status

VN Action Status is the status indicator whether the VN has been successfully instantiated, modified, or deleted in the server network or not in response to a particular VN action.

Note that this action status object can be implicitly indicated and thus not included in any of the VN primitives discussed in Section 3.

5.6. VN Topology

When a VN is seen by the customer as a topology, it is referred to as VN topology. This is associated with VN Type 2, which is composed of virtual nodes and virtual links.

```
<VN Topology> ::= <VN node list> <VN link list>

<VN node list> ::= <VN node> [<VN node list>]

<VN link list> :: = <VN link>  [<VN link list>]
```

5.7. VN Member

VN Member describes details of a VN Member which is a list of a set of VN Members represented as VN_Member_List.

```
<VN_Member_List> ::= <VN Member> [<VN_Member_List>]



Where <VN Member> ::= <Ingress VN End-Point>

                      [<VN Associated LSP>]

                      <Egress VN End-Point>
```

Ingress VN End-Point is the VN End-Point information for the ingress portion of the AP. See Section 5.3 for VN End-Point details.

Egress VN End-Point is the VN End-Point information for the egress portion of the AP. See Section 5.3 for VN End-Point details.

VN Associated LSP describes the instantiated LSPs in the Provider’s network for the VN Type 1. It describes the instantiated LSPs over the VN topology for VN Type 2.

5.7.1. VN Computed Path

The VN Computed Path is the list of paths obtained after the VN path computation request from a higher controller. Note that the computed path is to be distinguished from the LSP. When the computed path is signaled in the network (and thus the resource is reserved for that path), it becomes an LSP.

```
<VN Computed Path> ::= (<Path>...)
```

5.7.2. VN Service Preference

This section provides VN Service preference. VN Service is defined in Section 2.

```
<VN Service Preference> ::= [<Location Service Preference >]

                            [<Client-specific Preference >]

                            [<End-Point Dynamic Selection Preference >]
```

Where

Location Service Preference describes the End-Point Location’s (e.g. Data Centers) support for certain Virtual Network Functions (VNFs) (e.g., security function, firewall capability, etc.) and is used to find the path that satisfies the VNF constraint.

Client-specific Preference describes any preference related to Virtual Network Service (VNS) that application/client can enforce via CNC towards lower level controllers. For example, CNC can enforce client-specific preferences, e.g., selection of a destination data center from the set of candidate data centers based on some criteria in the context of VM migration. MSDC/PNC should then provide the data center interconnection that supports the client-specific preference.

End-Point Dynamic Selection Preference describes if the End-Point (e.g. Data Center) can support load balancing, disaster recovery or VM migration and so can be part of the selection by MDSC following service Preference enforcement by CNC.

6. TE Objects

6.1. TE Tunnel Characteristics

Tunnel Characteristics describes the parameters needed to configure TE tunnel.

```
<TE Tunnel Characteristics> ::= [<Tunnel Type>]

                                <Tunnel Id>

                                [<Tunnel Layer>]

                                [<Tunnel end-point>]

                                [<Tunnel protection-restoration>]

                                <Tunnel Constraints>

                               [<Tunnel Optimization>]
```

Where

<Tunnel Type> ::= <P2P>|<P2MP>|<MP2MP>|<MP2P>

The Tunnel Type identifies the type of required tunnel. In this draft, only P2P model is provided.

Tunnel Id is the TE tunnel identifier

Tunnel Layer represents the layer technology of the LSPs supporting the tunnel

<Tunnel End Points> ::= <Source> <Destination>

<Tunnel protection-restoration> ::= <prot 0:1>|<prot 1+1>|<prot 1:1>|<prot 1:N>|prot <M:N>|<restoration>

Tunnel Constraints are the base tunnel configuration constraints parameters.

```
Where <Tunnel Constraints> ::= [<Topology Id>]

                               [<Bandwidth>]

                               [<Disjointness>]

                               [<SRLG>]

                               [<Priority>]

                               [<Affinities>]

                               [<Tunnel Optimization>]

                               [<Objective Function>]
```

Topology Id references the topology used to compute the tunnel path.

Bandwidth is the bandwidth used as parameter in path computation

<Disjointness> ::= <node> | <link> | <srlg>

Disjointness provides the type of resources from which the tunnel has to be disjointed

SRLG is a group of physical resources impacted by the same risk from which an E2E tunnel is required to be disjointed.

<Priority> ::= <Holding Priority> <Setup Priority>

where

Setup Priority indicates the level of priority for taking resources from another tunnel [RFC3209]

Holding Priority indicates the level of priority to hold resources avoiding preemption from another tunnel [RFC3209]

Affinities it represent structure to validate link belonging to path of the tunnel [RFC3209]

<Tunnel Optimization> ::= <Metric> | <Objective Function>

Metric can include all the Metrics (cost, delay, delay variation, latency), bandwidth utilization parameters defined and referenced by [RFC3630] and [RFC7471].

```
<Objective Function> ::= <objective function type>

<objective function type> ::= <MCP> | <MLP> | <MBP> | <MBC> | <MLL>
| <MCC>
```

See chapter 5.4 for objective function type description.

7. Mapping of VN primitives with VN Objects

This section describes the mapping of VN primitives with VN Objects based on Section 5.

```
<VN Instantiate> ::= <VN Service Characteristics>

                     <VN Member-List>

                     [<VN Service Preference>]

                     [<VN Topology>]


<VN Modify> ::= <VN identifier>

                <VN Service Characteristics>

                <VN Member-List>

                [<VN Service Preference>]

                [<VN Topology>]


<VN Delete> ::= <VN Identifier>


<VN Update> :: = <VN Identifier>
```

```
                    [<VN Member-List>]

                    [<VN Topology>]


   <VN Path Compute Request> ::= <VN Service Characteristics>

                                 <VN Member-List>

                                 [<VN Service Preference>]


   <VN Path Compute Reply> ::= <VN Computed Path>


   <VN Query> ::= <VN Identifier>


   <VN Query Reply> ::= <VN Identifier>

                        <VN Associated LSP>

                        [<TE Topology Reference>]
```

8. Mapping of TE primitives with TE Objects

This section describes the mapping of TE primitives with TE Objects based on Section 6.

```
   <TE Instantiate> ::= <TE Tunnel Characteristics>

   <TE Modify> ::=  <TE Tunnel Characteristics>


   <TE Delete> ::= <Tunnel Id>
```

```
<TE Topology Update> ::= <TE-topology-list>

<Path Compute Request> ::= <TE Tunnel Characteristics>


<Path Compute Reply> ::= <TE Computed Path>

                         <TE Tunnel Characteristics>
```

9. Security Considerations

The ACTN information model is not directly relevant when considering potential security issues. Rather, it defines a set of interfaces for traffic engineered networks. The underlying protocols, procedures, and implementations used to exchange the information model described in this draft will need to secure the request and control of resources with proper authentication and authorization mechanisms. In addition, the data exchanged over the ACTN interfaces discussed in this document requires verification of data integrity. Backup or redundancies should also be available to restore the affected data to its correct state.

Implementations of the ACTN framework will have distributed functional components that will exchange a concrete instantiation that adheres to this information model. Implementations should encrypt data that flows between them, especially when they are implemented at remote nodes and irrespective of whether these data flows are on external or internal network interfaces. The information model may contain customer, application and network data that for business or privacy reasons may be considered sensitive. It should be stored only in an encrypted data store.

The ACTN security discussion is further split into two specific interfaces:

- Interface between the Customer Network Controller and Multi Domain Service Coordinator (MDSC), CNC-MDSC Interface (CMI)

- Interface between the Multi Domain Service Coordinator and Provisioning Network Controller (PNC), MDSC-PNC Interface (MPI)

See the detailed discussion of the CMI and MPI in Sections 9.1 and 9.2, respectively in [ACTN-Frame].

The conclusion is that all data models and protocols used to realize the ACTN info model should have rich security features as discussed in this section. Additional security risks may still exist. Therefore, discussion and applicability of specific security functions and protocols will be better described in documents that are use case and environment specific.

10. IANA Considerations

This document has no actions for IANA.

11. References

11.1. Normative References

[ACTN-Frame] D. Ceccarelli, et al., "Framework for Abstraction and Control of Transport Networks", draft-ietf-teas-actn-framework, work in progress.

11.2. Informative References

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

[RFC3209] D. Awduche, et al, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3630] D. Katz, K. Kompella, D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4427] E. Mannie, D. Papadimitriou (Editors), "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, March 2006.

[RFC5541] JL. Le Roux, JP. Vasseur and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.

[RFC7471] S. Giacalone, et al, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.

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

[Path-Compute] I. Busi, S. Belotti, et al., "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation", work in progress.

13. Contributors

Contributors’ Addresses

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

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

Authors’ Addresses

Young Lee (Editor)
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838
Email: leeyoung@huawei.com

Sergio Belotti (Editor)
Alcatel Lucent
Via Trento, 30
Vimercate, Italy
Email: sergio.belotti@alcatel-lucent.com

Dhruv Dhody
Huawei Technologies,
Divyashree Technopark, Whitefield
Bangalore, India
Email: dhruv.ietf@gmail.com

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

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

```
TEAS Working Group                                       Zafar Ali, Ed.
Internet Draft                                      George Swallow, Ed.
Intended status: Standard Track                           Cisco Systems
Updates RFC4874                                          F. Zhang, Ed.
Expires: September 03, 2018                                      Huawei
                                                          D. Beller, Ed.
                                                                  Nokia
                                                         March 02, 2018
```

Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Path Diversity using Exclude Route

draft-ietf-teas-lsp-diversity-10.txt

Status of this Memo

Copyright Notice

Abstract

Resource ReSerVation Protocol-Traffic Engineering provides support for the communication of exclusion information during label switched path (LSP) setup. A typical LSP diversity use case is for protection, where two LSPs should follow different paths through the network in order to avoid single points of failure, thus greatly improving service availability. This document specifies an approach which can be used for network scenarios where full knowledge of the path(s) is not necessarily known by use of an abstract identifier for the path. Three types of abstract identifiers are specified: client-based, Path Computation Engine (PCE)-based, network-based. This document specifies two new diversity subobjects for the RSVP eXclude Route Object (XRO) and the Explicit Exclusion Route Subobject (EXRS).

For the protection use case, LSPs are typically created at a slow rate and exist for a long time, so that it is reasonable to assume that a given (reference) path currently existing, with a well-known identifier, will continue to exist and can be used as a reference when creating the new diverse path. Re-routing of the existing (reference)LSP, before the new path is established, is not considered.

Conventions used in this document

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

Table of Contents

Terms and Abbreviations

Diverse LSP: a diverse Label-Switched Path (LSP) is an LSP that has a path that does not have any link or SRLG in common with the path of a given LSP. Diverse LSPs are meaningful in the context of protection or restoration.

ERO: Explicit Route Object as defined in [RFC3209]

EXRS: Explicit eXclusion Route Subobject as defined in [RFC4874]

SRLG: Shared Risk Link Group as defined in [RFC4202]

Reference Path: the reference path is the path of an existing LSP, to which the path of a diverse LSP shall be diverse.

XRO: eXclude Route Object as defined in [RFC4874]

1. Introduction

Path diversity for multiple connections is a well-known operational requirement. Diversity constraints ensure that Label-Switched Paths (LSPs) can be established without sharing network resources, thus greatly reducing the probability of simultaneous connection failures.

The source node can compute diverse paths for LSPs when it has full knowledge of the network topology and is permitted to signal an Explicit Route Object (ERO). However, there are scenarios where different nodes perform path computations, and therefore there is a need for relevant diversity constraints to be signaled to those nodes. These include (but are not limited to):

- LSPs with loose hops in the Explicit Route Object, e.g. inter-domain LSPs.

- Generalized Multi-Protocol Label Switching (GMPLS) User-Network Interface (UNI), where the core node may perform path computation [RFC4208].

[RFC4874] introduced a means of specifying nodes and resources to be excluded from a route, using the eXclude Route Object (XRO) and Explicit Exclusion Route Subobject (EXRS). It facilitates the calculation of diverse paths for LSPs based on known properties of those paths including addresses of links and nodes traversed, and Shared Risk Link Groups (SRLGs) of traversed links. Employing these mechanisms requires that the source node that initiates signaling knows the relevant properties of the path(s) from which diversity is desired. However, there are circumstances under which this may not be possible or desirable, including (but not limited to):

- Exclusion of a path which does not originate, terminate or traverse the source node of the diverse LSP, in which case the addresses of links and SRLGs of the path from which diversity is required are unknown to the source node.

- Exclusion of a path which is known to the source node of the diverse LSP for which the node has incomplete or no path information, e.g. due to operator policy. In this case, the source node is aware of the existence of the reference path but the information required to construct an XRO object to guarantee diversity from the reference path is not fully known. Inter-domain and GMPLS overlay networks can impose such restrictions.

This is illustrated in the Figure 1, where the overlay reference model from [RFC4208] is shown.

```
   Overlay                                                  Overlay
   Network       +----------------------------------+       Network
 +---------+     |                                  |     +---------+
 |  +----+ |     |  +-----+    +-----+    +-----+   |     | +----+  |
 |  |    | | UNI |  |     |    |     |    |     |   | UNI | |    |  |
 | -+ EN1+-+-----+--+ CN1 +----+ CN2 +----+ CN3 +---+-----+-+ EN3+- |
 |  |    | |  +--+--+     |    |     |    |     |   | +---+-|    |  |
 |  +----+ |  |  |  +--+--+    +--+--+    +--+--+   | |   | +----+  |
 +---------+  |  |     |          |          |      | |   +---------+
              |  |     |          |          |      | |
 +---------+  |  |  +--+--+       |       +--+--+   | |   +---------+
 |  +----+ |  |  |  |     |       +-------+     +-----+   | +----+  |
 |  |    +-+--+  |  | CN4 +---------------+ CN5 |   |     | |    |  |
 | -+ EN2+-+-----+--+     |               |     +---+-----+-+ EN4+- |
 |  |    | | UNI |  +-----+               +-----+   | UNI | |    |  |
 |  +----+ |     |                                  |     | +----+  |
 +---------+     +----------------------------------+     +---------+
   Overlay                 Core Network                     Overlay
   Network                                                  Network

                     Legend:   EN  -  Edge Node
                               CN  -  Core Node
```

Figure 1: Overlay Reference Model [RFC4208]

Figure 1 depicts two types of UNI connectivity: single-homed and dual-homed ENs (which also applies to higher order multi-homed connectivity). Single-homed EN devices are connected to a single CN device via a single UNI link. This single UNI link may constitute a single point of failure. UNI connection between EN1 and CN1 is an example of singled-homed UNI connectivity.

Such a single point of failure can be avoided when the EN device is connected to two different CN devices, as depicted for EN2 in Figure 1. For the dual-homing case, it is possible to establish two different UNI connections from the same source EN device to the same destination EN device. For example, two connections from EN2 to EN3 may use the two UNI links EN2-CN1 and EN2-CN4. To avoid single points of failure within the provider network, it is necessary to also ensure path (LSP) diversity within the core network.

In a network providing a set of UNI interfaces between ENs and CNs such as that shown in Figure 1, the CNs typically perform path computation. Information sharing across the UNI boundary is restricted based on the policy rules imposed by the core network. Typically, the core network topology information as well as LSP path information is not exposed to the ENs. In the network shown in Figure 1, consider a use case where an LSP from EN2 to EN4 needs to be SRLG diverse from an LSP from EN1 to EN3. In this case, EN2 may not know SRLG attributes of the EN1- EN3 LSP and hence cannot construct an XRO to exclude these SRLGs. In this example EN2 cannot use the procedures described in [RFC4874]. Similarly, an LSP from EN2 to EN3 traversing CN1 needs to be diverse from an LSP from EN2 to EN3 going via CN4. Again, in this case, exclusions based on [RFC4874] cannot be used.

This document addresses these diversity requirements by introducing an approach of excluding the path taken by these particular LSP(s). The reference LSP(s) or route(s) from which diversity is required is/are identified by an abstract "identifier". The type of identifier to use is highly dependent on the core network operator’s networking deployment scenario; it could be client-initiated (provided by the EN), provided by a PCE or allocated by the (core) network. This document defines three different types of identifiers corresponding to these three cases: a client-initiated identifier, a PCE allocated identifier and CN ingress node (UNI-N) allocated identifier (= network-assigned identifier).

1.1. Client-Initiated Identifier

The following fields MUST be used to represent the client-initiated identifier: IPv4/IPv6 tunnel sender address, IPv4/IPv6 tunnel endpoint address, Tunnel ID, and Extended Tunnel ID. Based on local policy, the client MAY also include the LSP ID to identify a specific LSP within the tunnel. These fields are defined in [RFC3209], sections 4.6.1.1 and 4.6.2.1.

The usage of the client-initiated identifier is illustrated by Figure 1. Suppose a LSP from EN2 to EN4 needs to be diverse with respect to a LSP from EN1 to EN3. The LSP identifier of the EN1-EN3 LSP is LSP-IDENTIFIER1, where LSP-IDENTIFIER1 is defined by the tuple (tunnel-id = T1, LSP ID = L1, source address = EN1.RID (ROUTE Identifier), destination address = EN3.RID, extended tunnel-id = EN1.RID). Similarly, LSP identifier of the EN2-EN4 LSP is LSP-IDENTIFIER2, where LSP-IDENTIFIER2 is defined by the tuple (tunnel-id = T2, LSP ID = L2, source address = EN2.RID, destination address = EN4.RID, extended tunnel-id = EN2.RID). The

EN1-EN3 LSP is signaled with an exclusion requirement from LSP-IDENTIFIER2, and the EN2-EN4 LSP is signaled with an exclusion requirement from LSP-IDENTIFIER1. In order to maintain diversity between these two connections within the core network, the core network SHOULD implement Crankback Signaling Extensions as defined in [RFC4920]. Note that crankback signaling is known to lead to slower setup times and sub-optimal paths under some circumstances as described by [RFC4920].

1.2. PCE-allocated Identifier

In scenarios where a PCE is deployed and used to perform path computation, the core edge node (e.g., node CN1 in Figure 1) could consult a PCE to allocate identifiers, which are used to signal path diversity constraints. In other deployment scenarios, a PCE is deployed at a network node(s) or a PCE is part of a Network Management System (NMS). In all these cases, the PCE is consulted and the Path-Key as defined in [RFC5520] can be used in RSVP signaling as the identifier to ensure diversity.

An example of specifying LSP diversity using a Path-Key is shown in Figure 2, where a simple network with two domains is shown. It is desired to set up a pair of path-disjoint LSPs from the source in Domain 1 to the destination in Domain 2, but the domains keep strict confidentiality about all path and topology information.

The first LSP is signaled by the source with ERO {A, B, loose Dst} and is set up with the path {Src, A, B, U, V, W, Dst}. However, when sending the Record Route Object (RRO) out of Domain 2, node U would normally strip the path and replace it with a loose hop to the destination. With this limited information, the source is unable to include enough detail in the ERO of the second LSP to avoid it taking, for example, the path {Src, C, D, X, V, W, Dst} for path-disjointness.

```
       --------------------    ----------------------------
      | Domain 1           |  |                   Domain 2 |
      |                    |  |                            |
      |       ---    ---   |  |   ---    ---     ---       |
      |      | A |--| B |--+--+--| U |--| V |---| W |      |
      |     / ---    ---   |  |   ---    ---     --- \     |
      |  ---/              |  |         /       /     \--- |
      | |Src|              |  |        /       /      |Dst| |
      |  ---\              |  |       /       /       /--- |
      |     \ ---    ---   |  |   --- /   --- /  --- /     |
      |      | C |--| D |--+--+--| X |---| Y |--| Z |      |
      |       ---    ---   |  |   ---     ---    ---       |
      |                    |  |                            |
       --------------------    ----------------------------
```

Figure 2: A Simple Multi-Domain Network

In order to support LSP diversity, node U consults the PCE and replaces the path segment {U, V, W} in the RRO with a Path Key subobject. The PCE function assigns an "identifier" and puts it into the Path Key field of the Path Key subobject. The PCE ID in the message indicates that this replacement operation was performed by node U.

With this additional information, the source node is able to signal the subsequent LSPs with the ERO set to {C, D, exclude Path Key(EXRS), loose Dst}. When the signaling message reaches node X, it can consult the PCE function associated with node U to expand the Path Key in order to calculate a path that is diverse with respect to the first LSP. Alternatively, the source node could use an ERO of {C, D, loose Dst} and include an XRO containing the Path Key.

This mechanism can work with all the Path Key resolution mechanisms, as detailed in [RFC5553] section 3.1. A PCE, co-located or not, may be used to resolve the Path Key, but the node (i.e., a Label Switching Router (LSR)) can also use the Path Key information to index a Path Segment previously supplied to it by the entity that originated the Path Key, for example the LSR that inserted the Path Key in the RRO or a management system.

1.3. Network-Assigned Identifier

There are scenarios in which the network provides diversity-related information for a service that allows the client device to include this information in the signaling message. If the

Shared Resource Link Group (SRLG) identifier information is both available and shareable (by policy) with the ENs, the procedure defined in [RFC8001] can be used to collect SRLG identifiers associated with an LSP (LSP1). When a second LSP (LSP2) needs to be diverse with respect to LSP1, the EN constructing the RSVP signaling message for setting up LSP2 can insert the SRLG identifiers associated with LSP1 as diversity constraints into the XRO using the procedure described in [RFC4874]. However, if the core network SRLG identifiers are either not available or not shareable with the ENs based on policies enforced by core network, existing mechanisms cannot be used.

In this draft, a signaling mechanism is defined where information signaled to the CN via the UNI does not require shared knowledge of core network SRLG information. For this purpose, the concept of a Path Affinity Set (PAS) is defined for abstracting SRLG information. The motive behind the introduction of the PAS is to minimize the exchange of diversity information between the core network (CNs) and the client devices (ENs). The PAS contains an abstract SRLG identifier associated with a given path rather than a detailed SRLG list. The PAS is a single identifier that can be used to request diversity and associate diversity. The means by which the processing node determines the path corresponding to the PAS is beyond the scope of this document.

A CN on the core network boundary interprets the specific PAS identifier (e.g. "123") as meaning to exclude the core network SRLG information (or equivalent) that has been allocated by LSPs associated with this PAS identifier value. For example, if a Path exists for the LSP with the PAS identifier "123", the CN would use local knowledge of the core network SRLGs associated with the LSPs tagged with PAS attribute "123" and use those SRLGs as constraints for path computation. If a PAS identifier is used as an exclusion identifier in the connection request, the CN (UNI-N) in the core network is assumed to be able to determine the existing core network SRLG information and calculate a path that meets the determined diversity constraints.

When a CN satisfies a connection setup for a (SRLG) diverse signaled path, the CN may optionally record the core network SRLG information for that connection in terms of CN based parameters and associates that with the EN addresses in the Path message. Specifically, for Layer 1 Virtual Private Networks (L1VPNs), Port Information Tables (PIT) [RFC5251] can be leveraged to translate between client (EN) addresses and core network addresses.

The means to distribute the PAS information within the core network is beyond the scope of this document. For example, the PAS and the associated SRLG information can be distributed within the core network by an Interior Gateway Protocol (IGP) or by other means such as configuration. Regardless of means used to distribute the PAS information, the information is kept inside the core network and is not shared with the overlay network (see Figure 1).

2. RSVP-TE signaling extensions

This section describes the signaling extensions required to address the aforementioned requirements and use cases.

2.1. Diversity XRO Subobject

New Diversity XRO subobjects are defined below for the IPv4 and IPv6 address families. Most of the fields in the IPv4 and IPv6 Diversity XRO subobjects are common and are described following the definition of the two subobjects.

IPv4 Diversity XRO subobject is defined as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|  XRO Type   |     Length    |DI Type|A-Flags|E-Flags| Resvd |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           IPv4 Diversity Identifier Source Address            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Diversity Identifier Value                   |
   //                            ...                              //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Similarly, the IPv6 Diversity XRO subobject is defined as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|  XRO Type   |     Length    |DI Type|A-Flags|E-Flags| Resvd |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           IPv6 Diversity Identifier source address            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         IPv6 Diversity Identifier source address (cont.)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         IPv6 Diversity Identifier source address (cont.)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         IPv6 Diversity Identifier source address (cont.)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Diversity Identifier Value                   |
//                             ...                             //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L:

The L-flag is used in the same way as for the XRO subobjects defined in [RFC4874], i.e.,

0 indicates that the diversity constraints MUST be satisfied.

1 indicates that the diversity constraints SHOULD be satisfied.

XRO Type

The value is set to TBA1 for the IPv4 Diversity XRO subobject (value to be assigned by IANA). The value is set to TBA2 for the IPv6 Diversity XRO subobject (value to be assigned by IANA).

Length

Per [RFC4874], the Length contains the total length of the IPv4/IPv6 subobject in bytes, including the XRO Type and Length fields. The Length is variable, depending on the diversity identifier value.

Diversity Identifier Type (DI Type)

Diversity Identifier Type (DI Type) indicates the way the reference LSP(s) or route(s) with which diversity is required is identified in the IPv4/IPv6 Diversity subobjects. The following three DI type values are defined in this document:

DI Type value	Definition
1	Client Initiated Identifier
2	PCE Allocated Identifier
3	Network Assigned Identifier

Attribute Flags (A-Flags):

The Attribute Flags (A-Flags) are used to communicate desirable attributes of the LSP being signaled in the IPv4/IPv6 Diversity subobjects. Each flag acts independently. Any combination of flags is permitted.

0x01 = Destination node exception

Indicates that the exclusion does not apply to the destination node of the LSP being signaled.

0x02 = Processing node exception

Indicates that the exclusion does not apply to the node(s) performing ERO expansion for the LSP being signaled. An ingress UNI-N node is an example of such a node.

0x04 = Penultimate node exception

Indicates that the penultimate node of the LSP being signaled MAY be shared with the excluded path even when this violates the exclusion flags. This flag is useful, for example, when an EN is not dual-homed (like EN4 in Figure 1 where all LSPs have to go through CN5).

The penultimate node exception flag is typically set when the destination node is single homed (e.g. EN1 or EN4 in Figure 1). In such a case, LSP diversity can only be accomplished inside the core network up to the egress node and the penultimate hop must be the same for the LSPs.

0x08 = LSP ID to be ignored

This flag is used to indicate tunnel level exclusion. Specifically, this flag is used to indicate that if diversity identifier contains LSP ID field, the LSP ID is to be ignored and the exclusion applies to any LSP matching the rest of the diversity identifier.

Exclusion Flags (E-Flags):

The Exclusion Flags are used to communicate the desired type(s) of exclusion requested in the IPv4/IPv6 diversity subobjects. The following flags are defined. Any combination of these flags is permitted. Please note that the exclusion specified by these flags may be modified by the value of the Attribute-flags. For example, node exclusion flag is ignored for the "Penultimate node" if the "Penultimate node exception" flag of the Attribute-flags is set.

0x01 = SRLG exclusion

Indicates that the path of the LSP being signaled is requested to be SRLG disjoint with respect to the excluded path specified by the IPv4/IPv6 Diversity XRO subobject.

0x02 = Node exclusion

Indicates that the path of the LSP being signaled is requested to be node-diverse from the excluded path specified by the IPv4/IPv6 Diversity XRO subobject.

0x04 = Link exclusion

Indicates that the path of the LSP being signaled is requested to be link-diverse from the path specified by the IPv4/IPv6 Diversity XRO subobject.

0x08 = reserved

This flag is reserved. It MUST be set to zero on transmission, and MUST be ignored on receipt for both IPv4/IPv6 Diversity XRO subobjects.

Resvd

This field is reserved. It MUST be set to zero on transmission, and MUST be ignored on receipt for both IPv4/IPv6 Diversity XRO subobjects.

IPv4 / IPv6 Diversity Identifier source address:

This field MUST be set to the IPv4/IPv6 address of the node that assigns the diversity identifier. Depending on the diversity identifier type, the diversity identifier source may be a client node, PCE entity or network node. Specifically:

- When the diversity identifier type is set to "IPv4/IPv6 Client Initiated Identifier", the value MUST be set to IPv4/IPv6 tunnel sender address of the reference LSP against which diversity is desired. IPv4/IPv6 tunnel sender address is as defined in [RFC3209].

- When the diversity identifier type is set to "IPv4/IPv6 PCE Allocated Identifier", the value MUST be set to the IPv4/IPv6 address of the node that assigned the Path Key identifier and that can return an expansion of the Path Key or use the Path Key as exclusion in a path computation. The Path Key is defined in [RFC5553]. The PCE-ID is carried in the Diversity Identifier Source Address field of the subobject.

- When the diversity identifier type is set to "IPv4/IPv6 Network Assigned Identifier", the value MUST be set to the

IPv4/IPv6 address of the node allocating the Path Affinity Set (PAS).

Diversity Identifier Value:

Encoding for this field depends on the diversity identifier type, as defined in the following.

When the diversity identifier type is set to "Client Initiated Identifier" in the IPv4 Diversity XRO subobject, the diversity identifier value MUST be encoded as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 IPv4 tunnel end point address                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |     Tunnel ID                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Extended Tunnel ID                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |            LSP ID             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv4 tunnel end point address, Tunnel ID, Extended Tunnel ID and LSP ID are as defined in [RFC3209].

When the diversity identifier type is set to "Client Initiated Identifier" in the IPv6 Diversity XRO subobject, the diversity identifier value MUST be encoded as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 IPv6 tunnel end point address                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             IPv6 tunnel end point address (cont.)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             IPv6 tunnel end point address (cont.)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             IPv6 tunnel end point address (cont.)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |     Tunnel ID                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Extended Tunnel ID                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Extended Tunnel ID (cont.)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Extended Tunnel ID (cont.)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Extended Tunnel ID (cont.)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Must Be Zero         |            LSP ID             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The IPv6 tunnel end point address, Tunnel ID, IPv6 Extended Tunnel ID and LSP ID are as defined in [RFC3209].

When the diversity identifier type is set to "PCE Allocated Identifier" in IPv4 or IPv6 Diversity XRO subobject, the diversity identifier value MUST be encoded as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Must Be Zero          |           Path Key            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Path Key is defined in [RFC5553].

When the diversity identifier type is set to "Network Assigned Identifier" in IPv4 or IPv6 Diversity XRO subobject, the diversity identifier value MUST be encoded as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Path Affinity Set (PAS) identifier                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Path Affinity Set (PAS) identifier field is a 32-bit value that is scoped by, i.e., is only meaningful when used in combination with, the Diversity Identifier source address field. There are no restrictions on how a node selects a PAS identifier value. Section 1.3 defines the PAS term and provides context on how values may be selected.

2.2. Diversity EXRS Subobject

[RFC4874] defines the EXRS ERO subobject. An EXRS is used to identify abstract nodes or resources that must not or should not be used on the path between two inclusive abstract nodes or resources in the explicit route. An EXRS contains one or more subobjects of its own, called EXRS subobjects [RFC4874].

An EXRS MAY include a Diversity subobject as specified in this document. The same type values TBA1 and TBA2 MUST be used.

2.3. Processing rules for the Diversity XRO and EXRS subobjects

The procedure defined in [RFC4874] for processing the XRO and EXRS is not changed by this document. The processing rules for the Diversity XRO and EXRS subobjects are similar unless the differences are explicitly described. Similarly, IPv4 and IPv6 Diversity XRO subobjects and IPv4 and IPv6 Diversity EXRS subobjects follow the same processing rules.

If the processing node cannot recognize the Diversity XRO/EXRS subobject, the node is expected to follow the procedure defined in [RFC4874].

An XRO/EXRS object MAY contain multiple Diversity subobjects of the same DI Type. E.g., in order to exclude multiple Path Keys, a node MAY include multiple Diversity XRO subobjects each with a different Path Key. Similarly, in order to exclude the routes taken by multiple LSPs, a node MAY include multiple Diversity XRO/EXRS subobjects each with a different LSP identifier. Likewise, to exclude multiple PAS identifiers, a node MAY include multiple Diversity XRO/EXRS subobjects each with a different PAS identifier. However, all Diversity subobjects in an XRO/EXRS MUST contain the same Diversity Identifier Type. If a Path message contains an XRO/EXRS with multiple Diversity subobjects of different DI Types, the processing node MUST return a PathErr with the error code "Routing Problem" (24) and error sub-code "XRO/EXRS Too Complex" (68/69).

If the processing node recognizes the Diversity XRO/EXRS subobject but does not support the DI type, it MUST return a

PathErr with the error code "Routing Problem" (24) and error sub-code "Unsupported Diversity Identifier Type" (TBA3).

In case of DI type "Client Initiated Identifier", all nodes along the path SHOULD process the diversity information signaled in the XRO/EXRS Diversity subobjects to verify that the signaled diversity constraint is satisfied. If a diversity violation is detected, crankback signaling MAY be initiated.

In case of DI type "PCE Allocated Identifier" and "Network Assigned Identifier", the nodes in the domain that perform path computation SHOULD process the diversity information signaled in the XRO/EXRS Diversity subobjects as follows. In the PCE case, the ingress node of a domain sends a path computation request for a path from ingress node to egress node including diversity constraints to a PCE. Or,in the PAS case, the ingress node is capable to calculate the path for the new LSP from ingress node to the egress node taking the diversity constraints into account. The calculated path is then carried in the explicit route object (ERO). Hence, the transit nodes in a domain and the domain egress node SHOULD NOT process the signaled diversity information unless path computation is performed.

While processing EXRS object, if a loose hop expansion results in the creation of another loose hop in the outgoing ERO, the processing node MAY include the EXRS in the newly created loose hop for further processing by downstream nodes.

The Attribute-flags affect the processing of the Diversity XRO/EXRS subobject as follows:

o When the "Processing node exception" flag is set, the exclusion MUST be ignored for the node processing the XRO or EXRS subobject.

o When the "Destination node exception" flag is set, the exclusion MUST be ignored for the destination node in processing the XRO subobject. The destination node exception for the EXRS subobject applies to the explicit node identified by the ERO subobject that identifies the next abstract node. When the "destination node exception" flag is set in the EXRS subobject, exclusion MUST be ignored for the said node (i.e., the next abstract node).

o When the "Penultimate node exception" flag is set in the XRO subobject, the exclusion MUST be ignored for the penultimate node on the path of the LSP being established.

The penultimate node exception for the EXRS subobject applies to the node before the explicit node identified by the ERO subobject that identifies the next abstract node. When the "penultimate node exception" flag is set in the EXRS subobject, the exclusion MUST be ignored for the said node (i.e., the node before the next abstract node).

If the L-flag of the Diversity XRO subobject or Diversity EXRS subobject is not set, the processing node proceeds as follows.

- If the Diversity Identifier Type is set to "Client Initiated Identifier", the processing node MUST ensure that the path calculated/expanded for the signaled LSP is diverse from the route taken by the LSP identified in the Diversity Identifier Value field.

- If the Diversity Identifier Type is set to "PCE Allocated Identifier", the processing node MUST ensure that any path calculated for the signaled LSP is diverse from the route identified by the Path Key. The processing node MAY use the PCE identified by the Diversity Identifier Source Address in the subobject for route computation. The processing node MAY use the Path Key resolution mechanisms described in [RFC5553].

- If the Diversity Identifier Type is set to "Network Assigned Identifier", the processing node MUST ensure that the path calculated for the signaled LSP is diverse with respect to the values associated with the PAS identifier and Diversity Identifier source address fields.

- Regardless of whether the path computation is performed locally or at a remote node (e.g., PCE), the processing node MUST ensure that any path calculated for the signaled LSP is diverse from the requested Exclusion Flags.

- If the excluded path referenced in the XRO subobject is unknown to the processing node, the processing node SHOULD ignore the Diversity XRO subobject and SHOULD proceed with the signaling request. After sending the Resv for the signaled LSP, the processing node MUST return a PathErr with the error code "Notify Error" (25) and error sub-code TBA4 "Route of XRO LSP identifier unknown" (value to be assigned by IANA) for the signaled LSP.

- If the processing node fails to find a path that meets the requested constraint, the processing node MUST return a PathErr with the error code "Routing Problem" (24) and error sub-code "Route blocked by Exclude Route" (67).

If the L-flag of the Diversity XRO subobject or Diversity EXRS subobject is set, the processing node proceeds as follows:

- If the Diversity Identifier Type is set to " Client Initiated Identifiers", the processing node SHOULD ensure that the path calculated/ expended for the signaled LSP is diverse from the route taken by the LSP identified in the Diversity Identifier Value field.

- If the Diversity Identifier Type is set to " PCE Allocated Identifiers", the processing node SHOULD ensure that the path calculated for the signaled LSP is diverse from the route identified by the Path Key.

- If the Diversity Identifier Type is set to "IPv4/IPv6 Network Assigned Identifiers", the processing node SHOULD ensure that the path calculated for the signaled LSP is diverse with respect to the values associated with the PAS identifier and Diversity Identifier source address fields.

- If the processing node fails to find a path that meets the requested constraint, it SHOULD proceed with signaling using a suitable path that meets the constraint as far as possible. After sending the Resv for the signaled LSP, it MUST return a PathErr message with error code "Notify Error" (25) and error sub-code TBA5 "Failed to satisfy Exclude Route" (value: to be assigned by IANA) to the source node.

If, subsequent to the initial signaling of a diverse LSP, an excluded path referenced in the XRO subobject becomes known to the processing node, or a change in the excluded path becomes known to the processing node, the processing node MUST re-evaluate the exclusion and diversity constraints requested by the diverse LSP to determine whether they are still satisfied.

- In case the L-flag was not set in the initial setup message, the exclusion and diversity constraints were satisfied at the time of the initial setup. If the processing node re-evaluating the exclusion and diversity constraints for a diverse LSP detects that the exclusion and diversity constraints are no longer met, it MUST send a PathErr message for the diverse LSP with the error code "Routing Problem" (24) and error sub-code "Route blocked by Exclude Route" (67). The Path_State_Removed flag (PSR) [RFC3473] MUST NOT be set. A source node receiving a

PathErr message with this error code and sub-code combination SHOULD take appropriate actions and move the diverse LSP to a new path that meets the original constraints.

- In case the L-flag was set in the initial setup message, the exclusion and diversity constraints may or may not be satisfied at any given time. If the exclusion constraints for a diverse LSP were satisfied before and if the processing node re-evaluating the exclusion and diversity constraints for a diverse LSP detects that exclusion and diversity constraints are no longer met, it MUST send a PathErr message for the diverse LSP with the error code error code "Notify Error" (25) and error sub-code TBA5 "Failed to satisfy Exclude Route" (value: to be assigned by IANA). The PSR flag MUST NOT be set. The source node MAY take no consequent action and keep the LSP along the path that does not meet the original constraints. Similarly, if the exclusion constraints for a diverse LSP were not satisfied before and if the processing node re-evaluating the exclusion and diversity constraints for a diverse LSP detects that the exclusion constraints are met, it MUST send a PathErr message for the diverse LSP with the error code "Notify Error" (25) and a new error sub- code TBA6 "Compliant path exists" (value: to be assigned by IANA). The PSR flag MUST NOT be set. A source node receiving a PathErr message with this error code and sub-code combination MAY move the diverse LSP to a new path that meets the original constraints.

3. Security Considerations

This document does not introduce any additional security issues in addition to those identified in [RFC5920], [RFC2205], [RFC3209], [RFC3473], [RFC2747], [RFC4874], [RFC5520], and [RFC5553].

The diversity mechanisms defined in this document, rely on the new diversity subobject that is carried in the XRO or EXRS, respectively. In section 7 of [RFC4874], it is noted some administrative boundaries may remove the XRO due to security concerns on explicit route information exchange. However, when the diversity subobjects specified in this document are used, removing at the administrative boundary an XRO containing these diversity subobjects would result in the request for diversity being dropped at the boundary, and path computation would be unlikely to produce the requested diverse path. As such, diversity subobjects MUST be retained in an XRO crossing an administrative boundary, even if other subobjects are removed. This retention would be based on operator policy. The use of diversity subobjects are based on mutual agreement. This avoids the need to share the identity of network resources when supporting diversity.

4. IANA Considerations

IANA is requested to administer the assignment of new values defined in this document and summarized in this section.

4.1. New XRO subobject types

IANA registry: RSVP PARAMETERS
Subsection: Class Names, Class Numbers, and Class Types

This document defines two new subobjects for the EXCLUDE_ROUTE object [RFC4874], C-Type 1. (see: http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml#rsvp-parameters-94)

Subobject Description	Subobject Type
IPv4 Diversity subobject	TBA1
IPv6 Diversity subobject	TBA2

4.2. New EXRS subobject types

The Diversity XRO subobjects are also defined as new EXRS subobjects. (EXPLICIT_ROUTE see: http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml#rsvp-parameters-24). The same numeric subobject type values TBA1 and TBA2 are being requested for the two new EXRS subobjects.

4.3. New RSVP error sub-codes

IANA registry: RSVP PARAMETERS
Subsection: Error Codes and Globally Defined Error Value Sub-Codes.

For Error Code "Routing Problem" (24) (see [RFC3209]) the following sub-codes are defined. (see: http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml#rsvp-parameters-105)

Error Value Sub-codes	Description	Reference
TBA3	Unsupported Diversity Identifier Type	This document

For Error Code "Notify Error" (25) (see [RFC3209]) the following sub-codes are defined. (see: http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml#rsvp-parameters-105)

Error Value Sub-codes	Description	Reference
TBA4	Route of XRO LSP identifier unknown	This document
TBA5	Failed to satisfy Exclude Route	This document
TBA6	Compliant path exists	This document

5. Acknowledgements

The authors would like to thank Xihua Fu for his contributions. The authors also would like to thank Luyuan Fang and Walid Wakim for their review comments.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, R. (Ed.), Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2747] Baker, F., Lindell, B. and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC4202] Kompella, Ed., K, Rekhter, Y, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, April 2007.

[RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, July 2007.

[RFC5553] Farrel, A., Ed., Bradford, R., and JP. Vasseur, "Resource Reservation Protocol (RSVP) Extensions for Path Key Support", RFC 5553, May 2009.

6.2. Informative References

[RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC 4208, October 2005.

[RFC5520] Bradford, R., Ed., Vasseur, JP., and A. Farrel, "Preserving Topology Confidentiality in Inter-Domain Path Computation Using a Path-Key-Based Mechanism", RFC 5520, April 2009.

[RFC8001] F. Zhang, D. Li, O. Gonzalez de Dios, C. Margaria, "RSVP-TE Extensions for Collecting SRLG Information", RFC 8001, January 2017.

[RFC2205] Braden, R. (Ed.), Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC5251] Fedyk, D. (Ed.), Rekhter, Y. (Ed.), Papadimitriou, D., Rabbat, R., and Berger, L., "Layer 1 VPN Basic Mode", RFC 5251, July 2008.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.

Contributors’ Addresses

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

Daniele Ceccarelli
Ericsson
Email: Daniele.Ceccarelli@ericsson.com

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

Oscar Gonzalez de Dios
Telefonica I+D
Email: ogondio@tid.es

Don Fedyk
Hewlett-Packard Enterprise
Email: don.fedyk@hpe.com

Clarence Filsfils
Cisco Systems, Inc.
Email: cfilsfil@cisco.com

Gabriele Maria Galimberti
Cisco Systems
Email: ggalimbe@cisco.com

Ori Gerstel
SDN Solutions Ltd.
Email: origerstel@gmail.com

Matt Hartley
Cisco Systems
Email: mhartley@cisco.com

Kenji Kumaki
KDDI Corporation
Email: ke-kumaki@kddi.com

Ruediger Kunze
Deutsche Telekom AG
Email: Ruediger.Kunze@telekom.de

Lieven Levrau
Nokia
Email: Lieven.Levrau@nokia.com

Cyril Margaria
cyril.margaria@gmail.com

Julien Meuric
France Telecom Orange
Email: julien.meuric@orange.com

Yuji Tochio
Fujitsu
Email: tochio@jp.fujitsu.com

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

Authors’ Addresses

Zafar Ali
Cisco Systems.
Email: zali@cisco.com

Dieter Beller

Nokia
Email: Dieter.Beller@nokia.com

George Swallow
Cisco Systems
Email: swallow@cisco.com

Fatai Zhang
Huawei Technologies
Email: zhangfatai@huawei.com

TEAS Working Group
Internet-Draft
Updates: 3471, 3473, 6205 (if approved)
Intended status: Standards Track
Expires: August 2, 2018

X. Zhang, Ed.
Huawei Technologies
V. Beeram, Ed.
Juniper Networks
I. Bryskin
Huawei Technologies
D. Ceccarelli
Ericsson
O. Gonzalez de Dios
Telefonica

January 29, 2018

Network Assigned Upstream-Label
draft-ietf-teas-network-assigned-upstream-label-12

Abstract

This document discusses a Generalized Multi-Protocol Label Switching (GMPLS) Resource reSerVation Protocol with Traffic Engineering (RSVP-TE) mechanism that enables the network to assign an upstream label for a bidirectional Label Switched Path (LSP). This is useful in scenarios where a given node does not have sufficient information to assign the correct upstream label on its own and needs to rely on the downstream node to pick an appropriate label. This document updates RFCs 3471, 3473 and 6205 as it defines processing for a special label value in the UPSTREAM_LABEL object.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A functional description of the Generalized Multi-Protocol Label Switching (GMPLS) signaling extensions for setting up a bidirectional Label Switched Path (LSP) is provided in [RFC3471]. The GMPLS Resource reSerVation Protocol with Traffic Engineering (RSVP-TE) extensions for setting up a bidirectional LSP are specified in [RFC3473]. The bidirectional LSP setup is indicated by the presence of an UPSTREAM_LABEL object in the Path message. As per the existing setup procedure outlined for a bidirectional LSP, each upstream node must allocate a valid upstream label on the outgoing interface before sending the initial Path message downstream. However, there are certain scenarios (see Section 3) where it is not desirable or possible for a given node to pick the upstream label on its own. This document defines the protocol mechanism to be used in such scenarios. This mechanism enables a given node to offload the task of assigning the upstream label for a given bidirectional LSP to nodes downstream in the network. It is meant to be used only for bidirectional LSPs that assign symmetric labels at each hop along the path of the LSP. Bidirectional Lambda Switch Capable (LSC) LSPs use symmetric lambda labels (format specified in [RFC6205]) at each hop along the path of the LSP.

As per the bidirectional LSP setup procedures specified in [RFC3471] and [RFC3473], the UPSTREAM_LABEL object must indicate a label that is valid for forwarding. This document updates that by allowing the UPSTREAM_LABEL object to indicate a special label that isn’t valid for forwarding. As per the bidirectional LSC LSP setup procedures specified in [RFC6205], the LABEL_SET object and the UPSTREAM_LABEL object must contain the same label value. This document updates that by allowing the UPSREAM_LABEL object to carry a special label value that is different from the one used in the LABEL_SET object.

2. Unassigned Upstream Label

This document defines a special label value - "0xFFFFFFFF" (for a 4-octet label) - to indicate an Unassigned Upstream Label. Similar "all-ones" patterns are expected to be used for labels of other sizes.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1|
   |                              ...                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Unassigned UPSTREAM_LABEL - "all-ones" Pattern

The presence of this value in the UPSTREAM_LABEL object of a Path message indicates that the upstream node has not assigned an upstream label on its own and has requested the downstream node to provide a label that it can use in both the forward and reverse directions. The presence of this value in the UPSTREAM_LABEL object of a Path message MUST also be interpreted by the receiving node as a request to mandate symmetric labels for the LSP.

2.1. Procedures

The scope of the procedures is limited to the exchange and processing of messages between an upstream node and its immediate downstream node. The Unassigned Upstream Label is used by an upstream node when it is not in a position to pick the upstream label on its own. In such a scenario, the upstream node sends a Path message downstream with an Unassigned Upstream Label and requests the downstream node to provide a symmetric label. If the upstream node desires to make the downstream node aware of its limitations with respect to label selection, it MUST specify a list of valid labels via the LABEL_SET object as specified in [RFC3473].

In response, the downstream node picks an appropriate symmetric label and sends it via the LABEL object in the Resv message. The upstream node would then start using this symmetric label for both directions of the LSP. If the downstream node cannot pick the symmetric label, it MUST issue a PathErr message with a "Routing Problem/Unacceptable Label Value" indication. If the upstream node that signals an Unassigned Upstream Label receives a label with the "all-ones" pattern or any other unacceptable label in the LABEL object of the Resv message, it MUST issue a ResvErr message with a "Routing Problem/Unacceptable Label" indication.

The upstream node will continue to signal the Unassigned Upstream Label in the Path message even after it receives an appropriate symmetric label in the Resv message. This is done to make sure that the downstream node would pick a different symmetric label if and when it needs to change the label at a later time. If the upstream node receives an unacceptable changed label, then it MUST issue a ResvErr message with a "Routing Problem/Unacceptable Label" indication.

```
                  +----------+                    +------------+
               ---| Upstream |--------------------| Downstream |---
                  +----------+                    +------------+

                              Path
                               Upstream Label (Unassigned)
                               Label-Set (L1, L2 ... Ln)
                              ------------------->

                              Resv
                               Label (Assigned - L2)
                              <-------------------
```

Figure 2: Signaling Sequence

2.2. Backwards Compatibility

If the downstream node is running an implementation that doesn’t support the semantics of an Unassigned UPSTREAM LABEL, it will either (a) reject the special label value and generate an error as specified in Section 3.1 of [RFC3473] or (b) accept it and treat it as a valid label.

If the behavior that is exhibited is (a), then there are no backwards compatibility concerns. If the behavior that is exhibited is (b), then the downstream node will send a label with the "all-ones" pattern in the LABEL object of the Resv message. In response, the upstream node will issue a ResvErr message with a "Routing Problem/ Unassigned Label" indication.

3. Use-Case: Wavelength Setup for IP over Optical Networks

Consider the network topology depicted in Figure 3. Nodes A and B are client IP routers that are connected to an optical Wavelength Division Multiplexing (WDM) transport network. F and I represent WDM nodes. The transponder sits on the router and is directly connected to the add-drop port on a WDM node.

The optical signal originating on "Router A" is tuned to a particular wavelength. On "WDM-Node F", it gets multiplexed with optical signals at other wavelengths. Depending on the implementation of this multiplexing function, it may not be acceptable to have the router send the signal into the optical network unless it is at the appropriate wavelength. In other words, having the router send signals with a wrong wavelength may adversely impact existing optical trails. If the clients do not have full visibility into the optical network, they are not in a position to pick the correct wavelength in advance.

The rest of this section examines how the protocol mechanism proposed in this document allows the optical network to select and communicate the correct wavelength to its clients.

3.1. Initial Setup

```
         +---+                 /-\             /-\                 +---+
         | A |----------------( F ) ~~~~~~~~~ ( I )----------------| B |
         +---+                 \-/             \-/                 +---+

            Path
              Upstream Label (Unassigned/0xFFFFFFFF)
            ---------------------->
                                  -- ~~ -- ~~ -->
                                                  Path
                                                  --------------------->
                                                  Resv
                                                  <---------------------
                                  <-- ~~ -- ~~ --
            Resv
              Label (Assigned)
            <----------------------
```

Figure 3: Initial Setup Sequence

Steps:

- "Router A" does not have enough information to pick an appropriate client wavelength. It sends a Path message downstream requesting the network to assign an appropriate symmetric label for its use. Since the client wavelength is unknown, the laser is off at the ingress client.

- The downstream node (Node F) receives the Path message, chooses the appropriate wavelength values and forwards them in appropriate label fields to the egress client ("Router B").

- "Router B" receives the Path message, turns the laser ON and tunes it to the appropriate wavelength (received in the UPSTREAM_LABEL/ LABEL_SET of the Path) and sends a Resv message upstream.

- The Resv message received by the ingress client carries a valid symmetric label in the LABEL object. "Router A" turns on the laser and tunes it to the wavelength specified in the network assigned symmetric LABEL.

For cases where the egress-node relies on RSVP signaling to determine exactly when to start using the LSP, implementations may choose to integrate the above sequence with any of the existing graceful setup procedures:

- "ResvConf" setup procedure ([RFC2205])

- 2-step "ADMIN STATUS" based setup procedure ("A" bit set in the first step; "A" bit cleared when the LSP is ready for use). ([RFC3473])

3.2. Wavelength Change

After the LSP is set up, the network may decide to change the wavelength for the given LSP. This could be for a variety of reasons including policy reasons, restoration within the core, preemption, etc.

In such a scenario, if the ingress client receives a changed label via the LABEL object in a modified Resv message, it retunes the laser at the ingress to the new wavelength. Similarly, if the egress client receives a changed label via UPSTREAM_LABEL/LABEL_SET in a modified Path message, it retunes the laser at the egress to the new wavelength.

4. IANA Considerations

IANA maintains the "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Parameters" registry. IANA is requested to add a new subregistry for "Special Purpose Generalized Label Values". New values are assigned according to Standards Action.

Special Purpose Generalized Label Values

Pattern/ Value	Label Name	Applicable Objects	Reference
all-ones	Unassigned Upstream Label	UPSTREAM_LABEL	[This.I-D]

5. Security Considerations

This document defines a special label value to be carried in the UPSTREAM_LABEL object of a Path message. This special label value is used to enable the function of requesting network assignment of an upstream label. The changes proposed in this document pertain to the semantics of a specific field in an existing RSVP object and the corresponding procedures. Thus, there are no new security implications raised by this document and the security considerations discussed by [RFC3473] still apply.

For a general discussion on MPLS and GMPLS related security issues, see the MPLS/GMPLS security framework [RFC5920].

6. Acknowledgements

The authors would like to thank Adrian Farrel and Chris Bowers for their inputs.

7. Contributors

John Drake
Juniper Networks
Email: jdrake@juniper.net

Gert Grammel
Juniper Networks
Email: ggrammel@juniper.net

Pawel Brzozowski
ADVA Optical Networking
Email: pbrzozowski@advaoptical.com

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

8. References

8.1. Normative References

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

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

[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>.

[RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <https://www.rfc-editor.org/info/rfc6205>.

[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>.

8.2. Informative References

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

Authors’ Addresses

Xian Zhang (editor)
Huawei Technologies

Email: zhang.xian@huawei.com

Vishnu Pavan Beeram (editor)
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin
Huawei Technologies

Email: igor.bryskin@huawei.com

Daniele Ceccarelli
Ericsson

Email: daniele.ceccarelli@ericsson.com

Oscar Gonzalez de Dios
Telefonica

Email: ogondio@tid.es

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 18, 2018

V. Beeram, Ed.
Juniper Networks
I. Minei
R. Shakir
Google, Inc
D. Pacella
Verizon
T. Saad
Cisco Systems

February 14, 2018

Techniques to Improve the Scalability of RSVP Traffic Engineering Deployments
draft-ietf-teas-rsvp-te-scaling-rec-09

Abstract

At the time of writing, networks which utilize RSVP Traffic Engineering (RSVP-TE) Label Switched Paths (LSPs) are encountering limitations in the ability of implementations to support the growth in the number of LSPs deployed.

This document defines two techniques, "Refresh-Interval Independent RSVP (RI-RSVP)" and "Per-Peer Flow-Control", that reduce the number of processing cycles required to maintain RSVP-TE LSP state in Label Switching Routers (LSRs) and hence allow implementations to support larger scale deployments.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

At the time of writing, networks which utilize RSVP Traffic Engineering (RSVP-TE) [RFC3209] Label Switched Paths (LSPs) are encountering limitations in the ability of implementations to support the growth in the number of LSPs deployed.

The set of RSVP Refresh Overhead Reduction procedures [RFC2961] serves as a powerful toolkit for RSVP-TE implementations to help cover a majority of the concerns about soft-state scaling. However, even with these tools in the toolkit, analysis of existing implementations [RFC5439] indicates that the processing required beyond a certain scale may still cause significant disruption to a Label Switching Router (LSR).

This document builds on the scaling work and analysis that has been done so far and defines protocol extensions to help RSVP-TE deployments push the envelope further on scaling by increasing the threshold above which an LSR struggles to achieve sufficient processing to maintain LSP state.

This document defines two techniques, "Refresh-Interval Independent RSVP (RI-RSVP)" and "Per-Peer Flow-Control", that cut down the number of processing cycles required to maintain LSP state. "RI-RSVP" helps completely eliminate RSVP’s reliance on refreshes and refresh-timeouts while "Per-Peer Flow-Control" enables a busy RSVP speaker to apply back pressure to its peer(s). This document defines a unique RSVP Capability [RFC5063] for each technique (Support for CAPABILITY object is a prerequisite for implementing these techniques). Note that the "Per-Peer Flow-Control" technique requires the "RI-RSVP" technique as a prerequisite. In order to reap maximum scaling benefits, it is strongly recommended that implementations support both the techniques and have them enabled by default. Both the techniques are fully backward compatible and can be deployed incrementally.

2. Requirement for RFC2961 Support

The techniques defined in Section 3 and Section 4 are based on proposals made in [RFC2961]. Implementations of these techniques will need to support the RSVP messages and procedures defined in [RFC2961] with some minor modifications and alterations to recommended time intervals and iteration counts (see Appendix A for the set of recommended defaults).

2.1. Required Functionality from RFC2961 to be Implemented

An implementation that supports the techniques discussed in Section 3 and Section 4 must support the functionality described in [RFC2961] as follows:

- o It MUST indicate support for RSVP Refresh Overhead Reduction extensions (as specified in Section 2 of [RFC2961]).
- o It MUST support receipt of any RSVP Refresh Overhead Reduction message as defined in [RFC2961].
- o It MUST initiate all RSVP Refresh Overhead Reduction mechanisms as defined in [RFC2961] (including the SRefresh message) with the default behavior being to initiate the mechanisms but offering a configuration override.
- o It MUST support reliable delivery of Path/Resv and the corresponding Tear/Err messages (as specified in Section 4 of [RFC2961]).
- o It MUST support retransmission of all unacknowledged RSVP-TE messages using exponential-backoff (as specified in Section 6 of [RFC2961]).

2.2. Making Acknowledgements Mandatory

The reliable message delivery mechanism specified in [RFC2961] states that "Nodes receiving a non-out of order message containing a MESSAGE_ID object with the ACK_Desired flag set, SHOULD respond with a MESSAGE_ID_ACK object."

In an implementation that supports the techniques discussed in Section 3 and Section 4, nodes receiving a non-out of order message containing a MESSAGE_ID object with the ACK-Desired flag set, MUST respond with a MESSAGE_ID_ACK object. This MESSAGE_ID_ACK object can be packed along with other MESSAGE_ID_ACK or MESSAGE_ID_NACK objects and sent in an Ack message (or piggy-backed in any other RSVP message). This improvement to the predictability of the system in terms of reliable message delivery is key for being able to take any action based on a non-receipt of an ACK.

3. Refresh-Interval Independent RSVP (RI-RSVP)

The RSVP protocol relies on periodic refreshes for state synchronization between RSVP neighbors and for recovery from lost RSVP messages. It relies on refresh timeout for stale state cleanup. The primary motivation behind introducing the notion of "Refresh

Interval Independent RSVP" (RI-RSVP) is to completely eliminate RSVP’s reliance on refreshes and refresh timeouts. This is done by simply increasing the refresh interval to a fairly large value. [RFC2961] and [RFC5439] do talk about increasing the value of the refresh interval to provide linear improvement of transmission overhead, but also point out the degree of functionality that is lost by doing so. This section revisits this notion, but also sets out additional requirements to make sure that there is no loss of functionality incurred by increasing the value of the refresh interval.

An implementation that supports RI-RSVP:

- MUST support all the requirements specified in Section 2.

- MUST make the default value of the configurable refresh interval (R) be a large value (10s of minutes). A default value of 20 minutes is RECOMMENDED by this document.

- MUST use a separate shorter refresh interval for refreshing state associated with unacknowledged Path/Resv messages (uR). A default value of 30 seconds is RECOMMENDED by this document.

- MUST implement coupling the state of individual LSPs with the state of the corresponding RSVP-TE signaling adjacency. When an RSVP-TE speaker detects RSVP-TE signaling adjacency failure, the speaker MUST act as if all the Path and Resv states learnt via the failed signaling adjacency have timed out.

- MUST make use of Node-ID based Hello Session ([RFC3209], [RFC4558]) for detection of RSVP-TE signaling adjacency failures; A default value of 9 seconds is RECOMMENDED by this document for the configurable node hello interval (as opposed to the 5ms default value proposed in Section 5.3 of [RFC3209]).

- MUST indicate support for RI-RSVP via the CAPABILITY object [RFC5063] in Hello messages.

3.1. Capability Advertisement

An implementation supporting the RI-RSVP technique MUST set a new flag "RI-RSVP Capable" in the CAPABILITY object signaled in Hello messages.

Bit Number TBA1 (TBA2) - RI-RSVP Capable (I-bit):

Indicates that the sender supports RI-RSVP.

Any node that sets the new I-bit in its CAPABILITY object MUST also set the Refresh-Reduction-Capable bit in the common header of all RSVP-TE messages. If a peer sets the I-bit in the CAPABILITY object but does not set the Refresh-Reduction-Capable bit, then the RI-RSVP functionality MUST NOT be activated for that peer.

3.2. Compatibility

The RI-RSVP functionality MUST NOT be activated with a peer that does not indicate support for this functionality. Inactivation of the RI-RSVP functionality MUST result in the use of the traditional smaller refresh interval [RFC2205].

4. Per-Peer RSVP Flow-Control

The functionality discussed in this section provides an RSVP speaker with the ability to apply back pressure to its peer(s) to reduce/eliminate a significant portion of the RSVP-TE control message load.

An implementation that supports "Per-Peer RSVP Flow-Control":

- MUST support all the requirements specified in Section 2.

- MUST support "RI-RSVP" (Section 3).

- MUST treat lack of ACKs from a peer as an indication of peer’s RSVP-TE control plane congestion. If congestion is detected, the local system MUST throttle RSVP-TE messages to the affected peer. This MUST be done on a per-peer basis. (Per-peer throttling MAY be implemented by a traffic shaping mechanism that proportionally reduces the RSVP signaling packet rate as the number of outstanding Acks increases. And when the number of outstanding Acks decreases, the send rate would be adjusted up again.)

- SHOULD use a Retry Limit (Rl) value of 7 (Section 6.2 of [RFC2961], suggests using 3).

- SHOULD prioritize Hello messages and messages carrying Acknowledgements over other RSVP messages.

- SHOULD prioritize Tear/Error over trigger Path/Resv (messages that bring up new LSP state) sent to a peer when the local system detects RSVP-TE control plane congestion in the peer.

- MUST indicate support for this technique via the CAPABILITY object [RFC5063] in Hello messages.

4.1. Capability Advertisement

An implementation supporting the "Per-Peer Flow-Control" technique MUST set a new flag "Per-Peer Flow-Control Capable" in the CAPABILITY object signaled in Hello messages.

Bit Number TBA3 (TBA4) - Per-Peer Flow-Control Capable (F-bit):

Indicates that the sender supports Per-Peer RSVP Flow-Control.

Any node that sets the new F-bit in its CAPABILITY object MUST also set Refresh-Reduction-Capable bit in common header of all RSVP-TE messages. If a peer sets the F-bit in the CAPABILITY object but does not set the Refresh-Reduction-Capable bit, then the Per-Peer Flow-Control functionality MUST NOT be activated for that peer.

4.2. Compatibility

The Per-Peer Flow-Control functionality MUST NOT be activated with a peer that does not indicate support for this functionality. If a peer hasn’t indicated that it is capable of participating in "Per-Peer Flow-Control", then it SHOULD NOT be assumed that the peer would always acknowledge a non-out of order message containing a MESSAGE_ID object with the ACK-Desired flag set.

5. Acknowledgements

The authors would like to thank Yakov Rekhter for initiating this work and providing valuable inputs. They would like to thank Raveendra Torvi and Chandra Ramachandran for participating in the many discussions that led to the techniques discussed in this document. They would also like to thank Adrian Farrel, Lou Berger and Elwyn Davies for providing detailed review comments and text suggestions.

6. Contributors

Markus Jork
Juniper Networks
Email: mjork@juniper.net

Ebben Aries
Juniper Networks
Email: exa@juniper.net

7. IANA Considerations

7.1. Capability Object Values

IANA maintains all the registries associated with "Resource Reservation Protocol (RSVP) Paramaters" (see http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml). "Capability Object Values" Registry (introduced by [RFC5063]) is one of them.

IANA is requested to assign two new Capability Object Value bit flags as follows:

Bit Number	Hex Value	Name	Reference
TBA1	TBA2	RI-RSVP Capable (I)	Section 3
TBA3	TBA4	Per-Peer Flow-Control Capable (F)	Section 4

8. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

9. References

9.1. Normative References

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <https://www.rfc-editor.org/info/rfc2961>.

[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>.

[RFC4558] Ali, Z., Rahman, R., Prairie, D., and D. Papadimitriou, "Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement", RFC 4558, DOI 10.17487/RFC4558, June 2006, <https://www.rfc-editor.org/info/rfc4558>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <https://www.rfc-editor.org/info/rfc5063>.

[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>.

9.2. Informative References

[RFC5439] Yasukawa, S., Farrel, A., and O. Komolafe, "An Analysis of Scaling Issues in MPLS-TE Core Networks", RFC 5439, DOI 10.17487/RFC5439, February 2009, <https://www.rfc-editor.org/info/rfc5439>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

Appendix A. Recommended Defaults

(a) Refresh-Interval (R)- 20 minutes (Section 3):
Given that an implementation supporting RI-RSVP doesn’t rely on refreshes for state sync between peers, the function of RSVP refresh interval is analogous to that of IGP refresh interval (the default of which is typically in the order of 10s of minutes). Choosing a default of 20 minutes allows the refresh timer to be randomly set to a value in the range [10 minutes (0.5R), 30 minutes (1.5R)].

(b) Node Hello-Interval - 9 Seconds (Section 3):
[RFC3209] defines the hello timeout as 3.5 times the hello interval. Choosing 9 seconds for the node hello-interval gives a hello timeout of 3.5*9 = 31.5 seconds. This puts the hello timeout value in the vicinity of the IGP hello timeout value.

(c) Retry-Limit (Rl) - 7 (Section 4):
Choosing 7 as the retry-limit results in an overall rapid retransmit phase of 31.5 seconds. This matches up with the 31.5 seconds hello timeout.

(d) Refresh-Interval for refreshing state associated with unacknowledged Path/Resv messages (uR) - 30 seconds (Section 3):
The recommended refresh interval (R) value of 20 minutes (for an implementation supporting RI-RSVP) can not be used for refreshing state associated with unacknowledged Path/Resv messages. This document recommends the use of the traditional default refresh interval value of 30 seconds for uR.

Authors’ Addresses

Vishnu Pavan Beeram (editor)
Juniper Networks

Email: vbeeram@juniper.net

Ina Minei
Google, Inc

Email: inaminei@google.com

Rob Shakir
Google, Inc

Email: rjs@rob.sh

Dante Pacella
Verizon

Email: dante.j.pacella@verizon.com

Tarek Saad
Cisco Systems

Email: tsaad@cisco.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 31 August 2024

V. P. Beeram
Juniper Networks
T. Saad
Cisco Systems Inc
R. Gandhi
Cisco Systems, Inc.
X. Liu
Volta Networks
I. Bryskin
Individual

28 February 2024

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-19

Abstract

This document defines a YANG data model for the configuration and management of the RSVP protocol. The YANG data model covers the building blocks that may be augmented by other RSVP extension data models such as RSVP Traffic-Engineering (RSVP-TE). It is divided into two modules that cover the basic and extended RSVP features.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

This document defines a YANG data model for the configuration and management of the RSVP protocol [RFC2205]. The data model is divided into two modules: a base and extended RSVP YANG modules. The RSVP base YANG 'ietf-rsvp' module covers the data that is core to the function of the RSVP protocol and MUST be supported by vendors that support RSVP protocol [RFC2205]. The RSVP extended 'ietf-rsvp-extended' module covers the data that is optional, or provides ability to tune RSVP protocol base functionality. The support for RSVP extended module features by vendors is considered optional.

The RSVP YANG model provides the building blocks needed to allow augmentation by other models that extend the RSVP protocol-- such as using RSVP extensions to signal Label Switched Paths (LSPs) as defined in [RFC3209].

The YANG module(s) defined in this document are compatible with the Network Management Datastore Architecture (NMDA) [RFC7950].

2. Requirements Language

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

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

2.1. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
rt	ietf-routing	[RFC8349]
rt-types	ietf-routing-types	[RFC8294]
inet	ietf-inet-types	[RFC6991]
yang	ietf-yang-types	[RFC6991]
key-chain	ietf-key-chain	[RFC8177]

Table 1: Prefixes and corresponding YANG modules

2.2. Model Tree Diagram

A full tree diagram of the module(s) defined in this document is given in subsequent sections as per the syntax defined in [RFC8340].

3. Model Overview

The RSVP YANG module augments the "control-plane-protocol" entry from the ’ietf-routing’ module defined in [RFC8349]. It also defines the identity "rsvp" of base type "rt:routing-protocol" to identify the RSVP routing protocol.

The ’ietf-rsvp’ model defines a single instance of the RSVP protocol. The top ’rsvp’ container encompases data for one such RSVP protocol instance. Multiple instances can be defined as multiple control-plane protocols instances as described in [RFC8349].

The YANG data model defined has the common building blocks for the operation of the base RSVP protocol for the session type defined in [RFC2205]. The augmentation of this model by other models (e.g. to support RSVP Traffic Engineering (TE) extensions for signaling Label Switched Paths (LSPs)) are outside the scope of this document and are discussed in separate document(s).

3.1. Module(s) Relationship

This RSVP YANG data model defined in this document is divided into two modules: a base and extended modules. The RSVP data covered in 'ietf-rsvp' module are categorized as core to the function of the protocol and MUST be supported by vendors claiming the support for RSVP protocol [RFC2205].

The RSVP extended features that are covered in 'ietf-rsvp-extended' module are categorized as either optional or providing ability to better tune the basic functionality of the RSVP protocol. The support for RSVP extended features by all vendors is considered optional.

The relationship between the base and RSVP extended YANG modules and the IETF routing YANG model is shown in Figure 1.

```
                  +---------------+
       Routing    | ietf-routing |
                  +---------------+
                        ^
                        |
                   +-----------+
  RSVP module      | ietf-rsvp |
                   +-----------+
                        ^
                        |                   ^: augment relationship
  RSVP extended         |
    module         +--------------------+
                   | ietf-rsvp-extended |
                   +--------------------+
```

Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

3.2. Core Features

The RSVP data covered in the 'ietf-rsvp' YANG module provides the common building blocks that are required to configure, operate and manage the RSVP protocol and MUST be supported by vendors that claim the support for base RSVP protocol defined in [RFC2205].

In addition, the following standard RSVP core features are modeled under the 'ietf-rsvp' module:

* Basic operational statistics, including protocol messages, packets and errors.

* Basic RSVP authentication feature as defined in [RFC2747]) using string based authentication key.

* Basic RSVP Refresh Reduction feature as defined in ([RFC2961]).

* Basic RSVP Hellos feature as defined in ([RFC3209])

* Basic RSVP Graceful Restart feature as defined in [RFC3473], [RFC5063], and [RFC5495].

3.3. Optional Features

Optional features are beyond the basic configuration, and operation of the RSVP protocol. The decision whether to support these RSVP features on a particular device is left to the vendor that supports the RSVP core features.

The following optional features that are covered in the 'ietf-rsvp-extended' YANG module:

* Advanced operational statistics, including protocol messages, packets and errors.

* Advanced RSVP authentication features as defined in [RFC2747]) using various authentication key types including those defined in [RFC8177].

* Advanced RSVP Refresh Reduction features defined in ([RFC2961]).

* Advanced RSVP Hellos features as defined in [RFC3209], and [rfc4558].

* Advanced RSVP Graceful Restart features as defined in [RFC3473], [RFC5063], and [RFC5495].

3.4. Data Model Structure

The RSVP YANG data model defines the 'rsvp' top-level container that contains the configuration and operational state for the RSVP protocol. The presence of this container enables the RSVP protocol functionality.

The 'rsvp' top-level container also includes data that has router level scope (i.e. applicable to all objects modeled under rsvp). It also contains configuration and state data about the following types of RSVP objects:

* interfaces

- neighbors

- sessions

The derived state data is contained in "read-only" nodes directly under the intended object as shown in Figure 2.

```
module: ietf-rsvp
   +--rw rsvp!
      +--rw <<router-level scope data>>
         .
         .
      +--rw interfaces
            .
            +-- ro <<derived state associated with interfaces>>
         .
         .
      +--rw neighbors
            .
            +-- ro <<derived state associated with the LSP Tunnel>>
         .
         .
      +--rw sessions
            .
            +-- ro <<derived state associated with the LSP Tunnel>>
         .
   rpcs:
      +--x clear-session
      +--x clear-neighbor
      +--x clear-authentication
```

Figure 2: RSVP high-level tree model view

The following

'router-level':

The router-level scope configuration and state data are applicable to all modeled objects under the top-level 'rsvp' container, and MAY affect the RSVP protocol behavior.

'interfaces':

The ’interfaces’ container includes a list of RSVP enabled interfaces. It also includes RSVP configuration and state data that is applicable to all interfaces. An entry in the interfaces list MAY carry its own configuration or state data. Any data or state under the "interfaces" container level is equally applicable to all interfaces unless it is explicitly overridden by configuration or state under a specific interface.

’neighbors’ :

The ’neighbors’ container includes a list of RSVP neighbors. An entry in the RSVP neighbor list MAY carry its own configuration and state relevant to the specific RSVP neighbor. The RSVP neighbors can be dynamically discovered using RSVP signaling, or can be explicitly configured.

’sessions’:

The ’sessions’ container includes a list RSVP sessions. An entry in the RSVP session list MAY carry its own configuration and state relevant to a specific RSVP session. RSVP sessions are usually derived state that are created as result of signaling. This model defines attributes related to IP RSVP sessions as defined in [RFC2205].

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

3.5. Model Notifications

Modeling notifications data is key in any defined YANG data model. [RFC8639] and [RFC8641] define a subscription and push mechanism for YANG datastores. This mechanism currently allows the user to:

- Subscribe notifications on a per client basis

- Specify subtree filters [RFC6241] or XPath filters [RFC8639] so that only interested contents will be sent.

- Specify either periodic or on-demand notifications.

4. RSVP Base YANG Model

The RSVP base module includes the core features and building blocks for modeling the RSVP protocol as described in Section 3.2.

4.1. Tree Diagram

Figure 3 shows the YANG tree representation for configuration, state data and RPCs that are covered in 'ietf-rsvp' YANG module:

```
   module: ietf-rsvp

     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol:
       +--rw rsvp!
          +--rw interfaces
          |  +--rw refresh-reduction
          |  |  +--rw enabled?   boolean
          |  +--rw hellos
          |  |  +--rw enabled?   boolean
          |  +--rw authentication
          |  |  +--rw enabled?              boolean
          |  |  +--rw authentication-key?   string
          |  |  +--rw crypto-algorithm?     identityref
          |  +--ro statistics
          |  |  +--ro messages
          |  |  |  +--ro ack-sent?                       yang:counter64
          |  |  |  +--ro ack-received?                   yang:counter64
          |  |  |  +--ro bundle-sent?                    yang:counter64
          |  |  |  +--ro bundle-received?                yang:counter64
          |  |  |  +--ro hello-sent?                     yang:counter64
          |  |  |  +--ro hello-received?                 yang:counter64
          |  |  |  +--ro integrity-challenge-sent?       yang:counter64
          |  |  |  +--ro integrity-challenge-received?   yang:counter64
          |  |  |  +--ro integrity-response-sent?        yang:counter64
          |  |  |  +--ro integrity-response-received?    yang:counter64
          |  |  |  +--ro notify-sent?                    yang:counter64
          |  |  |  +--ro notify-received?                yang:counter64
          |  |  |  +--ro path-sent?                      yang:counter64
          |  |  |  +--ro path-received?                  yang:counter64
          |  |  |  +--ro path-err-sent?                  yang:counter64
          |  |  |  +--ro path-err-received?              yang:counter64
          |  |  |  +--ro path-tear-sent?                 yang:counter64
          |  |  |  +--ro path-tear-received?             yang:counter64
          |  |  |  +--ro resv-sent?                      yang:counter64
          |  |  |  +--ro resv-received?                  yang:counter64
          |  |  |  +--ro resv-confirm-sent?              yang:counter64
          |  |  |  +--ro resv-confirm-received?          yang:counter64
          |  |  |  +--ro resv-err-sent?                  yang:counter64
          |  |  |  +--ro resv-err-received?              yang:counter64
          |  |  |  +--ro resv-tear-sent?                 yang:counter64
          |  |  |  +--ro resv-tear-received?             yang:counter64
          |  |  |  +--ro srefresh-sent?                  yang:counter64
          |  |  |  +--ro srefresh-received?              yang:counter64
          |  |  |  +--ro unknown-messages-received?      yang:counter64
          |  |  +--ro packets
          |  |  |  +--ro sent?       yang:counter64
          |  |  |  +--ro received?   yang:counter64
          |  |  +--ro errors
          |  |     +--ro authenticate?    yang:counter64
          |  |     +--ro checksum?        yang:counter64
          |  |     +--ro packet-length?   yang:counter64
          |  +--rw interface* [name]
          |     +--rw name                 if:interface-ref
          |     +--rw refresh-reduction
          |     |  +--rw enabled?   boolean
          |     +--rw hellos
          |     |  +--rw enabled?   boolean
          |     +--rw authentication
          |     |  +--rw enabled?              boolean
          |     |  +--rw authentication-key?   string
          |     |  +--rw crypto-algorithm?     identityref
          |     +--ro statistics
          |        +--ro messages
          |        |  +--ro ack-sent?
          |        |  |       yang:counter64
          |        |  +--ro ack-received?
          |        |  |       yang:counter64
          |        |  +--ro bundle-sent?
          |        |  |       yang:counter64
          |        |  +--ro bundle-received?
          |        |  |       yang:counter64
          |        |  +--ro hello-sent?
          |        |  |       yang:counter64
          |        |  +--ro hello-received?
          |        |  |       yang:counter64
          |        |  +--ro integrity-challenge-sent?
          |        |  |       yang:counter64
          |        |  +--ro integrity-challenge-received?
          |        |  |       yang:counter64
          |        |  +--ro integrity-response-sent?
          |        |  |       yang:counter64
          |        |  +--ro integrity-response-received?
          |        |  |       yang:counter64
          |        |  +--ro notify-sent?
          |        |  |       yang:counter64
          |        |  +--ro notify-received?
          |        |  |       yang:counter64
          |        |  +--ro path-sent?
          |        |  |       yang:counter64
          |        |  +--ro path-received?
          |        |  |       yang:counter64
          |        |  +--ro path-err-sent?
          |        |  |       yang:counter64
          |        |  +--ro path-err-received?
          |        |  |       yang:counter64
          |        |  +--ro path-tear-sent?
          |        |  |       yang:counter64
          |        |  +--ro path-tear-received?
          |        |  |       yang:counter64
          |        |  +--ro resv-sent?
          |        |  |       yang:counter64
          |        |  +--ro resv-received?
          |        |  |       yang:counter64
          |        |  +--ro resv-confirm-sent?
          |        |  |       yang:counter64
          |        |  +--ro resv-confirm-received?
          |        |  |       yang:counter64
          |        |  +--ro resv-err-sent?
          |        |  |       yang:counter64
          |        |  +--ro resv-err-received?
          |        |  |       yang:counter64
          |        |  +--ro resv-tear-sent?
          |        |  |       yang:counter64
          |        |  +--ro resv-tear-received?
          |        |  |       yang:counter64
          |        |  +--ro srefresh-sent?
          |        |  |       yang:counter64
          |        |  +--ro srefresh-received?
          |        |  |       yang:counter64
          |        |  +--ro unknown-messages-received?
          |        |          yang:counter64
          |        +--ro packets
          |        |  +--ro sent?       yang:counter64
          |        |  +--ro received?   yang:counter64
          |        +--ro errors
          |           +--ro authenticate?    yang:counter64
          |           +--ro checksum?        yang:counter64
          |           +--ro packet-length?   yang:counter64
          +--rw sessions
          |  +--ro session-ip*
          |          [destination protocol-id destination-port]
          |     +--ro destination-port    uint16
          |     +--ro protocol-id         uint8
          |     +--ro source?             inet:ip-address
          |     +--ro destination         inet:ip-address
          |     +--ro session-name?       string
          |     +--ro session-status?     enumeration
          |     +--ro session-type        identityref
          |     +--ro psbs
          |     |  +--ro psb* []
          |     |     +--ro source-port?   inet:port-number
          |     |     +--ro expires-in?    uint32
          |     +--ro rsbs
          |        +--ro rsb* []
          |           +--ro source-port?         inet:port-number
          |           +--ro reservation-style?   identityref
          |           +--ro expires-in?          uint32
          +--rw neighbors
          |  +--rw neighbor* [address]
          |     +--rw address                      inet:ip-address
          |     +--rw epoch?                       uint32
          |     +--rw expiry-time?                 uint32
          |     +--rw graceful-restart
          |     |  +--ro neighbor-restart-time?    uint32
          |     |  +--ro neighbor-recovery-time?   uint32
          |     |  +--ro helper-mode
          |     |     +--ro neighbor-restart-time-remaining?    uint32
          |     |     +--ro neighbor-recovery-time-remaining?   uint32
          |     +--ro hello-status?                enumeration
          |     +--rw interface?                   if:interface-ref
          |     +--ro neighbor-status?             enumeration
          |     +--rw refresh-reduction-capable?   boolean
          |     +--ro restart-count?               yang:counter32
          |     +--ro restart-time?                yang:date-and-time
          +--rw graceful-restart
             +--rw enabled?               boolean
             +--rw local-restart-time?    uint32
             +--rw local-recovery-time?   uint32
             +--rw helper-mode
                +--rw enabled?                    boolean
                +--rw max-helper-restart-time?    uint32
                +--rw max-helper-recovery-time?   uint32

     rpcs:
       +---x clear-session
       |  +---w input
       |     +---w routing-protocol-instance-name    leafref
       |     +---w (filter-type)
       |        +--:(match-all)
       |        |  +---w all                         empty
       |        +--:(match-one)
```

```
|           +---w session-info
|              +---w (session-type)
|                 +--:(rsvp-session-ip)
|                    +---w destination          leafref
|                    +---w protocol-id          uint8
|                    +---w destination-port     inet:ip-address
+---x clear-neighbor
|  +---w input
|     +---w routing-protocol-instance-name     leafref
|     +---w (filter-type)
|        +--:(match-all)
|        |  +---w all                          empty
|        +--:(match-one)
|           +---w neighbor-address             leafref
+---x clear-authentication
   +---w input
      +---w routing-protocol-instance-name     leafref
      +---w (filter-type)
         +--:(match-all)
         |  +---w all                          empty
         +--:(match-one-interface)
            +---w interface?                   if:interface-ref
```

Figure 3: RSVP model tree diagram

4.2. YANG Module

The ietf-rsvp module imports from the following modules:

* ietf-interfaces defined in [RFC8343]

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

* ietf-routing defined in [RFC8349]

* ietf-key-chain defined in [RFC8177]

* ietf-netconf-acm defined in [RFC8341]

This module also references the following documents: [RFC2205], [RFC5495], [RFC3473], [RFC5063], [RFC2747], [RFC3209], and [RFC2961].

```
<CODE BEGINS> file "ietf-rsvp@2024-02-28.yang"
module ietf-rsvp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp";

  /* Replace with IANA when assigned */
```

```
     prefix rsvp;

     import ietf-interfaces {
       prefix if;
       reference
         "RFC8343: A YANG Data Model for Interface Management";
     }
     import ietf-inet-types {
       prefix inet;
       reference
         "RFC6991: Common YANG Data Types";
     }
     import ietf-yang-types {
       prefix yang;
       reference
         "RFC6991: Common YANG Data Types";
     }
     import ietf-routing {
       prefix rt;
       reference
         "RFC8349: A YANG Data Model for Routing Management
          (NMDA Version)";
     }
     import ietf-key-chain {
       prefix key-chain;
       reference
         "RFC8177: YANG Data Model for Key Chains";
     }
     import ietf-netconf-acm {
       prefix nacm;
       reference
         "RFC8341: Network Configuration Access Control Model";
     }
     organization
       "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";
     contact
       "WG Web:   <http://tools.ietf.org/wg/teas/>
        WG List:  <mailto:teas@ietf.org>

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

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

        Editor:   Rakesh Gandhi
                  <mailto:rgandhi@cisco.com>
        Editor:   Xufeng Liu
                  <mailto: xufeng.liu.ietf@gmail.com>

        Editor:   Igor Bryskin
                  <mailto:i_bryskin@yahoo.com>";
     description
       "This module contains the RSVP YANG data model.
        The model fully conforms to the Network Management Datastore
        Architecture (NMDA).

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

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

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

     revision 2024-02-28 {
       description
         "Initial version.";
       reference
         "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
          (RSVP)";
     }

     identity rsvp {
       base rt:routing-protocol;
       description
         "RSVP protocol";
     }

     identity rsvp-session-type {
       description
         "Base RSVP session type";
     }

     identity rsvp-session-ip {
       base rsvp-session-type;
       description
         "RSVP IP session type";
     }

     identity reservation-style {
       description
         "Base identity for reservation style.";
     }

     identity reservation-wildcard-filter {
       base reservation-style;
       description
         "Wildcard-Filter (WF) Style.";
       reference
         "RFC2205";
     }

     identity reservation-fixed-filter {
       base reservation-style;
       description
         "Fixed-Filter (FF) Style.";
       reference
         "RFC2205";
     }

     identity reservation-shared-explicit {
       base reservation-style;
       description
         "Shared Explicit (SE) Style.";
       reference
         "RFC2205";
     }

     grouping intf-attributes {
       description
         "Top level grouping for RSVP interface properties.";
       container refresh-reduction {
         description
           "Top level container for RSVP refresh reduction parameters.";
         leaf enabled {
           type boolean;
           default "true";
           description
             "'true' if RSVP Refresh Reduction is enabled.
              'false' if RSVP Refresh Reduction is disabled.";
         }
         reference
           "RFC2961 RSVP Refresh Overhead Reduction Extensions";
       }
       container hellos {
         description
           "Top level container for RSVP hello parameters.";
         leaf enabled {
           type boolean;
           default "true";
           description
             "'true' if RSVP Hello is enabled.
              'false' if RSVP Hello is disabled.";
         reference
           "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
            RFC5495: Description of the Resource Reservation Protocol -
            Traffic-Engineered (RSVP-TE) Graceful Restart Procedures.";
         }
       }
       container authentication {
         description
           "Top level container for RSVP authentication parameters.";
         leaf enabled {
           type boolean;
           default "false";
           description
             "'true' if RSVP Authentication is enabled.
              'false' if RSVP Authentication is disabled.";
         }
         leaf authentication-key {
           type string;
           description
             "An authentication key string.";
           reference
             "RFC2747: RSVP Cryptographic Authentication";
         }
         leaf crypto-algorithm {
           type identityref {
             base key-chain:crypto-algorithm;
           }
           description
             "Cryptographic algorithm associated with key.";
         }
       }
       container statistics {
         config false;
         description
           "RSVP statistics container.";
         container messages {
           description
             "RSVP protocol statistics container.";
           leaf ack-sent {
             type yang:counter64;
             description
               "RSVP Hello sent count.";
           }
           leaf ack-received {
             type yang:counter64;
             description
               "RSVP Hello received count.";
           }
           leaf bundle-sent {
             type yang:counter64;
             description
               "RSVP Bundle message sent count.";
           }
           leaf bundle-received {
             type yang:counter64;
             description
               "RSVP Bundle message received count.";
           }
           leaf hello-sent {
             type yang:counter64;
             description
               "RSVP Hello message sent count.";
           }
           leaf hello-received {
             type yang:counter64;
             description
               "RSVP Hello message received count.";
           }
           leaf integrity-challenge-sent {
             type yang:counter64;
             description
               "RSVP Integrity Challenge message sent count.";
           }
           leaf integrity-challenge-received {
             type yang:counter64;
             description
               "RSVP Integrity Challenge message received count.";
           }
           leaf integrity-response-sent {
             type yang:counter64;
             description
               "RSVP Integrity Response message sent count.";
           }
           leaf integrity-response-received {
             type yang:counter64;
             description
```

```
              "RSVP Integrity Response message received count.";
          }
          leaf notify-sent {
            type yang:counter64;
            description
              "RSVP Notify message sent count.";
          }
          leaf notify-received {
            type yang:counter64;
            description
              "RSVP Notify message received count.";
          }
          leaf path-sent {
            type yang:counter64;
            description
              "RSVP Path message sent count.";
          }
          leaf path-received {
            type yang:counter64;
            description
              "RSVP Path message received count.";
          }
          leaf path-err-sent {
            type yang:counter64;
            description
              "RSVP Path error message sent count.";
          }
          leaf path-err-received {
            type yang:counter64;
            description
              "RSVP Path error message received count.";
          }
          leaf path-tear-sent {
            type yang:counter64;
            description
              "RSVP Path tear message sent count.";
          }
          leaf path-tear-received {
            type yang:counter64;
            description
              "RSVP Path tear message received count.";
          }
          leaf resv-sent {
            type yang:counter64;
            description
              "RSVP Resv message sent count.";
          }
          leaf resv-received {
            type yang:counter64;
            description
              "RSVP Resv message received count.";
          }
          leaf resv-confirm-sent {
            type yang:counter64;
            description
              "RSVP Confirm message sent count.";
          }
          leaf resv-confirm-received {
            type yang:counter64;
            description
              "RSVP Confirm message received count.";
          }
          leaf resv-err-sent {
            type yang:counter64;
            description
              "RSVP Resv error message sent count.";
          }
          leaf resv-err-received {
            type yang:counter64;
            description
              "RSVP Resv error message received count.";
          }
          leaf resv-tear-sent {
            type yang:counter64;
            description
              "RSVP Resv tear message sent count.";
          }
          leaf resv-tear-received {
            type yang:counter64;
            description
              "RSVP Resv tear message received count.";
          }
          leaf srefresh-sent {
            type yang:counter64;
            description
              "RSVP Srefresh message sent count.";
          }
          leaf srefresh-received {
            type yang:counter64;
            description
              "RSVP Srefresh message received count.";
          }
          leaf unknown-messages-received {
            type yang:counter64;
            description
              "Unknown messages received count.";
          }
        }
        container packets {
          description
            "Packet statistics container.";
          leaf sent {
            type yang:counter64;
            description
              "RSVP packet sent count.";
          }
          leaf received {
            type yang:counter64;
            description
              "RSVP packet received count.";
          }
        }
        container errors {
          description
            "Error statistics container.";
          leaf authenticate {
            type yang:counter64;
            description
              "The total number of RSVP packets received with an
               authentication failure.";
          }
          leaf checksum {
            type yang:counter64;
            description
              "The total number of RSVP packets received with an
               invalid checksum value.";
          }
          leaf packet-length {
            type yang:counter64;
            description
              "The total number of packets received with an invalid
               packet length.";
          }
        }
      }
    }

    augment "/rt:routing/rt:control-plane-protocols/"
          + "rt:control-plane-protocol" {
      when "rt:type = 'rsvp:rsvp'" {
        description
          "This augment is only valid when routing protocol instance
           type is RSVP.";
      }
      description
        "RSVP protocol augmentation.";
      container rsvp {
        presence "Enable RSVP feature";
        description
          "RSVP feature container";
        container interfaces {
          description
            "RSVP interfaces container.";
          uses intf-attributes;
          list interface {
            key "name";
            description
              "RSVP interfaces.";
            leaf name {
              type if:interface-ref;
              description
                "RSVP interface.";
            }
            uses intf-attributes;
          }
        }
        container sessions {
          description
            "RSVP sessions container.";
          list session-ip {
            key "destination protocol-id destination-port";
            config false;
            description
              "List of RSVP sessions.";
            leaf destination-port {
              type uint16;
              description
                "RSVP destination port.";
              reference
                "RFC2205";
            }
            leaf protocol-id {
              type uint8;
              description
                "The IP protocol ID.";
              reference
                "RFC2205, section 3.2";
            }
            leaf source {
              type inet:ip-address;
              description
                "RSVP source address.";
              reference
                "RFC2205";
            }
            leaf destination {
              type inet:ip-address;
              description
                "RSVP destination address.";
              reference
                "RFC2205";
            }
            leaf session-name {
              type string;
              description
                "The signaled name of this RSVP session.";
            }
            leaf session-status {
              type enumeration {
                enum up {
                  description
                    "RSVP session is up.";
                }
                enum down {
                  description
                    "RSVP session is down.";
                }
              }
              description
                "Enumeration of RSVP session states.";
            }
            leaf session-type {
              type identityref {
                base rsvp-session-type;
              }
              mandatory "true";
              description
                "RSVP session type.";
            }
            container psbs {
              description
                "Path State Block (PSB) container.";
              list psb {
                description
                  "List of Path State Blocks.";
                leaf source-port {
                  type inet:port-number;
                  description
                    "RSVP source port.";
                  reference
```

```
                     "RFC2205";
                 }
                 leaf expires-in {
                   type uint32;
                   units "seconds";
                   description
                     "Time to expiry (in seconds).";
                 }
               }
             }
             container rsbs {
               description
                 "Reservation State Block (RSB) container.";
               list rsb {
                 description
                   "List of Reservation State Blocks.";
                 leaf source-port {
                   type inet:port-number;
                   description
                     "RSVP source port.";
                   reference
                     "RFC2205";
                 }
                 leaf reservation-style {
                   type identityref {
                     base reservation-style;
                   }
                   description
                     "RSVP reservation style.";
                 }
                 leaf expires-in {
                   type uint32;
                   units "seconds";
                   description
                     "Time to expiry (in seconds).";
                 }
               }
             }
           }
         }
         container neighbors {
           description
             "RSVP neighbors container";
           list neighbor {
             key "address";
             description
               "List of RSVP neighbors";
             leaf address {
               type inet:ip-address;
               description
                 "Address of the RSVP neighbor.";
             }
             leaf epoch {
               type uint32;
               description
                 "Neighbor epoch.";
               reference "RFC2961 and RFC5063";
             }
             leaf expiry-time {
               type uint32;
               units "seconds";
               description
                 "Neighbor expiry time after which the neighbor state is
                  purged if no states associated with it.";
             }
             container graceful-restart {
               description
                 "Graceful restart information.";
               leaf neighbor-restart-time {
                 type uint32;
                 units "seconds";
                 config false;
                 description
                   "Time it takes the neighbor node to restart its
                    RSVP-TE component (to the point where it can
                    exchange RSVP Hello with its neighbors). A value of
                    0xffffffff indicates that the restart of the
                    neighbor's control plane may occur over an
                    indeterminate interval and that the operation of its
                    data plane is unaffected by control plane
                    failures.";
                 reference "RFC3473";
               }
               leaf neighbor-recovery-time {
                 type uint32;
                 units "seconds";
                 config false;
                 description
                   "The period of time, in milliseconds, that the
                    neighbor node requires to re-synchronize RSVP and
                    MPLS forwarding state with its neighbor. A value of
                    zero (0) indicates that MPLS forwarding state was
                    not preserved across a particular reboot.";
                 reference "RFC3473";
               }
               container helper-mode {
                 config false;
                 description
                   "Helper mode information.";
                 leaf neighbor-restart-time-remaining {
                   type uint32;
                   units "seconds";
                   description
                     "Number of seconds remaining for neighbor to send
                      Hello message after restart.";
                   reference "RFC5063";
                 }
                 leaf neighbor-recovery-time-remaining {
                   type uint32;
                   units "seconds";
                   description
                     "Number of seconds remaining for neighbor to
                      refresh.";
                   reference "RFC5063";
                 }
               }
               // helper-mode
             }
             leaf hello-status {
               type enumeration {
                 enum enabled {
                   description
                     "RSVP Hellos enabled.";
                 }
                 enum disabled {
                   description
                     "RSVP Hellos disabled.";
                 }
                 enum restarting {
                   description
                     "RSVP restarting.";
                 }
               }
               config false;
               description
                 "RSVP Hello status.";
             }
             leaf interface {
               type if:interface-ref;
               description
                 "Interface where RSVP neighbor was detected.";
             }
             leaf neighbor-status {
               type enumeration {
                 enum up {
                   description
                     "Neighbor state up.";
                 }
                 enum down {
                   description
                     "Neighbor state down.";
                 }
                 enum hello-disable {
                   description
                     "RSVP Hellos disabled.";
                 }
                 enum restarting {
                   description
                     "RSVP neighbor restarting.";
                 }
               }
               config false;
               description
                 "RSVP neighbor state.";
             }
             leaf refresh-reduction-capable {
               type boolean;
               default "true";
               description
                 "Enables all RSVP refresh reduction message bundling,
                  RSVP message ID, reliable message delivery and
                  Srefresh messages.";
               reference
                 "RFC2961 RSVP Refresh Overhead Reduction Extensions";
             }
             leaf restart-count {
               type yang:counter32;
               config false;
               description
                 "Number of times this RSVP neighbor has restarted.";
             }
             leaf restart-time {
               type yang:date-and-time;
               config false;
               description
                 "Last restart time of the RSVP neighbor.";
               reference "RFC3473";
             }
           }
         }
         container graceful-restart {
           description
```

```
             "Graceful restart local information.";
           leaf enabled {
             type boolean;
             description
               "'true' if RSVP Graceful Restart is enabled.
                'false' if RSVP Graceful Restart is disabled.";
             reference "RFC5495";
           }
           leaf local-restart-time {
             type uint32;
             units "seconds";
             description
               "Time it takes the local node to restart its RSVP-TE
                component (to the point where it can exchange RSVP
                Hello with its neighbors). A value of 0xffffffff
                indicates that the restart of the neighbor's control
                plane may occur over an indeterminate interval and that
                the operation of its data plane is unaffected by control
                plane failures.";
             reference "RFC3473";
           }
           leaf local-recovery-time {
             type uint32;
             units "seconds";
             description
               "The period of time, in seconds, that the local
                node requires to re-synchronize RSVP and MPLS
                forwarding state with its neighbor. A value of zero (0)
                indicates that MPLS forwarding state was not preserved
                across a particular reboot.";
             reference "RFC3473";
           }
           container helper-mode {
             description
               "Helper mode information. In this mode, the node
                resynchronizes its stored states with a neighbor whose
                control plane has restarted. The helper mode term is
                borrowed from RFC3623 and adopted by several vendors
                vendors in their implementation of RSVP graceful
                restart.";
             leaf enabled {
               type boolean;
               description
                 "'true' if helper mode is enabled.";
             }
             leaf max-helper-restart-time {
               type uint32;
               units "seconds";
               default "20";
               description
                 "The maximum time the router or switch waits after it
                  discovers that the neighboring router has gone down
                  before it declares the neighbor down.";
               reference "RFC5063";
             }
             leaf max-helper-recovery-time {
               type uint32;
               units "seconds";
               default "180";
               description
                 "The maximum amount of time the router retains the
                  state of its RSVP neighbors while they undergo a
                  graceful restart.";
               reference "RFC5063";
             }
           }
         }
       }
     }

     grouping session-ref {
       description
         "Session reference information";
       leaf destination {
         type leafref {
           path "/rt:routing/rt:control-plane-protocols"
              + "/rt:control-plane-protocol/rsvp:rsvp"
              + "/rsvp:sessions/rsvp:session-ip/destination";
         }
         mandatory true;
         description
           "The RSVP session destination.";
       }
       leaf protocol-id {
         type uint8;
         mandatory true;
         description
           "The RSVP session protocol ID.";
       }
       leaf destination-port {
         type inet:ip-address;
         mandatory true;
         description
           "The RSVP session destination port.";
       }
     }
     rpc clear-session {
       nacm:default-deny-all;
       description
         "Clears RSVP sessions RPC";
       input {
         leaf routing-protocol-instance-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory true;
           description
             "Name of the RSVP protocol instance whose session
              is being cleared.

              If the corresponding RSVP instance doesn’t exist,
              then the operation will fail with an error-tag of
              ’data-missing’ and an error-app-tag of
              ’routing-protocol-instance-not-found’.";
         }
         choice filter-type {
           mandatory true;
           description
             "Filter choice";
           case match-all {
             leaf all {
               type empty;
               mandatory true;
               description
                 "Match all RSVP sessions.";
             }
           }
           case match-one {
             container session-info {
               description
                 "Specifies the specific session to invoke the operation
                  on.";
               choice session-type {
                 mandatory true;
                 description
                   "The RSVP session type.";
                 case rsvp-session-ip {
                   uses session-ref;
                 }
               }
             }
           }
         }
       }
     }

     rpc clear-neighbor {
       nacm:default-deny-all;
       description
         "RPC to clear the RSVP Hello session to a neighbor.";
       input {
         leaf routing-protocol-instance-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory true;
           description
             "Name of the RSVP protocol instance whose session
              is being cleared.

              If the corresponding RSVP instance doesn't exist,
              then the operation will fail with an error-tag of
              'data-missing' and an error-app-tag of
              'routing-protocol-instance-not-found'.";
         }
         choice filter-type {
           mandatory true;
           description
             "The Filter choice.";
           case match-all {
             leaf all {
               type empty;
               mandatory true;
               description
                 "Match all RSVP neighbor sessions.";
             }
           }
           case match-one {
             leaf neighbor-address {
               type leafref {
                 path "/rt:routing/rt:control-plane-protocols"
                    + "/rt:control-plane-protocol/rsvp:rsvp"
                    + "/rsvp:neighbors/rsvp:neighbor/address";
               }
               mandatory true;
               description
                 "Match the specific RSVP neighbor session.";
             }
           }
         }
       }
     }

     rpc clear-authentication {
       nacm:default-deny-all;
       description
         "Clears the RSVP Security Association (SA) before the
          lifetime expires.";
       input {
         leaf routing-protocol-instance-name {
           type leafref {
             path "/rt:routing/rt:control-plane-protocols/"
                + "rt:control-plane-protocol/rt:name";
           }
           mandatory true;
           description
             "Name of the RSVP protocol instance whose session
              is being cleared.

              If the corresponding RSVP instance doesn’t exist,
              then the operation will fail with an error-tag of
              ’data-missing’ and an error-app-tag of
              ’routing-protocol-instance-not-found’.";
         }
         choice filter-type {
           mandatory true;
           description
             "Filter choice";
           case match-all {
             leaf all {
               type empty;
               mandatory true;
               description
                 "Match all RSVP security associations.";
             }
           }
           case match-one-interface {
             leaf interface {
               type if:interface-ref;
               description
                 "Interface where RSVP security association(s) to be
                  detected.";
             }
           }
         }
       }
     }
   }
   <CODE ENDS>
```

Figure 4: RSVP YANG module

5. RSVP Extended YANG Model

The RSVP extended module augments the RSVP base module with optional feature data as described in Section 3.3.

5.1. Tree Diagram

Figure 5 shows the YANG tree representation for configuration and state data that are covered in 'ietf-rsvp-extended' YANG module:

```
   module: ietf-rsvp-extended

     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
       +--rw refresh-interval?        uint32
       +--rw refresh-misses?          uint32
       +--rw checksum-enable?         empty
       +--rw patherr-state-removal?   empty
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:statistics/rsvp:packets:
       +--ro discontinuity-time?   yang:date-and-time
       +--ro out-dropped?          yang:counter64
       +--ro in-dropped?           yang:counter64
       +--ro out-errors?           yang:counter64
       +--ro in-errors?            yang:counter64
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:refresh-reduction:
       +--rw bundle-message-max-size?    uint32
       +--rw ack-hold-time?              uint32
       +--rw ack-max-size?               uint32
       +--rw ack-retransmit-time?        uint32
       +--rw srefresh-ack-desired?       empty
       +--rw srefresh-max-size?          uint32
       +--rw srefresh-relative-period?   uint8
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:hellos:
       +--rw interface-based?   empty
       +--rw hello-interval?    uint32
       +--rw hello-misses?      uint32
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:authentication:
       +--rw lifetime?      uint32
       +--rw window-size?   uint32
       +--rw challenge?     empty
       +--rw retransmits?   uint32
       +--rw key-chain?     key-chain:key-chain-ref
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface:
       +--rw refresh-interval?        uint32
       +--rw refresh-misses?          uint32
       +--rw checksum-enable?         empty
       +--rw patherr-state-removal?   empty
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface/rsvp:statistics/rsvp:packets:
       +--ro discontinuity-time?   yang:date-and-time
       +--ro out-dropped?          yang:counter64
       +--ro in-dropped?           yang:counter64
       +--ro out-errors?           yang:counter64
       +--ro in-errors?            yang:counter64
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface/rsvp:refresh-reduction:
       +--rw bundle-message-max-size?    uint32
       +--rw ack-hold-time?              uint32
       +--rw ack-max-size?               uint32
       +--rw ack-retransmit-time?        uint32
       +--rw srefresh-ack-desired?       empty
       +--rw srefresh-max-size?          uint32
       +--rw srefresh-relative-period?   uint8
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface/rsvp:hellos:
       +--rw interface-based?   empty
       +--rw hello-interval?    uint32
       +--rw hello-misses?      uint32
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface/rsvp:authentication:
       +--rw lifetime?      uint32
       +--rw window-size?   uint32
       +--rw challenge?     empty
       +--rw retransmits?   uint32
       +--rw key-chain?     key-chain:key-chain-ref
```

Figure 5: RSVP extended module tree diagram

5.2. YANG Module

The 'ietf-rsvp-extended' module imports from the following modules:

- ietf-rsvp defined in this document

- ietf-routing defined in [RFC8349]

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

- ietf-key-chain defined in [RFC8177]

Figure 6 shows the RSVP extended YANG module:

This module also references the following documents: [RFC3473], [RFC2747], [RFC3209], [RFC2205], [RFC2961], and [RFC5495].

```
<CODE BEGINS> file "ietf-rsvp-extended@2024-02-28.yang"
module ietf-rsvp-extended {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";
  prefix rsvp-extended;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
       (RSVP)";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management
       (NMDA Version)";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC6991: Common YANG Data Types";
  }
  import ietf-key-chain {
    prefix key-chain;
    reference
      "RFC8177: YANG Data Model for Key Chains";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
```

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

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

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.com>

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

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

        Editor:   Igor Bryskin
                  <mailto:i_bryskin@yahoo.com>";
     description
       "This module contains the Extended RSVP YANG data model.
        The model fully conforms to the Network Management Datastore
        Architecture (NMDA).

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

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

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

     revision 2024-02-28 {
       description
         "Initial version.";
       reference
         "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
          (RSVP)";
     }
     grouping authentication-extended {
       description
         "Configuration parameters relating to RSVP authentication.";
       leaf lifetime {
         type uint32 {
           range "30..86400";
         }
         units "seconds";
         description
           "Life time for each security association.";
         reference
           "RFC2747: RSVP Cryptographic Authentication";
       }
       leaf window-size {
         type uint32 {
           range "1..64";
         }
         description
           "Window-size to limit number of out-of-order messages.";
         reference
           "RFC2747: RSVP Cryptographic Authentication";
       }
       leaf challenge {
         type empty;
         description
           "Enable challenge messages.";
         reference
           "RFC2747: RSVP Cryptographic Authentication";
       }
       leaf retransmits {
         type uint32 {
           range "1..10000";
         }
         description
           "Number of retransmits when messages are dropped.";
         reference
           "RFC2747: RSVP Cryptographic Authentication";
       }
       leaf key-chain {
         type key-chain:key-chain-ref;
         description
           "Key chain name to authenticate RSVP
            signaling messages.";
         reference
           "RFC2747: RSVP Cryptographic Authentication";
       }
     }
     grouping hellos-extended {
       description
         "Configuration parameters relating to RSVP hellos";
       leaf interface-based {
         type empty;
         description
           "Enable interface-based Hello adjacency if present.";
       }
       leaf hello-interval {
         type uint32;
         units "milliseconds";
         description
           "Configure interval between successive Hello messages in
            milliseconds.";
         reference
           "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
            RFC5495: Description of the Resource Reservation Protocol -
            Traffic-Engineered (RSVP-TE) Graceful Restart Procedures.";
       }
       leaf hello-misses {
         type uint32 {
           range "1..10";
         }
         description
           "Configure max number of consecutive missed Hello messages.";
         reference
           "RFC3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
            RFC5495: Description of the Resource Reservation Protocol -
            Traffic- Engineered (RSVP-TE) Graceful Restart Procedures.";
       }
     }

     grouping signaling-parameters-extended {
       description
         "Configuration parameters relating to RSVP signaling";
       leaf refresh-interval {
         type uint32;
         units "seconds";
         description
           "Set interval between successive refreshes";
         reference "RFC2205";
       }
       leaf refresh-misses {
         type uint32;
         description
           "Set max number of consecutive missed messages for state
            expiry";
         reference "RFC2205";
       }
       leaf checksum-enable {
         type empty;
         description
           "Enable RSVP message checksum computation";
         reference "RFC2205";
       }
       leaf patherr-state-removal {
         type empty;
         description
           "State-Removal flag in Path Error message if present.";
         reference "RFC3473";
       }
     }

     grouping refresh-reduction-extended {
       description
         "Configuration parameters relating to RSVP refresh reduction.";
       leaf bundle-message-max-size {
         type uint32 {
           range "512..65000";
         }
         description
           "Configure maximum size (bytes) of a single RSVP Bundle
            message.";
         reference "RFC2961";
       }
       leaf ack-hold-time {
         type uint32;
         units "milliseconds";
         description
           "Configure hold time in milliseconds for sending RSVP ACK
            message(s).";
         reference "RFC2961";
       }
       leaf ack-max-size {
         type uint32;
         description
           "Configure max size of a single RSVP ACK message.";
         reference "RFC2961";
       }
       leaf ack-retransmit-time {
         type uint32;
         units "milliseconds";
         description
           "Configure min delay in milliseconds to wait for an
            acknowledgment before being retransmitted.";
         reference "RFC2961";
       }
       leaf srefresh-ack-desired {
         type empty;
         description
           "Enables the sending of MESSAGE_ID with ACK_Desired
            set with Srefresh messages.";
         reference "RFC2961";
       }
       leaf srefresh-max-size {
         type uint32 {
           range "20..65000";
         }
         description
           "Configure max size (bytes) of a single RSVP Srefresh
            message.";
         reference "RFC2961";
       }
       leaf srefresh-relative-period {
         type uint8 {
           range "10..100";
         }
         description
           "Configures the period of Srefreshes relative to standard
            refresh message period in percentage.";
       }
     }

     grouping packets-extended-statistics {
       description
         "Packet statistics.";
       leaf discontinuity-time {
         type yang:date-and-time;
         description
           "The time on the most recent occasion at which any one or
            more of the statistic counters suffered a discontinuity.
            If no such discontinuities have occurred since the last
            re-initialization of the local management subsystem, then
            this node contains the time the local management subsystem
            re-initialized itself.";
       }
       leaf out-dropped {
         type yang:counter64;
         description
           "Out RSVP packet drop count.";
       }
       leaf in-dropped {
         type yang:counter64;
         description
```

```
           "In RSVP packet drop count.";
       }
       leaf out-errors {
         type yang:counter64;
         description
           "Out RSVP packet errors count.";
       }
       leaf in-errors {
         type yang:counter64;
         description
           "In RSVP packet rx errors count.";
       }
     }

     /**
      * RSVP extensions augmentations
      */

     /**
      * RSVP all interfaces extensions
      */

     /* RSVP interface signaling extensions */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
       description
         "RSVP signaling all interfaces configuration extensions";
       uses signaling-parameters-extended;
     }

     /* Packet statistics extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:statistics/rsvp:packets" {
       description
         "RSVP packets all interfaces configuration extensions";
       uses packets-extended-statistics;
     }

     /* RSVP refresh reduction extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:refresh-reduction" {
       description
         "RSVP refresh-reduction all interface configuration
          extensions";
       uses refresh-reduction-extended;
     }
     /* RSVP hellos extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:hellos" {
       description
         "RSVP hello all interfaces configuration extensions";
       uses hellos-extended;
     }

     /* RSVP authentication extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:authentication" {
       description
         "RSVP authentication all interfaces configuration extensions";
       uses authentication-extended;
     }

     /**
      * RSVP per interface extensions
      */
     /* RSVP interface signaling extensions */

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:interface" {
       description
         "RSVP signaling interface configuration extensions";
       uses signaling-parameters-extended;
     }

     /* Packet statistics extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:interface/rsvp:statistics/rsvp:packets" {
       description
         "RSVP packet stats extensions";
       uses packets-extended-statistics;
     }

     /* RSVP refresh reduction extension */
     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:interface/rsvp:refresh-reduction" {
       description
         "RSVP refresh-reduction interface configuration extensions";
       uses refresh-reduction-extended;
     }
```

```
  /* RSVP hellos extension */
  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
        + "rsvp:interface/rsvp:hellos" {
    description
      "RSVP hello interface configuration extensions";
    uses hellos-extended;
  }

  /* RSVP authentication extension */
  augment "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
        + "rsvp:interface/rsvp:authentication" {
    description
      "RSVP authentication interface configuration extensions";
    uses authentication-extended;
  }
}
<CODE ENDS>
```

Figure 6: RSVP extended YANG module

6. 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-rsvp
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

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

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

```
name:       ietf-rsvp
namespace:  urn:ietf:params:xml:ns:yang:ietf-rsvp
prefix:     rsvp
reference:  RFCXXXX

name:       ietf-rsvp-extended
namespace:  urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
prefix:     rsvp-extended
reference:  RFCXXXX
```

7. Security Considerations

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

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

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

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions

> All of which are considered sensitive and if access to either of these is compromised, it can result in temporary network outages or be employed to mount DoS attacks.

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

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/ /rsvp:globals /rsvp:interfaces /rsvp:sessions

> Additional information from these state data nodes can be inferred with respect to the network topology, and device location and subsequently be used to mount other attacks in the network.

For RSVP authentication, the configuration supported is via the specification of key-chains [RFC8177] or the direct specification of key and authentication algorithm, and hence security considerations of [RFC8177] are inherited. This includes the considerations with respect to the local storage and handling of authentication keys.

Some of the RPC operations defined in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. The RSVP YANG module support the "clear-session" and "clear-neighbor" RPCs. If access to either of these is compromised, they can result in temporary network outages be employed to mount DoS attacks.

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

8. Acknowledgement

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Lou Berger, Xia Chen, Bin Wen, and Raqib Jones for reviewing and providing valuable feedback on this document.

9. Appendix A

A simple network setup is shown in {fig-example title}. R1 runs the RSVP routing protocol on both interfaces 'ge0/0/0/1', and 'ge0/0/0/2'.

```
State on R1:

Sessions:
=========
Destination         Protocol-ID Dest-port
198.51.100.1        10          10

Neighbors:
==========
Neighbor Address    Interface
192.0.2.6           ge0/0/0/1
```

```
                               192.0.2.5/30
                              ge0/0/0/1
                              +---
                             /
                        +-----+
                        | R1  |
                        +-----+
                             \
                              +---
                              ge0/0/0/2
                               192.0.2.13/30
```

Figure 7: Example of network configuration.

The instance data tree could then be as follows:

```
{
  "ietf-routing:routing": {
    "control-plane-protocols": {
      "control-plane-protocol": [
        {
          "type": "rt:routing-protocol",
          "name": "rsvp:rsvp",
          "ietf-rsvp:rsvp": {
            "interfaces": {
              "refresh-reduction": {
                "enabled": true,
                "ietf-rsvp-extended:bundle-message-max-size": 2000,
                "ietf-rsvp-extended:reliable-ack-hold-time": 180,
                "ietf-rsvp-extended:reliable-ack-max-size": 2000,
                "ietf-rsvp-extended:reliable-retransmit-time": 180,
                "ietf-rsvp-extended:reliable-srefresh": [
                  null
                ],
                "ietf-rsvp-extended:summary-max-size": 2000
              },
              "hellos": {
                "enabled": true,
                "ietf-rsvp-extended:interface-based": [
                  null
                ],
                "ietf-rsvp-extended:hello-interval": 27000,
                "ietf-rsvp-extended:hello-misses": 3
              },
              "statistics": {
                "messages": {
                  "ack-sent": "777",
                  "ack-received": "4840",
                  "bundle-sent": "2195",
                  "bundle-received": "293",
                  "hello-sent": "2516",
                  "hello-received": "3535",
                  "integrity-challenge-sent": "2737",
                  "integrity-challenge-received": "2330",
                  "integrity-response-sent": "895",
                  "integrity-response-received": "1029",
                  "path-sent": "1197",
                  "path-received": "3568",
                  "path-err-sent": "4658",
                  "path-err-received": "695",
                  "path-tear-sent": "3706",
                  "path-tear-received": "2604",
                  "resv-sent": "3353",
                  "resv-received": "3129",
                  "resv-err-sent": "1787",
                  "resv-err-received": "3205",
                  "resv-tear-sent": "4465",
                  "resv-tear-received": "3056",
                  "summary-refresh-sent": "655",
                  "summary-refresh-received": "3856"
                },
                "packets": {
                  "sent": "2147",
                  "received": "4374",
                  "ietf-rsvp-extended:discontinuity-time":
                  "2015-10-24T17:11:27+02:00",
                  "ietf-rsvp-extended:out-dropped": "2696",
                  "ietf-rsvp-extended:in-dropped": "941",
                  "ietf-rsvp-extended:out-errors": "19",
                  "ietf-rsvp-extended:in-errors": "2732"
                },
                "errors": {
                  "authenticate": "2540",
                  "checksum": "2566",
                  "packet-length": "267"
                }
              },
              "interface": [
                {
                  "interface": "ge0/0/0/1",
                  "statistics": {
                    "messages": {
                      "ack-sent": "2747",
                      "ack-received": "4934",
                      "bundle-sent": "1618",
                      "bundle-received": "3668",
                      "hello-sent": "4288",
                      "hello-received": "1194",
                      "integrity-challenge-sent": "4850",
                      "integrity-challenge-received": "3979",
                      "integrity-response-sent": "479",
                      "integrity-response-received": "1773",
                      "path-sent": "2230",
                      "path-received": "1793",
                      "path-err-sent": "465",
                      "path-err-received": "1859",
                      "path-tear-sent": "923",
                      "path-tear-received": "3924",
                      "resv-sent": "3203",
                      "resv-received": "2507",
                      "resv-err-sent": "1259",
                      "resv-err-received": "2445",
                      "resv-tear-sent": "3045",
                      "resv-tear-received": "4676",
                      "summary-refresh-sent": "365",
                      "summary-refresh-received": "2129"
                    },
                    "packets": {
                      "sent": "847",
                      "received": "3114",
                      "ietf-rsvp-extended:discontinuity-time":
                      "2015-10-24T17:11:27+02:00",
                      "ietf-rsvp-extended:out-dropped": "1841",
                      "ietf-rsvp-extended:in-dropped": "4832",
                      "ietf-rsvp-extended:out-errors": "1334",
                      "ietf-rsvp-extended:in-errors": "3900"
                    },
                    "errors": {
                      "authenticate": "3494",
                      "checksum": "4374",
                      "packet-length": "2456"
                    }
                  }
                },
                {
                  "interface": "ge0/0/0/2",
                  "statistics": {
                    "messages": {
                      "ack-sent": "1276",
                      "ack-received": "2427",
                      "bundle-sent": "4053",
                      "bundle-received": "3509",
                      "hello-sent": "3261",
                      "hello-received": "2863",
                      "integrity-challenge-sent": "4744",
                      "integrity-challenge-received": "3554",
                      "integrity-response-sent": "3155",
                      "integrity-response-received": "169",
                      "path-sent": "3853",
                      "path-received": "409",
                      "path-err-sent": "4227",
                      "path-err-received": "2830",
                      "path-tear-sent": "1742",
                      "path-tear-received": "3344",
                      "resv-sent": "3154",
                      "resv-received": "3492",
                      "resv-err-sent": "3112",
                      "resv-err-received": "3974",
                      "resv-tear-sent": "3657",
                      "resv-tear-received": "533",
                      "summary-refresh-sent": "4036",
                      "summary-refresh-received": "2123"
                    },
                    "packets": {
                      "sent": "473",
                      "received": "314",
                      "ietf-rsvp-extended:discontinuity-time":
                      "2015-10-24T17:11:27+02:00",
                      "ietf-rsvp-extended:out-dropped": "2042",
                      "ietf-rsvp-extended:in-dropped": "90",
                      "ietf-rsvp-extended:out-errors": "1210",
                      "ietf-rsvp-extended:in-errors": "1361"
                    },
                    "errors": {
                      "authenticate": "543",
                      "checksum": "2241",
                      "packet-length": "480"
                    }
                  }
                }
```

```
                 ],
                 "ietf-rsvp-extended:refresh-interval": 30,
                 "ietf-rsvp-extended:refresh-misses": 5,
                 "ietf-rsvp-extended:checksum_enabled": true,
                 "ietf-rsvp-extended:patherr-state-removal": [
                   null
                 ]
               },
               "sessions": {
                 "session-ip": [
                   {
                     "destination-port": 10,
                     "protocol-id": 10,
                     "destination": "198.51.100.1",
                     "psbs": {
                       "psb": [
                         {
                           "source-port": 10,
                           "expires-in": 100
                         }
                       ]
                     },
                     "rsbs": {
                       "rsb": [
                         {
                           "source-port": 10,
                           "reservation-style":
                           "rsvp:reservation-wildcard-filter",
                           "expires-in": 100
                         }
                       ]
                     }
                   }
                 ]
               },
               "neighbors": {
                 "neighbor": [
                   {
                     "address": "192.0.2.6",
                     "epoch": 130,
                     "expiry-time": 260,
                     "graceful-restart": {
                       "enabled": true,
                       "local-restart-time": 271,
                       "local-recovery-time": 138,
                       "neighbor-restart-time": 341,
                       "neighbor-recovery-time": 342
                     },
                     "hello-status": "enabled",
                     "interface": "ge0/0/0/1",
                     "restart-count": 2,
                     "restart-time": "2015-10-24T17:11:27+02:00"
                   }
                 ]
               },
               "graceful-restart": {
                 "enabled": true,
                 "local-restart-time": 60,
                 "local-recovery-time": 180,
                 "neighbor-restart-time": 80,
                 "neighbor-recovery-time": 200,
                 "helper-mode": {
                   "enabled": true
                 }
               }
             }
           }
         ]
       }
     }
   }
```

Figure 8: Example RSVP JSON encoded data instance tree.

10. Contributors

Himanshu Shah
Ciena

Email: hshah@ciena.com

11. References

11.1. Normative References

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

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

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

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

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

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

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

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

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

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

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

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

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

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

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

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

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

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

11.2. Informative References

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, DOI 10.17487/RFC2747, January 2000, <https://www.rfc-editor.org/info/rfc2747>.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <https://www.rfc-editor.org/info/rfc2961>.

[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>.

[rfc4558] Ali, Z., Rahman, R., Prairie, D., and D. Papadimitriou, "Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement", RFC 4558, DOI 10.17487/RFC4558, June 2006, <https://www.rfc-editor.org/info/rfc4558>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <https://www.rfc-editor.org/info/rfc5063>.

[RFC5495] Li, D., Gao, J., Satyanarayana, A., and S. Bardalai, "Description of the Resource Reservation Protocol - Traffic-Engineered (RSVP-TE) Graceful Restart Procedures", RFC 5495, DOI 10.17487/RFC5495, March 2009, <https://www.rfc-editor.org/info/rfc5495>.

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc
Email: tsaad.net@gmail.com

Rakesh Gandhi
Cisco Systems, Inc.
Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com

Igor Bryskin
Individual
Email: i_bryskin@yahoo.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

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

February 22, 2021

A YANG Data Model for RSVP-TE Protocol
draft-ietf-teas-yang-rsvp-te-09

Abstract

This document defines a YANG data model for the configuration and management of RSVP (Resource Reservation Protocol) to establish Traffic-Engineered (TE) Label-Switched Paths (LSPs) for MPLS (Multi-Protocol Label Switching) and other technologies.

The model defines a generic RSVP-TE module for signaling LSPs that are technology agnostic. The generic RSVP-TE module is to be augmented by technology specific RSVP-TE modules that define technology specific data. This document also defines the augmentation for RSVP-TE MPLS LSPs model.

This model covers data for the 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 August 26, 2021.

Copyright Notice

Table of Contents

1. Introduction

YANG [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 defines a generic YANG data model for configuring and managing RSVP-TE LSP(s) [RFC3209]. The RSVP-TE generic model augments the RSVP base and extended models defined in [I-D.ietf-teas-yang-rsvp], and adds TE extensions to the RSVP protocol [RFC2205] model configuration and state data. The technology specific RSVP-TE models augment the generic RSVP-TE model with additional technology specific parameters. For example, this document also defines the MPLS RSVP-TE model for configuring and managing MPLS RSVP TE LSP(s).

In addition to augmenting the RSVP YANG module, the modules defined in this document augment the TE Interfaces, Tunnels and LSP(s) YANG module defined in [I-D.ietf-teas-yang-te] to define additional parameters to enable signaling for RSVP-TE.

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]
te	ietf-te	[I-D.ietf-teas-yang-te]
rsvp	ietf-rsvp	[I-D.ietf-teas-yang-rsvp]
te-dev	ietf-te-device	[I-D.ietf-teas-yang-te]
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]
rsvp-te	ietf-rsvp-te	this document
rsvp-te-mpls	ietf-rsvp-te-mpls	this document

Table 1: Prefixes and corresponding YANG modules

2. Model Overview

The RSVP-TE generic model augments the RSVP base and extended YANG models defined in [I-D.ietf-teas-yang-rsvp]. It also augments the TE tunnels and interfaces module defined in [I-D.ietf-teas-yang-te] to cover parameters specific to the configuration and management of RSVP-TE interfaces, tunnels and LSP(s).

The RSVP-TE MPLS YANG model augments the RSVP-TE generic model with parameters to configure and manage signaling of MPLS RSVP-TE LSPs. RSVP-TE model augmentation for other dataplane technologies (e.g. OTN or WDM) are outside the scope of this document.

There are three types of configuration and state data nodes in module(s) defined in this document:

- those augmenting or extending the base RSVP module that is defined in [I-D.ietf-teas-yang-rsvp]

- those augmenting or extending the base TE module defined in [I-D.ietf-teas-yang-te]

- those that are specific to the RSVP-TE and RSVP-TE MPLS modules defined in this document.

2.1. Module Relationship

The data pertaining to RSVP-TE in this document is divided into two modules: a technology agnostic RSVP-TE module that holds generic parameters for RSVP-TE applicable to all technologies, and a MPLS technology specific RSVP-TE module that holds parameters specific to MPLS technology.

The relationship between the different modules is shown in Figure 1.

```
     TE basic       +---------+
     module         | ietf-te |        o: augment
                    +---------+
                         o
                         |
                         |
                    +--------------+
     RSVP-TE module | ietf-rsvp-te |o . . .
                    +--------------+          \
                         |                     \
                         o                 +---------------------+
                    +-----------+          | ietf-rsvp-te-mpls   |
     RSVP module    | ietf-rsvp |          +---------------------+
                    +-----------+             RSVP-TE with MPLS
                         o
                         |
     RSVP extended       |
       module       +--------------------+
                    | ietf-rsvp-extended |
                    +--------------------+
```

Figure 1: Relationship of RSVP and RSVP-TE modules with other protocol modules

2.2. Model Tree Diagrams

A full tree diagram of the module(s) defined in this document as per the syntax defined in [RFC8340] are given in subsequent sections.

2.2.1. RSVP-TE Model Tree Diagram

Figure 2 shows the YANG tree diagram of the RSVP-TE generic YANG model defined in module ietf-rsvp-te.yang.

```
   module: ietf-rsvp-te
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp:
       +--rw global-soft-preemption!
          +--rw soft-preemption-timeout?   uint16
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
       +--rw rsvp-te-interface-attributes
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface:
       +--rw rsvp-te-interface-attributes
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:sessions:
       +--ro session-te* [tunnel-endpoint tunnel-id extended-tunnel-id]
          +--ro tunnel-endpoint       inet:ip-address
          +--ro tunnel-id             uint16
          +--ro extended-tunnel-id    inet:ip-address
          +--ro destination-port?     inet:port-number
          +--ro protocol-id?          uint8
          +--ro source?               inet:ip-address
          +--ro destination?          inet:ip-address
          +--ro session-name?         string
          +--ro session-status?       enumeration
          +--ro session-type          identityref
          +--ro psbs
          |  +--ro psb* []
          |     +--ro source-port?          inet:port-number
          |     +--ro expires-in?           uint32
          |     +--ro tspec-average-rate?
          |     |       rt-types:bandwidth-ieee-float32
          |     +--ro tspec-size?
          |     |       rt-types:bandwidth-ieee-float32
          |     +--ro tspec-peak-rate?
          |     |       rt-types:bandwidth-ieee-float32
          |     +--ro min-policed-unit?     uint32
          |     +--ro max-packet-size?      uint32
          +--ro rsbs
             +--ro rsb* []
                +--ro source-port?          inet:port-number
                +--ro reservation-style     identityref
                +--ro expires-in?           uint32
                +--ro fspec-average-rate?
                |       rt-types:bandwidth-ieee-float32
                +--ro fspec-size?
                |       rt-types:bandwidth-ieee-float32
                +--ro fspec-peak-rate?
                |       rt-types:bandwidth-ieee-float32
                +--ro min-policed-unit?     uint32
                +--ro max-packet-size?      uint32
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors:
     augment /te:te/te:tunnels/te:tunnel:
       +--rw lsp-signaled-name?   string
       +--rw session-attribute*   identityref
       +--rw lsp-attribute*       identityref
       +--rw retry-timer?         uint16
     augment /te:te/te:lsps/te:lsp:
       +--ro associated-rsvp-session?            leafref
       +--ro lsp-signaled-name?                  string
       +--ro session-attribute*                  identityref
       +--ro lsp-attribute*                      identityref
       +--ro rsvp-message-type?                  identityref
       +--ro rsvp-error-code?                    uint8
       +--ro rsvp-error-subcode?                 uint16
       +--ro explicit-route-objects
       |  +--ro incoming-explicit-route-hop* [index]
       |  |  +--ro index                        uint32
       |  |  +--ro (type)?
       |  |     +--:(numbered-node-hop)
       |  |     |  +--ro numbered-node-hop
       |  |     |     +--ro node-id     te-node-id
       |  |     |     +--ro hop-type?   te-hop-type
       |  |     +--:(numbered-link-hop)
       |  |     |  +--ro numbered-link-hop
       |  |     |     +--ro link-tp-id    te-tp-id
       |  |     |     +--ro hop-type?     te-hop-type
       |  |     |     +--ro direction?    te-link-direction
       |  |     +--:(unnumbered-link-hop)
       |  |     |  +--ro unnumbered-link-hop
       |  |     |     +--ro link-tp-id    te-tp-id
       |  |     |     +--ro node-id       te-node-id
       |  |     |     +--ro hop-type?     te-hop-type
       |  |     |     +--ro direction?    te-link-direction
       |  |     +--:(as-number)
       |  |     |  +--ro as-number-hop
       |  |     |     +--ro as-number    inet:as-number
       |  |     |     +--ro hop-type?    te-hop-type
       |  |     +--:(label)
       |  |        +--ro label-hop
       |  |           +--ro te-label
       |  |              +--ro (technology)?
       |  |              |  +--:(generic)
       |  |              |     +--ro generic?
       |  |              |             rt-types:generalized-label
       |  |              +--ro direction?       te-label-direction
       |  +--ro outgoing-explicit-route-hop* [index]
       |     +--ro index                        uint32
       |     +--ro (type)?
       |        +--:(numbered-node-hop)
       |        |  +--ro numbered-node-hop
       |        |     +--ro node-id     te-node-id
       |        |     +--ro hop-type?   te-hop-type
       |        +--:(numbered-link-hop)
       |        |  +--ro numbered-link-hop
       |        |     +--ro link-tp-id    te-tp-id
       |        |     +--ro hop-type?     te-hop-type
       |        |     +--ro direction?    te-link-direction
       |        +--:(unnumbered-link-hop)
       |        |  +--ro unnumbered-link-hop
       |        |     +--ro link-tp-id    te-tp-id
       |        |     +--ro node-id       te-node-id
       |        |     +--ro hop-type?     te-hop-type
       |        |     +--ro direction?    te-link-direction
       |        +--:(as-number)
       |        |  +--ro as-number-hop
       |        |     +--ro as-number    inet:as-number
       |        |     +--ro hop-type?    te-hop-type
       |        +--:(label)
       |           +--ro label-hop
       |              +--ro te-label
       |                 +--ro (technology)?
       |                 |  +--:(generic)
       |                 |     +--ro generic?
       |                 |             rt-types:generalized-label
       |                 +--ro direction?       te-label-direction
       +--ro incoming-record-route-subobjects
       |  +--ro incoming-record-route-subobject* [index]
       |     +--ro index                        uint32
       |     +--ro (type)?
       |        +--:(numbered-node-hop)
       |        |  +--ro numbered-node-hop
       |        |     +--ro node-id    te-node-id
       |        |     +--ro flags*     path-attribute-flags
       |        +--:(numbered-link-hop)
       |        |  +--ro numbered-link-hop
       |        |     +--ro link-tp-id    te-tp-id
       |        |     +--ro flags*        path-attribute-flags
       |        +--:(unnumbered-link-hop)
       |        |  +--ro unnumbered-link-hop
       |        |     +--ro link-tp-id    te-tp-id
       |        |     +--ro node-id?      te-node-id
       |        |     +--ro flags*        path-attribute-flags
       |        +--:(label)
       |           +--ro label-hop
       |              +--ro te-label
       |              |  +--ro (technology)?
       |              |  |  +--:(generic)
       |              |  |     +--ro generic?
       |              |  |             rt-types:generalized-label
       |              |  +--ro direction?       te-label-direction
       |              +--ro flags*      path-attribute-flags
       +--ro outgoing-record-route-subobjects
          +--ro outgoing-record-route-subobject* [index]
             +--ro index                        uint32
             +--ro (type)?
```

```
                +--:(numbered-node-hop)
                |  +--ro numbered-node-hop
                |     +--ro node-id    te-node-id
                |     +--ro flags*     path-attribute-flags
                +--:(numbered-link-hop)
                |  +--ro numbered-link-hop
                |     +--ro link-tp-id    te-tp-id
                |     +--ro flags*        path-attribute-flags
                +--:(unnumbered-link-hop)
                |  +--ro unnumbered-link-hop
                |     +--ro link-tp-id    te-tp-id
                |     +--ro node-id?      te-node-id
                |     +--ro flags*        path-attribute-flags
                +--:(label)
                   +--ro label-hop
                      +--ro te-label
                      |  +--ro (technology)?
                      |  |  +--:(generic)
                      |  |     +--ro generic?
                      |  |             rt-types:generalized-label
                      |  +--ro direction?       te-label-direction
                      +--ro flags*      path-attribute-flags
     augment /te:te/te-dev:interfaces/te-dev:interface:
```

Figure 2: RSVP-TE model Tree diagram

2.2.2. RSVP-TE MPLS Model Tree Diagram

Figure 5 shows the YANG tree diagram of the RSVP-TE MPLS YANG model defined in module ietf-rsvp-te-mpls.yang and that augments RSVP-TE module as well as RSVP and TE YANG modules.

```
   module: ietf-rsvp-te-mpls
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp:
       +--rw rsvp-frr-local-revert-delay?   uint32
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces:
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
               /rsvp:interface:
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:sessions:
     augment /rt:routing/rt:control-plane-protocols
               /rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors:
     augment /te:te/te:tunnels/te:tunnel:
       +--rw session-attribute*   identityref
     augment /te:te/te:lsps/te:lsp:
       +--ro session-attribute*   identityref
       +--ro backup-info
          +--ro backup-tunnel-name?         string
          +--ro backup-frr-on?              uint8
          +--ro backup-protected-lsp-num?   uint32
     augment /te:te/te:tunnels/te:tunnel/te:primary-paths
               /te:primary-path/te:lsps/te:lsp:
       +--ro session-attribute*   identityref
       +--ro backup-info
          +--ro backup-tunnel-name?         string
          +--ro backup-frr-on?              uint8
          +--ro backup-protected-lsp-num?   uint32
     augment /te:te/te:tunnels/te:tunnel/te:secondary-paths
               /te:secondary-path/te:lsps/te:lsp:
       +--ro session-attribute*   identityref
       +--ro backup-info
          +--ro backup-tunnel-name?         string
          +--ro backup-frr-on?              uint8
          +--ro backup-protected-lsp-num?   uint32
     augment /te:te/te-dev:interfaces/te-dev:interface:
       +--rw bandwidth-mpls-reservable
          +--rw (bandwidth-value)?
          |  +--:(absolute)
          |  |  +--rw absolute-value?   te-packet-types:bandwidth-kbps
          |  +--:(percentage)
          |     +--rw percent-value?    uint32
          +--rw (bc-model-type)?
             +--:(bc-model-rdm)
             |  +--rw bc-model-rdm
             |     +--rw bandwidth-mpls-constraints
             |        +--rw maximum-reservable?
             |        |       te-packet-types:bandwidth-kbps
             |        +--rw bc-value*             uint32
             +--:(bc-model-mam)
             |  +--rw bc-model-mam
             |     +--rw bandwidth-mpls-constraints
             |        +--rw maximum-reservable?
             |        |       te-packet-types:bandwidth-kbps
             |        +--rw bc-value*             uint32
             +--:(bc-model-mar)
                +--rw bc-model-mar
                   +--rw bandwidth-mpls-constraints
                      +--rw maximum-reservable?
                      |       te-packet-types:bandwidth-kbps
                      +--rw bc-value*             uint32
     augment /te:te/te-dev:interfaces/te-dev:interface:
       +--rw rsvp-te-frr-backups
          +--rw (type)?
```

```
             +--:(static-tunnel)
             |  +--rw static-backups
             |     +--rw static-backup* [backup-tunnel-name]
             |        +--rw backup-tunnel-name
             |                -> /te:te/tunnels/tunnel/name
             +--:(auto-tunnel)
                +--rw auto-tunnel-backups
                   +--rw auto-backup-protection?         identityref
                   +--rw auto-backup-path-computation?   identityref
```

Figure 3: RSVP-TE MPLS Tree diagram

2.3. YANG Modules

2.3.1. RSVP-TE YANG Module

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

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

o ietf-routing-types defined in [RFC8294]

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

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

This module references the following documents: [I-D.ietf-teas-yang-rsvp], [RFC8349], [I-D.ietf-teas-yang-te], [I-D.ietf-teas-yang-te-types], [RFC2210], [RFC4920], [RFC5420], [RFC7570], [RFC4859].

```
<CODE BEGINS> file "ietf-rsvp-te@2021-02-21.yang"
module ietf-rsvp-te {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te";
  prefix rsvp-te;

  import ietf-rsvp {
    prefix rsvp;
    reference
      "draft-ietf-teas-yang-rsvp: A YANG Data Model for
       Resource Reservation Protocol (RSVP)";
  }
  import ietf-routing {
    prefix rt;
    reference
      "RFC8349: A YANG Data Model for Routing Management";
```

```
     }
     import ietf-routing-types {
       prefix rt-types;
       reference
         "RFC8294: Common YANG Data Types for the Routing Area";
     }
     import ietf-te {
       prefix te;
       reference
         "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
          Engineering Tunnels and Interfaces";
     }
     import ietf-te-device {
       prefix te-dev;
       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
         "RFC8776: Common YANG Data Types for Traffic Engineering.";
     }
     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>

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

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.com>

        Editor:   Rakesh Gandhi
                  <mailto:rgandhi@cisco.com>
        Editor:   Xufeng Liu
                  <mailto: xufeng.liu.ietf@gmail.com>

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

        Editor:   Himanshu Shah
                  <mailto:hshah@ciena.com>";
     description
       "This module contains the RSVP-TE YANG generic data model.
        The model fully conforms to the Network Management Datastore
        Architecture (NMDA).

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

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

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

     revision 2021-02-21 {
       description
         "A YANG Data Model for RSVP-TE";
       reference
         "RFCXXXX: A YANG Data Model for RSVP-TE Protocol";
     }

     identity rsvp-message-type {
       description
         "RSVP message types";
     }

     identity rsvp-message-path {
       base rsvp-message-type;
       description
         "RSVP Path message";
       reference
         "RFC2205";
     }

     identity rsvp-message-resv {
       base rsvp-message-type;
       description
         "RSVP Resv message";
       reference
         "RFC2205";
     }

     identity rsvp-message-path-err {
       base rsvp-message-type;
       description
         "RSVP Path-Err message";
       reference
         "RFC2205";
     }

     identity rsvp-message-resv-err {
       base rsvp-message-type;
       description
         "RSVP Resv-Err message";
       reference
         "RFC2205";
     }

     identity rsvp-message-path-tear {
       base rsvp-message-type;
       description
         "RSVP Path Tear message";
       reference
         "RFC2205";
     }

     identity rsvp-message-resv-conf {
       base rsvp-message-type;
       description
         "RSVP Resv Confirm message";
       reference
         "RFC2205";
     }

     identity rsvp-message-srefresh {
       base rsvp-message-type;
       description
         "RSVP SRefresh message";
       reference
         "RFC2961";
     }

     identity rsvp-message-hello {
       base rsvp-message-type;
       description
         "RSVP Hello message";
       reference
         "RFC3209";
     }

     identity rsvp-message-bundle {
       base rsvp-message-type;
       description
         "RSVP Bundle message";
       reference
         "RFC2961";
     }

     identity rsvp-message-notify {
       base rsvp-message-type;
       description
         "RSVP Notify message";
       reference
         "RFC3473";
     }

     /**
      * RSVP-TE LSPs groupings.
      */

     grouping lsp-record-route-information-state {
       description
         "recorded route information grouping";
       container incoming-record-route-subobjects {
         description
           "RSVP recorded route object incoming information";
         list incoming-record-route-subobject {
           when "../../te:origin-type != 'ingress'" {
             description
               "Applicable on non-ingress LSPs only";
           }
           key "index";
           ordered-by user;
           description
             "List of RSVP Path record-route objects";
           uses te-types:record-route-state;
         }
       }
       container outgoing-record-route-subobjects {
         description
           "RSVP recorded route object outgoing information";
         list outgoing-record-route-subobject {
           when "../../te:origin-type != ’egress’" {
             description
               "Applicable on non-egress LSPs only";
           }
           key "index";
           ordered-by user;
           description
             "List of RSVP Resv record-route objects";
           uses te-types:record-route-state;
         }
       }
     }

     grouping lsp-explicit-route-information-state {
       description
         "RSVP-TE LSP explicit-route information";
       container explicit-route-objects {
         description
           "Explicit route object information";
         list incoming-explicit-route-hop {
           when "../../te:origin-type != ’ingress’" {
             description
               "Applicable on non-ingress LSPs only";
           }
           key "index";
           ordered-by user;
           description
             "List of incoming RSVP Path explicit-route objects";
           leaf index {
             type uint32;
             description
               "Explicit route hop index. The index is used to
                identify an entry in the list. The order of entries
                is defined by the user without relying on key values";
           }
           uses te-types:explicit-route-hop;
         }
         list outgoing-explicit-route-hop {
           when "../../te:origin-type != ’egress’" {
             description
               "Applicable on non-egress LSPs only";
           }
           key "index";
           ordered-by user;
           description
             "List of outgoing RSVP Path explicit-route objects";
           leaf index {
             type uint32;
             description
               "Explicit route hop index. The index is used to
                identify an entry in the list. The order of entries
                is defined by the user without relying on key values";
           }
           uses te-types:explicit-route-hop;
         }
       }
     }

     grouping lsp-attributes-flags {
       description
         "Configuration parameters relating to RSVP-TE LSP
          attribute flags";
       leaf-list lsp-attribute {
         type identityref {
           base te-types:lsp-attributes-flags;
         }
         description
           "RSVP per LSP attributes flags";
         reference
           "RFC4920, RFC5420, RFC7570";
       }
     }

     grouping lsp-session-attributes-obj-flags {
       description
         "Configuration parameters relating to RSVP-TE LSP
          session attribute flags";
       reference
         "RFC4859: Registry for RSVP-TE Session Flags";
       leaf-list session-attribute {
         when "../session-attribute !=
               ’te-types:bandwidth-protection-desired’ or
               ../session-attribute !=
               ’te-types:soft-preemption-desired’" {
           description
             "Session attributes applicable to generic technologies
              only.";
         }
         type identityref {
           base te-types:session-attributes-flags;
         }
         description
```

```
           "RSVP session attributes flags";
         reference
           "RFC4859: Registry for RSVP-TE Session Flags";
       }
     }

     grouping lsp-properties {
       description
         "Configuration parameters relating to RSVP-TE LSP
          session attribute flags";
       leaf lsp-signaled-name {
         type string;
         description
           "Sets the session name to use in the session
            attribute object.";
       }
       uses lsp-session-attributes-obj-flags;
       uses lsp-attributes-flags;
     }

     grouping tunnel-properties {
       description
         "RSVP-TE Tunnel properties grouping";
       leaf retry-timer {
         type uint16 {
           range "1..600";
         }
         units "seconds";
         description
           "sets the time between attempts to establish the
            LSP";
       }
     }

     /*** End of RSVP-TE LSP groupings ***/
     /**
      * RSVP-TE generic global properties.
      */

     grouping global-soft-preemption {
       description
         "Configuration for global RSVP-TE soft preemption";
       container global-soft-preemption {
         presence "Enables soft preemption on a node.";
         description
           "Top level container for RSVP-TE soft-preemption";
         leaf soft-preemption-timeout {
           type uint16 {
             range "0..300";
           }
           units "seconds";
           default "0";
           description
             "Timeout value for soft preemption to revert
              to hard preemption";
         }
       }
     }

     /*** End of RSVP-TE generic global properties. ***/
     /**
      * RSVP-TE interface generic groupings.
      */

     grouping rsvp-te-interface-attributes {
       description
         "Top level grouping for RSVP-TE interface properties.";
       container rsvp-te-interface-attributes {
         description
           "Top level container for RSVP-TE interface
            properties";
       }
     }

     /*** End of RSVP-TE generic groupings ***/
     /* RSVP-TE global properties */

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp" {
       description
         "RSVP-TE augmentation to RSVP globals";
       uses global-soft-preemption;
     }

     /* Linkage to the base RSVP all links */

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
       description
         "RSVP-TE generic data augmentation pertaining to interfaces";
       uses rsvp-te-interface-attributes;
     }

     /* Linkage to per RSVP interface */

     augment "/rt:routing/rt:control-plane-protocols/"
```

```
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
           + "rsvp:interface" {
       description
         "RSVP-TE generic data augmentation pertaining to specific
          interface";
       uses rsvp-te-interface-attributes;
     }

     /* add augmentation for sessions and neighbors */

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/"
           + "rsvp:sessions" {
       description
         "RSVP-TE generic data augmentation pertaining to session";
       list session-te {
         key "tunnel-endpoint tunnel-id extended-tunnel-id";
         config false;
         description
           "List of RSVP sessions";
         leaf tunnel-endpoint {
           type inet:ip-address;
           description
             "XX";
         }
         leaf tunnel-id {
           type uint16;
           description
             "XX";
         }
         leaf extended-tunnel-id {
           type inet:ip-address;
           description
             "XX";
         }
         uses rsvp:session-attributes;
       }
     }

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/"
           + "rsvp:sessions/session-te/psbs/psb" {
       description
         "RSVP-TE generic data augmentation pertaining to session";
       /* To be added */
       leaf tspec-average-rate {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Tspec Token Bucket Average Rate";
         reference
           "RFC2210: RSVP with INTSERV";
       }
       leaf tspec-size {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Tspec Token Bucket Burst Rate";
         reference
           "RFC2210";
       }
       leaf tspec-peak-rate {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Tspec Token Bucket Peak Data Rate";
         reference
           "RFC2210";
       }
       leaf min-policed-unit {
         type uint32;
         description
           "Tspec Minimum Policed Unit";
         reference
           "RFC2210";
       }
       leaf max-packet-size {
         type uint32;
         description
           "Tspec Maximum Packet Size";
         reference
           "RFC2210";
       }
     }

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/"
           + "rsvp:sessions/session-te/rsbs/rsb" {
       description
         "RSVP-TE generic data augmentation pertaining to session";
       leaf fspec-average-rate {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Fspec Token Bucket Average Rate";
         reference
```

```
           "RFC2210";
       }
       leaf fspec-size {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Fspec Token Bucket Burst Rate";
         reference
           "RFC2210";
       }
       leaf fspec-peak-rate {
         type rt-types:bandwidth-ieee-float32;
         units "Bytes per second";
         description
           "Fspec Token Bucket Peak Data Rate";
         reference
           "RFC2210";
       }
       leaf min-policed-unit {
         type uint32;
         description
           "Fspec Minimum Policed Unit";
         reference
           "RFC2210";
       }
       leaf max-packet-size {
         type uint32;
         description
           "Fspec Maximum Packet Size";
         reference
           "RFC2210";
       }
     }

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors" {
       description
         "RSVP-TE generic data augmentation pertaining to neighbors";
       /* To be added */
     }

     /**
      * RSVP-TE generic augmentations of generic TE model.
      */
     /* TE tunnel augmentation */

     augment "/te:te/te:tunnels/te:tunnel" {
       when "/te:te/te:tunnels/te:tunnel"
```

```
          + "/te:primary-paths/te:primary-path"
          + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
         description
           "When the path signaling protocol is RSVP-TE ";
       }
       description
         "RSVP-TE generic data augmentation pertaining to TE tunnels";
       uses lsp-properties;
       uses tunnel-properties;
     }

     /* TE LSP augmentation */

     grouping rsvp-te-lsp-error-info {
       description
         "Grouping for RSVP-TE error reporting information";
       leaf rsvp-message-type {
         type identityref {
           base rsvp-message-type;
         }
         description
           "The RSVP message type that delivered the error";
       }
       leaf rsvp-error-code {
         type uint8;
         description
           "RSVP error code";
         reference
           "RFC2205";
       }
       leaf rsvp-error-subcode {
         type uint16;
         description
           "RSVP Error sub-codes";
         reference
           "RFC2205";
       }
     }

     augment "/te:te/te:lsps/te:lsp" {
       when "/te:te/te:lsps/te:lsp"
          + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
         description
           "When the signaling protocol is RSVP-TE ";
       }
       description
         "RSVP-TE generic data augmentation pertaining to specific TE
          LSP";
       leaf associated-rsvp-session {
         type leafref {
           path "/rt:routing/rt:control-plane-protocols/"
              + "rt:control-plane-protocol/rsvp:rsvp/"
              + "rsvp:sessions/session-te/tunnel-id";
         }
         config false;
         description
           "If the signalling protocol specified for this path is
            RSVP-TE, this leaf provides a reference to the associated
            session within the RSVP-TE protocol sessions list, such
            that details of the signaling can be retrieved.";
       }
       uses lsp-properties;
       uses rsvp-te-lsp-error-info;
       uses lsp-explicit-route-information-state;
       uses lsp-record-route-information-state;
     }

     /* TE interface augmentation */

     augment "/te:te/te-dev:interfaces/te-dev:interface" {
       description
         "RSVP-TE generic data augmentation pertaining to specific TE
          interface";
     }
   }
   <CODE ENDS>
```

Figure 4: RSVP TE generic YANG module

2.3.2. RSVP-TE MPLS YANG Module

The RSVP-TE MPLS YANG module "ietf-rsvp-te-mpls" imports the following module(s):

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

o ietf-routing-types defined in [RFC8294]

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

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

This module references the following documents: [I-D.ietf-teas-yang-rsvp], [RFC8349], [I-D.ietf-teas-yang-te-types], [I-D.ietf-teas-yang-te], [RFC3209].

```
   <CODE BEGINS> file "ietf-rsvp-te-mpls@2021-02-21.yang"
   module ietf-rsvp-te-mpls {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-te-mpls";
     prefix rsvp-te-mpls;

     import ietf-rsvp {
       prefix rsvp;
       reference
         "draft-ietf-teas-yang-rsvp: A YANG Data Model for
          Resource Reservation Protocol (RSVP)";
     }
     import ietf-routing {
       prefix rt;
       reference
         "RFC8349: A YANG Data Model for Routing Management";
     }
     import ietf-te-packet-types {
       prefix te-packet-types;
       reference
         "RFC8776: Common YANG Data Types for Traffic Engineering.";
     }
     import ietf-te-types {
       prefix te-types;
       reference
         "RFC8776: Common YANG Data Types for Traffic Engineering.";
     }
     import ietf-te {
       prefix te;
       reference
         "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
          Engineering Tunnels and Interfaces";
     }
     import ietf-te-device {
       prefix te-dev;
       reference
         "draft-ietf-teas-yang-te: A YANG Data Model for Traffic
          Engineering Tunnels and Interfaces";
     }

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

        Editor:   Vishnu Pavan Beeram
```

```
                  <mailto:vbeeram@juniper.net>

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.com>

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

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

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

        Editor:   Himanshu Shah
                  <mailto:hshah@ciena.com>";
     description
       "Latest update to MPLS RSVP-TE YANG data model.
        The model fully conforms to the Network Management Datastore
        Architecture (NMDA).

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

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

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

     revision 2021-02-21 {
       description
         "Update to MPLS RSVP-TE YANG initial revision.";
       reference
         "RFCXXXX: A YANG Data Model for RSVP-TE Protocol";
     }

     /* RSVP-TE MPLS LSPs groupings */

     grouping lsp-attributes-flags-mpls {
       description
         "Configuration parameters relating to RSVP-TE MPLS LSP
          attribute flags";
     }

     grouping lsp-session-attributes-obj-flags-mpls {
       description
         "Configuration parameters relating to RSVP-TE MPLS LSP
          session attribute flags";
       reference
         "RFC4859: Registry for RSVP-TE Session Flags";
       leaf-list session-attribute {
         when "../session-attribute =
               'te-types:bandwidth-protection-desired' or
               ../session-attribute =
               'te-types:soft-preemption-desired'" {
           description
             "Session attributes applicable to mpls technology";
         }
         type identityref {
           base te-types:session-attributes-flags;
         }
         description
           "RSVP session attributes flags";
         reference
           "RFC4859: Registry for RSVP-TE Session Flags";
       }
     }

     grouping tunnel-properties-mpls {
       description
         "Top level grouping for LSP properties.";
       uses lsp-session-attributes-obj-flags-mpls;
       uses lsp-attributes-flags-mpls;
     }

     grouping lsp-properties-mpls {
       description
         "Top level grouping for LSP properties.";
       uses lsp-session-attributes-obj-flags-mpls;
       uses lsp-attributes-flags-mpls;
     }

     /* End of RSVP-TE MPLS LSPs groupings */
     /* MPLS RSVP-TE interface groupings */

     grouping rsvp-te-interface-state {
       description
```

```
         "The RSVP-TE interface state grouping";
       leaf over-subscribed-bandwidth {
         type te-packet-types:bandwidth-kbps;
         description
           "The amount of over-subscribed bandwidth on
            the interface";
       }
     }

     grouping rsvp-te-interface-softpreemption-state {
       description
         "The RSVP-TE interface preeemptions state grouping";
       container interface-softpreemption-state {
         description
           "The RSVP-TE interface preeemptions state grouping";
         leaf soft-preempted-bandwidth {
           type te-packet-types:bandwidth-kbps;
           description
             "The amount of soft-preempted bandwidth on
              this interface";
         }
         list lsps {
           key "source destination tunnel-id lsp-id "
             + "extended-tunnel-id";
           description
             "List of LSPs that are soft-preempted";
           leaf source {
             type leafref {
               path "/te:te/te:lsps/te:lsp/"
                  + "te:source";
             }
             description
               "Tunnel sender address extracted from
                SENDER_TEMPLATE  object";
             reference
               "RFC3209";
           }
           leaf destination {
             type leafref {
               path "/te:te/te:lsps/te:lsp/"
                  + "te:destination";
             }
             description
               "Tunnel endpoint address extracted from
                SESSION object";
             reference
               "RFC3209";
           }
```

```
          leaf tunnel-id {
            type leafref {
              path "/te:te/te:lsps/te:lsp/"
                 + "te:tunnel-id";
            }
            description
              "Tunnel identifier used in the SESSION
               that remains constant over the life
               of the tunnel.";
            reference
              "RFC3209";
          }
          leaf lsp-id {
            type leafref {
              path "/te:te/te:lsps/te:lsp/"
                 + "te:lsp-id";
            }
            description
              "Identifier used in the SENDER_TEMPLATE
               and the FILTER_SPEC that can be changed
               to allow a sender to share resources with
               itself.";
            reference
              "RFC3209";
          }
          leaf extended-tunnel-id {
            type leafref {
              path "/te:te/te:lsps/te:lsp/"
                 + "te:extended-tunnel-id";
            }
            description
              "Extended Tunnel ID of the LSP.";
            reference
              "RFC3209";
          }
          leaf type {
            type leafref {
              path "/te:te/te:lsps/te:lsp/"
                 + "te:type";
            }
            description
              "LSP type P2P or P2MP";
          }
        }
      }
    }

    grouping bandwidth-mpls-constraints {
      description
        "Bandwidth constraints.";
      container bandwidth-mpls-constraints {
        description
          "Holds the bandwidth constraints properties";
        leaf maximum-reservable {
          type te-packet-types:bandwidth-kbps;
          description
            "The maximum reservable bandwidth on the
             interface in kbps";
        }
        leaf-list bc-value {
          type uint32 {
            range "0..4294967295";
          }
          max-elements 8;
          description
            "The bandwidth constraint type";
        }
      }
    }

    grouping bandwidth-constraint-values {
      description
        "Packet bandwidth contraints values";
      choice value-type {
        description
          "Value representation";
        case percentages {
          container perc-values {
            uses bandwidth-mpls-constraints;
            description
              "Percentage values";
          }
        }
        case absolutes {
          container abs-values {
            uses bandwidth-mpls-constraints;
            description
              "Absolute values";
          }
        }
      }
    }

    grouping bandwidth-mpls-reservable {
      description
        "Interface bandwidth reservable configuration grouping";
      container bandwidth-mpls-reservable {
        description
          "Interface bandwidth reservable container";
        choice bandwidth-value {
          description
            "Reservable bandwidth configuration choice";
          case absolute {
            leaf absolute-value {
              type te-packet-types:bandwidth-kbps;
              description
                "Absolute value of the bandwidth";
            }
          }
          case percentage {
            leaf percent-value {
              type uint32 {
                range "0..4294967295";
              }
              description
                "Percentage reservable bandwidth";
            }
            description
              "The maximum reservable bandwidth on the
               interface";
          }
        }
        choice bc-model-type {
          description
            "Reservable bandwidth percentage capacity
             values.";
          case bc-model-rdm {
            container bc-model-rdm {
              description
                "Russian Doll Model Bandwidth Constraints.";
              uses bandwidth-mpls-constraints;
            }
          }
          case bc-model-mam {
            container bc-model-mam {
              uses bandwidth-mpls-constraints;
              description
                "Maximum Allocation Model Bandwidth
                 Constraints.";
            }
          }
          case bc-model-mar {
            container bc-model-mar {
              uses bandwidth-mpls-constraints;
              description
                "Maximum Allocation with Reservation Model
                 Bandwidth Constraints.";
            }
          }
        }
      }
    }

    /* End of RSVP-TE interface groupings */
    /* RSVP-TE FRR groupings */

    grouping rsvp-te-frr-auto-tunnel-backup {
      description
        "Auto-tunnel backup configuration grouping";
      leaf auto-backup-protection {
        type identityref {
          base te-packet-types:backup-protection-type;
        }
        default "te-packet-types:backup-protection-node-link";
        description
          "Describes whether the backup should offer
           protection against link, node, or either";
      }
      leaf auto-backup-path-computation {
        type identityref {
          base te-types:path-computation-srlg-type;
        }
        description
          "FRR backup computation type";
      }
    }

    grouping rsvp-te-frr-backups {
      description
        "Top level container for RSVP-TE FRR backup parameters";
      container rsvp-te-frr-backups {
        description
          "RSVP-TE facility backup properties";
          choice type {
            description
              "FRR backup tunnel type";
            case static-tunnel {
              container static-backups {
                description
                  "List of static backups";
                list static-backup {
                  key "backup-tunnel-name";
                  description
                    "List of static backup tunnels that
                     protect the RSVP-TE interface.";
                  leaf backup-tunnel-name {
                    type leafref {
                      path "/te:te/te:tunnels/te:tunnel/te:name";
                    }
                    description
                      "FRR Backup tunnel name";
                  }
                }
              }
            }
            case auto-tunnel {
              container auto-tunnel-backups {
                description
                  "Auto-tunnel choice";
                uses rsvp-te-frr-auto-tunnel-backup;
              }
            }
          }
      }
    }

    grouping lsp-backup-info-state {
      description
        "LSP backup information grouping";
      leaf backup-tunnel-name {
        type string;
        description
          "If an LSP has an FRR backup LSP that can protect it,
           this field identifies the tunnel name of the backup LSP.
           Otherwise, this field is empty.";
      }
      leaf backup-frr-on {
        type uint8;
        description
          "Whether currently this backup is carrying traffic";
      }
      leaf backup-protected-lsp-num {
        type uint32;
        description
          "Number of LSPs protected by this backup";
      }
    }

    grouping lsp-backup-info {
      description
```

```
          "Backup/bypass LSP related information";
        container backup-info {
          description
            "backup information";
          uses lsp-backup-info-state;
        }
      }

      /*** End of RSVP-TE FRR backup information ***/

      /* RSVP-TE global properties */

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rsvp:rsvp" {
        description
          "RSVP-TE augmentation to RSVP globals";
        leaf rsvp-frr-local-revert-delay {
          type uint32;
          description
            "Time to wait after primary link is restored
             before node attempts local revertive
             procedures.";
        }
      }

      /* Linkage to the base RSVP all interfaces */

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces" {
        description
          "Augmentations for RSVP-TE MPLS all interfaces properties";
        /* To be added */
      }

      /* Linkage to per RSVP interface */

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces/"
            + "rsvp:interface" {
        description
          "Augmentations for RSVP-TE MPLS per interface properties";
        /* To be added */
      }

      /* add augmentation for sessions neighbors */

      augment "/rt:routing/rt:control-plane-protocols/"
            + "rt:control-plane-protocol/rsvp:rsvp/"
```

```
           + "rsvp:sessions" {
       description
         "Augmentation for RSVP-TE MPLS sessions";
       /* To be added */
     }

     augment "/rt:routing/rt:control-plane-protocols/"
           + "rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors" {
       description
         "Augmentations for RSVP-TE MPLS neighbors properties";
       /* To be added */
     }

     /**
      * Augmentation to TE generic module
      */

     augment "/te:te/te:tunnels/te:tunnel" {
       description
         "Augmentations for RSVP-TE MPLS TE tunnel properties";
       uses tunnel-properties-mpls;
     }

     augment "/te:te/te:lsps/te:lsp" {
       when "/te:te/te:lsps/te:lsp"
          + "/te:signaling-type = 'te-types:path-setup-rsvp'" {
         description
           "When the signaling protocol is RSVP-TE ";
       }
       description
         "RSP-TE MPLS LSP state properties";
       uses lsp-properties-mpls;
       uses lsp-backup-info;
     }

     augment "/te:te/te:tunnels/te:tunnel/te:primary-paths"
           + "/te:primary-path/te:lsps/te:lsp" {
       when "/te:te/te:tunnels/te:tunnel"
          + "/te:secondary-paths/te:secondary-path/"
          + "te:signaling-type = 'te-types:path-setup-rsvp'" {
         description
           "When the signaling protocol is RSVP-TE ";
       }
       description
         "RSVP-TE MPLS LSP state properties";
       uses lsp-properties-mpls;
       uses lsp-backup-info;
     }
     augment "/te:te/te:tunnels/te:tunnel/te:secondary-paths"
           + "/te:secondary-path/te:lsps/te:lsp" {
       when "/te:te/te:tunnels/te:tunnel"
          + "/te:secondary-paths/te:secondary-path/"
          + "te:signaling-type = 'te-types:path-setup-rsvp'" {
         description
           "When the signaling protocol is RSVP-TE ";
       }
       description
         "RSVP-TE MPLS LSP state properties";
       uses lsp-properties-mpls;
       uses lsp-backup-info;
     }

     augment "/te:te/te-dev:interfaces/te-dev:interface" {
       description
         "RSVP reservable bandwidth configuration properties";
       uses bandwidth-mpls-reservable;
     }

     augment "/te:te/te-dev:interfaces/te-dev:interface" {
       description
         "RSVP reservable bandwidth configuration properties";
       uses rsvp-te-frr-backups;
     }
   }
   <CODE ENDS>
```

Figure 5: RSVP TE MPLS YANG module

3. IANA Considerations

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

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

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

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

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

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

4. 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(s) defined in this document 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.

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/globals: The data nodes defined defined in this document and under this branch are applicable device-wide and can affect all RSVP established sessions. Unauthorized access to this container can potentially cause disruptive event(s) on all established sessions.

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/rsvp:sessions: The data nodes defined in this document and under this branch are applicable to one or all RSVP-TE session(s). Unauthorized access to this container can potentially affect the impacted RSVP session(s).

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces: The data nodes defined defined in this document and under this branch are applicable to one or all RSVP interfaces. Unauthorized access to this container can potentially affect established session(s) over impacted interface(s).

5. Acknowledgement

The authors would like to thank Lou Berger for reviewing and providing valuable feedback on this document.

6. 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

7. References

7.1. Normative References

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

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

[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-13 (work in progress), November 2019.

[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>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

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

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

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

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

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

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

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

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

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

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

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

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

7.2. Informative References

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <https://www.rfc-editor.org/info/rfc2210>.

[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>.

[RFC4859] Farrel, A., "Codepoint Registry for the Flags Field in the Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Session Attribute Object", RFC 4859, DOI 10.17487/RFC4859, April 2007, <https://www.rfc-editor.org/info/rfc4859>.

[RFC4920] Farrel, A., Ed., Satyanarayana, A., Iwata, A., Fujita, N., and G. Ash, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE", RFC 4920, DOI 10.17487/RFC4920, July 2007, <https://www.rfc-editor.org/info/rfc4920>.

[RFC5420] Farrel, A., Ed., Papadimitriou, D., Vasseur, JP., and A. Ayyangar, "Encoding of Attributes for MPLS LSP Establishment Using Resource Reservation Protocol Traffic Engineering (RSVP-TE)", RFC 5420, DOI 10.17487/RFC5420, February 2009, <https://www.rfc-editor.org/info/rfc5420>.

[RFC7570] Margaria, C., Ed., Martinelli, G., Balls, S., and B. Wright, "Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)", RFC 7570, DOI 10.17487/RFC7570, July 2015, <https://www.rfc-editor.org/info/rfc7570>.

Authors' Addresses

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Rakesh Gandhi
Cisco Systems, Inc.

Email: rgandhi@cisco.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Igor Bryskin
Huawei Technologies

Email: Igor.Bryskin@huawei.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 5 August 2024

T. Saad
R. Gandhi
Cisco Systems Inc
X. Liu
Alef Edge
V. P. Beeram
Juniper Networks
I. Bryskin
Individual

2 February 2024

A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces
draft-ietf-teas-yang-te-36

Abstract

This document defines a YANG data model for the provisioning and management of Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The model covers data that is independent of any technology or dataplane encapsulation and is divided into two YANG modules that cover device-specific, and device independent data.

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

This document describes a YANG data model for Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs), and interfaces. The data model is divided into two YANG modules. The module 'ietf-te.yang' includes data that is generic and device-independent, while the module 'ietf-te-device.yang' includes data that is device-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 modules that model TE signaling protocols, such as RSVP-TE ([RFC3209], [RFC3473]), or Segment-Routing TE (SR-TE) [RFC9256] will augment the generic TE YANG module.

2. Terms and Conventions

2.1. Requirements Language

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

2.2. Terminology

The following terms are defined in [RFC6241] and are used in this specification:

* client

* configuration data

- state data

This document also makes use of the following terminology introduced in the YANG Data Modeling Language [RFC7950]:

- augment
- data model
- data node

2.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te-types	ietf-te-types	[I-D.draft-ietf-teas-rfc8776-update]
te-packet-types	ietf-te-packet-types	[I-D.draft-ietf-teas-rfc8776-update]
te	ietf-te	this document
te-dev	ietf-te-device	this document

Table 1: Prefixes and corresponding YANG modules

2.4. Model Tree Diagrams

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

3. Design Considerations

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

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

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

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

* The generic TE YANG data model 'ietf-te' contains device independent data and can be used to model data off a device (e.g. on a TE controller). When the model is used to manage a specific device, the model contains the TE Tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnel' list contains all TE Tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

* The device-specific TE data is defined in 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.

* Suitable defaults are specified for all configurable elements.

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

3.1. State Data Organization

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

4. Model Overview

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

4.1. Module Relationship

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

The TE data models for specific instances of data plane technology exist in separate YANG modules that augment the generic TE YANG data model. The TE data models for specific instances of signaling protocols are outside the scope of this document and are defined in other documents. For example, the RSVP-TE YANG model augmentation of the TE model is covered in a separate document.

```
     TE generic     +---------+         o: augment
     module         | ietf-te |o-------------+
                    +---------+               \
                           o                   \
                           |\                   \
                           | \ TE device module  \
                           |  +----------------+  \
                           |  | ietf-te-device |   \
                           |  +----------------+    \
                           |       o                 \
                           |     /                    \
                           |   /                       \
                    +---------------+           +---------------+
     RSVP-TE module | ietf-rsvp-te^ |o .       | ietf-te-mpls^ |
                    +---------------+  \       +---------------+
                       |                \
                       |                 \
                       |                  \
                       |                   \
                       |                    \
                       o                 +-------------------+
                    +------------+       | ietf-rsvp-otn-te^ |
     RSVP module    | ietf-rsvp^ |       +-------------------+
                    +------------+          RSVP-TE with OTN
                                            extensions

                   X---oY indicates that module X augments module Y
                   ^ indicates a module defined in other documents
```

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

5. TE YANG Model

The generic TE YANG module ('ietf-te') is meant for the management and operation of a TE network. This includes creating, modifying and retrieving information about TE Tunnels, LSPs, and interfaces and their associated attributes (e.g. Administrative-Groups, SRLGs, etc.).

A full tree diagram of the TE model is shown in the Appendix in Figure 13.

5.1. Module Structure

The 'te' container is the top level container in the 'ietf-te' module. The presence of the 'te' container enables TE function system wide. Below provides further descriptions of containers that exist under the 'te' top level container.

There are three further containers grouped under the 'te' container as shown in Figure 2 and described below.

globals:

The 'globals' container maintains the set of global TE attributes that can be applicable to TE Tunnels and interfaces.

tunnels:

The 'tunnels' container includes the list of TE Tunnels that are instantiated. Refer to Section 5.1.2 for further details on the properties of a TE Tunnel.

lsps:

The 'lsps' container includes the list of TE LSP(s) that are instantiated for TE Tunnels. Refer to Section 5.1.3 for further details on the properties of a TE LSP.

The model also contains two Remote Procedure Calls (RPCs) as shown in Figure 13 and described below.

tunnels-path-compute:

A RPC to request path computation for a specific TE Tunnel. The RPC allows requesting path computation using atomic and stateless operation. A tunnel may also be configured in 'compute-only' mode to provide stateful path updates - see Section 5.1.2 for further details.

tunnels-action:

An RPC to request a specific action (e.g. reoptimize, or tear-and-setup) to be taken on a specific tunnel or all tunnels.

Figure 13 shows the relationships of these containers and RPCs within the 'ietf-te' module.

```
   module: ietf-te
     +--rw te!
        +--rw globals
        |     ...
        +--rw tunnels
        |     ...
        +--ro lsps
              ...

     rpcs:
       +---x tunnels-path-compute
       |  +---w input
       |  |     ...
       |  +--ro output
       |        ...
       +---x tunnels-actions
          +---w input
          |     ...
          +--ro output
                ...
```

Figure 2: TE Tunnel model high-level YANG tree view

5.1.1. TE Globals

The 'globals' container covers properties that control a TE feature's behavior system-wide, and its respective state as shown in Figure 3 and described in the text that follows.

```
        +--rw globals
        |  +--rw named-admin-groups
        |  |  +--rw named-admin-group* [name]
        |  |        ...
        |  +--rw named-srlgs
        |  |  +--rw named-srlg* [name]
        |  |        ...
        |  +--rw named-path-constraints
        |     +--rw named-path-constraint* [name]
```

Figure 3: TE globals YANG subtree high-level structure

named-admin-groups:

A YANG container for the list of named (extended) administrative groups that may be applied to TE links.

named-srlgs:

A YANG container for the list of named Shared Risk Link Groups (SRLGs) that may be applied to TE links.

named-path-constraints:

A YANG container for a list of named path constraints. Each named path constraint is composed of a set of constraints that can be applied during path computation. A named path constraint can be applied to multiple TE Tunnels. Path constraints may also be specified directly under the TE Tunnel. The path constraints specified under the TE Tunnel take precedence over the path constraints derived from the referenced named path constraint. A named path constraint entry can be formed of the path constraints shown in Figure 4:

```
+--rw named-path-constraints
   +--rw named-path-constraint* [name]
           {te-types:named-path-constraints}?
      +--rw name                             string
      +--rw te-bandwidth
      |     ...
      +--rw link-protection?                 identityref
      +--rw setup-priority?                  uint8
      +--rw hold-priority?                   uint8
      +--rw signaling-type?                  identityref
      +--rw path-metric-bounds
      |     ...
      +--rw path-affinities-values
      |     ...
      +--rw path-affinity-names
      |     ...
      +--rw path-srlgs-lists
      |     ...
      +--rw path-srlgs-names
      |     ...
      +--rw disjointness?
      |       te-path-disjointness
      +--rw explicit-route-objects
      |     ...
      +--rw path-in-segment!
      |     ...
      +--rw path-out-segment!
            ...
```

Figure 4: Named path constraints YANG subtree

o name: A YANG leaf that holds the named path constraint entry. This is unique in the list and used as a key.

- te-bandwidth: A YANG container that holds the technology agnostic TE bandwidth constraint.

- link-protection: A YANG leaf that holds the link protection type constraint required for the links to be included in the computed path.

- setup/hold priority: YANG leafs that hold the LSP setup and hold admission priority as defined in [RFC3209].

- signaling-type: A YANG leaf that holds the LSP setup type, such as RSVP-TE or SR.

- path-metric-bounds: A YANG container that holds the set of metric bounds applicable on the computed TE tunnel path.

- path-affinities-values: A YANG container that holds the set of affinity values and mask to be used during path computation.

- path-affinity-names: A YANG container that holds the set of named affinity constraints and corresponding inclusion or exclusion instructions for each to be used during path computation.

- path-srlgs-lists: A YANG container that holds the set of SRLG values and corresponding inclusion or exclusion instructions to be used during path computation.

- path-srlgs-names: A YANG container that holds the set of named SRLG constraints and corresponding inclusion or exclusion instructions for each to be used during path computation.

- disjointness: The level of resource disjointness constraint that the secondary path of a TE tunnel has to adhere to.

- explicit-route-objects: A YANG container that holds path constraints in the form of route entries present in following two lists:

  - 'route-object-exclude-always': a list of route entries that are always excluded from the path computation. The exclusion of a route entry in this list during path computation is not order sensitive.

- + 'route-object-include-exclude': a list of route entries to include or exclude route entry constraints for the path computation. The constraint type (include or exclude) is specified with each route entry. The path computation considers route entry constraints in the order they appear in this list. Once a route entry constraint is consumed from this list, it is not considered any further in the computation of the path.

The 'route-object-include-exclude' is used to configure constraints on which route objects (e.g., nodes, links) are included or excluded in the path computation.

The interpretation of an empty 'route-object-include-exclude' list depends on the TE Tunnel (end-to-end or Tunnel Segment) and on the specific path, according to the following rules:

1. An empty 'route-object-include-exclude' list for the primary path of an end-to-end TE Tunnel indicates that there are no route objects to be included or excluded in the path computation.

2. An empty 'route-object-include-exclude' list for the primary path of a TE Tunnel Segment indicates that no primary LSP is required for that TE Tunnel.

3. An empty 'route-object-include-exclude' list for a reverse path means it always follows the forward path (i.e., the TE Tunnel is co-routed). When the 'route-object-include-exclude' list is not empty, the reverse path is routed independently of the forward path.

4. An empty 'route-object-include-exclude' list for the secondary (forward) path indicates that the secondary path has the same endpoints as the primary path.

- o path-in-segment: A YANG container that contains a list of label restrictions that have to be taken into considerations when crossing domains. This TE tunnel segment in this case is being stitched to the upstream TE tunnel segment.

- o path-out-segment: A YANG container that contains a list of label restrictions that have to be taken into considerations when crossing domains. The TE tunnel segment in this case is being stitched to the downstream TE tunnel segment.

5.1.2. TE Tunnels

The 'tunnels' container holds the list of TE Tunnels that are provisioned on ingress LER devices in the network as shown in Figure 5.

```
module: ietf-te
  +--rw te
     +--rw tunnels
        +--rw tunnel* [name]
           +--rw name                                  string
           +--rw alias?                                string
           +--rw identifier?                           uint32
           +--rw color?                                uint32
           +--rw description?                          string
           +--rw admin-state?                          identityref
           +--ro operational-state?                    identityref
           +--rw encoding?                             identityref
           +--rw switching-type?                       identityref
           +--rw source
           |     ...
           +--rw destination
           |     ...
           +--rw bidirectional?                        boolean
           +--rw controller
           |     ...
           +--rw reoptimize-timer?                     uint16
           +--rw association-objects
           |     ...
           +--rw protection
           |     ...
           +--rw restoration
           |     ...
           +--rw network-id?                           nw:network-id
           +--rw te-topology-identifier
           |     ...
           +--rw te-bandwidth
           |     ...
           +--rw link-protection?                      identityref
           +--rw setup-priority?                       uint8
           +--rw hold-priority?                        uint8
           +--rw signaling-type?                       identityref
           +--rw hierarchy
           |     ...
           +--rw primary-paths
           |     ...
           +--rw secondary-paths
           |     ...
```

```
             +--rw secondary-reverse-paths
             |      ...
             +---x tunnel-action
             |      ...
             +---x protection-external-commands
                    ...
```

Figure 5: TE Tunnel YANG subtree structure

When the model is used to manage a specific device, the 'tunnel' list contains the TE Tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnel' list contains all TE Tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

The TE Tunnel model allows the configuration and management of the following TE tunnel objects:

TE Tunnel:

A YANG container of one or more TE LSPs established between the source and destination TE Tunnel termination points.

TE Path:

An engineered path that once instantiated in the forwarding plane can be used to forward traffic from the source to the destination TE Tunnel termination points.

TE LSP:

A TE LSP is a connection-oriented service established over a TE Path and that allows the delivery of traffic between the TE Tunnel source and destination termination points.

TE Tunnel Segment:

A part of a multi-domain TE Tunnel that is within a specific network domain.

The TE Tunnel has a number of attributes that are set directly under the tunnel (as shown in Figure 5). The main attributes of a TE Tunnel are described below:

operational-state:

A YANG leaf that holds the operational state of the tunnel.

name:

A YANG leaf that holds the name of a TE Tunnel. The name of the TE Tunnel uniquely identifies the tunnel within the TE tunnel list. The name of the TE Tunnel can be formatted as a Uniform Resource Indicator (URI) by including the namespace to ensure uniqueness of the name amongst all the TE Tunnels present on devices and controllers. The configured TE Tunnels can be reported with the name of the device embedded within the TE Tunnel name. For initiated TE Tunnels from the controller, the controller is responsible to ensures that TE Tunnel names are unique.

alias:

A YANG leaf that holds an alternate name to the TE tunnel. Unlike the TE tunnel name, the alias can be modified at any time during the lifetime of the TE tunnel.

identifier:

A YANG leaf that holds an identifier of the tunnel. This identifier is unique amongst tunnels originated from the same ingress device.

color:

A YANG leaf that holds the color associated with the TE tunnel. The color is used to map or steer services that carry matching color on to the TE tunnel as described in [RFC9012].

admin-state:

A YANG leaf that holds the tunnel administrative state. The administrative status in state datastore transitions to 'tunnel-admin-up' when the tunnel used by the client layer, and to 'tunnel-admin-down' when it is not used by the client layer.

operational-state:

A YANG leaf that holds the tunnel operational state.

encoding/switching:

The 'encoding' and 'switching-type' are YANG leafs that define the specific technology in which the tunnel operates in as described in [RFC3945].

source/destination:

YANG containers that hold the tunnel source and destination node endpoints identities, including:

- te-node-id: A YANG leaf that holds the identifier of the source or destination of the TE Tunnel TE node identifiers as defined in [I-D.draft-ietf-teas-rfc8776-update].

- node-id: A YANG leaf that holds the identifier of the source or destination of the TE Tunnel node identifiers as defined in [RFC8345].

- tunnel-tp-id: A YANG leaf that holds the identifier of the source or destination of the TE Tunnel Termination Points (TTPs) as defined in [RFC8795]. The TTP identifiers are optional on nodes that have a single TTP per node. For example, TTP identifiers are optional for packet (IP/MPLS) routers.

bidirectional:

A YANG leaf that when present indicates the LSP of a TE Tunnel is bidirectional as defined in [rfc3473].

controller:

A YANG container that holds tunnel data relevant to an optional external TE controller that may initiate or control a tunnel. This target node may be augmented by external module(s), for example, to add data for PCEP initiated and/or delegated tunnels.

reoptimize-timer:

A YANG leaf to set the interval period for tunnel reoptimization.

association-objects:

A YANG container that holds the set of associations of the TE Tunnel to other TE Tunnels. Associations at the TE Tunnel level apply to all paths of the TE Tunnel. The TE tunnel associations can be overridden by associations configured directly under the TE Tunnel path.

protection:

A YANG container that holds the TE Tunnel protection properties.

restoration:

A YANG container that holds the TE Tunnel restoration properties.

te-topology-identifier:

A YANG container that holds the topology identifier associated with the topology where paths for the TE tunnel are computed as defined in [RFC8795].

network-id:

A YANG leaf that can optionally be used to identify the network topology where paths for the TE tunnel are computed as defined in [RFC8345].

hierarchy:

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

o dependency-tunnels: A set of hierarchical TE Tunnels provisioned or to be provisioned in the immediate lower layer that this TE tunnel depends on for multi-layer path computation. A dependency TE Tunnel is provisioned if and only if it is used (selected by path computation) at least by one client layer TE Tunnel. The TE link in the client layer network topology supported by a dependent TE Tunnel is dynamically created only when the dependency TE Tunnel is actually provisioned.

o hierarchical-link: A YANG container that holds the identity of the hierarchical link (in the client layer) that is supported by this TE Tunnel. The endpoints of the hierarchical link are defined by TE tunnel source and destination node endpoints. The hierarchical link can be identified by its source and destination link termination point identifiers.

primary-paths:

A YANG container that holds the list of primary paths. A primary path is identified by 'name'. A primary path is selected from the list to instantiate a primary forwarding LSP for the tunnel. The list of primary paths is visited by order of preference. A primary path has the following attributes:

- primary-reverse-path: A YANG container that holds properties of the primary reverse path. The reverse path is applicable to bidirectional TE Tunnels.

- candidate-secondary-paths: A YANG container that holds a list of candidate secondary paths which may be used for the primary path to support path protection. The candidate secondary path(s) reference path(s) from the tunnel secondary paths list. The preference of the secondary paths is specified within the list and dictates the order of visiting the secondary path from the list. The attributes of a secondary path can be defined separately from the primary path. The attributes of a secondary path will be inherited from the associated 'active' primary when not explicitly defined for the secondary path.

secondary-paths:

A YANG container that holds the set of secondary paths. A secondary path is identified by 'name'. A secondary path can be referenced from the TE Tunnel's 'candidate-secondary-path' list.

secondary-reverse-paths:

A YANG container that holds the set of secondary reverse paths. A secondary reverse path is identified by 'name'. A secondary reverse path can be referenced from the TE Tunnel's 'candidate-secondary-reverse-paths' list. A secondary reverse path contains attributes similar to a primary path.

The following set of common path attributes are shared for primary (forward and reverse) and secondary paths:

path-computation-method:

A YANG leaf that specifies the method used for computing the TE path.

path-computation-server:

A YANG container that holds the path computation server properties when the path is externally queried.

compute-only:

A path of a TE Tunnel is, by default, provisioned so that it can be instantiated in the forwarding plane so that it can carry traffic as soon as a valid path is computed. In some cases, a TE path may be configured only for the purpose of computing a path and reporting it without the need to instantiate the LSP or commit any resources. In such a case, the path is configured in 'compute-only' mode to distinguish it from the default behavior. A 'compute-only' path is configured as a usual with the associated per path constraint(s) and properties on a device or TE controller. The device or TE controller computes the feasible path(s) subject to configured constraints. A client may query the 'compute-only' computed path properties 'on-demand', or alternatively, can subscribe to be notified of computed path(s) and whenever the path properties change.

use-path-computation:

A YANG leaf that indicates whether or not path computation is to be used for a specified path.

lockdown:

A YANG leaf that when set indicates the existing path should not be reoptimized after a failure on any of its traversed links.

path-scope:

A YANG leaf that specifies the path scope if segment or an end-to-end path.

preference:

A YANG leaf that specifies the preference for the path. The lower the number higher the preference.

k-requested-paths:

A YANG leaf that specifies the number of k-shortest-paths requested from the path computation server and returned sorted by its optimization objective.

association-objects:

A YANG container that holds a list of tunnel association properties.

optimizations:

A YANG container that holds the optimization objectives that path computation will use to select a path.

named-path-constraint:

A YANG leafref that references an entry from the global list of named path constraints.

te-bandwidth:

A YANG container that holds the path bandwidth (see [I-D.draft-ietf-teas-rfc8776-update]).

link-protection:

A YANG leaf that specifies the link protection type required for the links to be included the computed path (see [I-D.draft-ietf-teas-rfc8776-update]).

setup/hold-priority:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

signaling-type:

see description provided in Section 5.1.1. This value overrides the provided one in the referenced named-path-constraint.

path-metric-bounds:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-affinities-values:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-affinity-names:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-srlgs-lists:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-srlgs-names:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

disjointness:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

explicit-route-objects:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-in-segment:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

path-out-segment:

see description provided in Section 5.1.1. These values override those provided in the referenced named-path-constraint.

computed-paths-properties:

A YANG container that holds properties for the list of computed paths.

computed-path-error-infos:

A YANG container that holds a list of errors related to the path.

lsp-provisioning-error-infos:

A YANG container that holds the list of LSP provisioning error information. The TE system populates entries in this list whenever an error is encountered during the LSP provisioning.

computed-path-error-infos:

A YANG container that holds the list of path computation error information. The TE system populates entries in this list whenever an error is encountered during the compuation of the TE path.

path-compute-info:

A YANG grouping that contains leafs representing the path attributes that are passed to the TE path computation engine to be considered during the path computation. This includes:

- path constraints,

- path optimization objectives, and

- path assocociations

Note, unless overriden under a specific path of the TE tunnel, the TE tunnel's primary path constraints, optimization objectives, and associations are inherited by the primary reverse path, secondary path and secondary reverse path.

lsps:

A YANG container that holds a list of LSPs that have been instantiated for this specific path.

In addition to the path common attributes, the primary path has the following attributes that are not present in the secondary path:

* Only the primary path contains the list of 'candidate-secondary-paths' that can protect the primary path.

* Only the primary path can contain a primary-reverse-path associated with the primary path (and its associated list of 'candidate-secondary-reverse-path').

5.1.3. TE LSPs

The 'lsps' container includes the set of TE LSP(s) that have been instantiated. A TE LSP is identified by a 3-tuple ('tunnel-name', 'lsp-id', 'node').

When the model is used to manage a specific device, the 'lsps' list contains all TE LSP(s) that traverse the device (including ingressing, transiting and egressing the device).

When the model is used to manage a TE controller, the 'lsps' list contains the TE LSP(s) on devices managed by the controller that act as ingress, and may optionally include TE LSPs on devices managed by the controller that act as transit or egress role.

5.2. Tree Diagram

Figure 6 shows the tree diagram of depth=4 for the generic TE YANG model defined in modules 'ietf-te.yang'. The full tree diagram is shown in Section 13.

```
   module: ietf-te
     +--rw te
        +--rw enable?    boolean
        +--rw globals
        |  +--rw named-admin-groups
        |  |  +--rw named-admin-group* [name]
        |  |          {te-types:extended-admin-groups,
        |  |           te-types:named-extended-admin-groups}?
        |  |        ...
        |  +--rw named-srlgs
        |  |  +--rw named-srlg* [name] {te-types:named-srlg-groups}?
        |  |        ...
        |  +--rw named-path-constraints
        |     +--rw named-path-constraint* [name]
        |             {te-types:named-path-constraints}?
        |           ...
        +--rw tunnels
        |  +--rw tunnel* [name]
        |     +--rw name                               string
        |     +--rw alias?                             string
        |     +--rw identifier?                        uint32
        |     +--rw color?                             uint32
        |     +--rw description?                       string
        |     +--rw admin-state?                       identityref
        |     +--ro operational-state?                 identityref
        |     +--rw encoding?                          identityref
        |     +--rw switching-type?                    identityref
        |     +--rw source
        |     |     ...
        |     +--rw destination
        |     |     ...
        |     +--rw bidirectional?                     boolean
        |     +--rw controller
        |     |     ...
        |     +--rw reoptimize-timer?                  uint16
        |     +--rw association-objects
        |     |     ...
        |     +--rw protection
        |     |     ...
        |     +--rw restoration
        |     |     ...
        |     +--rw network-id?                   nw:network-id
        |     +--rw te-topology-identifier
        |     |     ...
        |     +--rw te-bandwidth
        |     |     ...
        |     +--rw link-protection?              identityref
        |     +--rw setup-priority?               uint8
        |     +--rw hold-priority?                uint8
        |     +--rw signaling-type?               identityref
        |     +--rw hierarchy
        |     |     ...
        |     +--rw primary-paths
        |     |     ...
        |     +--rw secondary-paths
        |     |     ...
        |     +--rw secondary-reverse-paths
        |     |     ...
        |     +---x tunnel-action
        |     |     ...
        |     +---x protection-external-commands
        |           ...
        +--ro lsps
           +--ro lsp* [tunnel-name lsp-id node]
              +--ro tunnel-name                       string
              +--ro lsp-id                            uint16
              +--ro node
              |       te-types:te-node-id
              +--ro source?
              |       te-types:te-node-id
              +--ro destination?
              |       te-types:te-node-id
              +--ro tunnel-id?                        uint16
              +--ro extended-tunnel-id?
              |       yang:dotted-quad
              +--ro operational-state?                identityref
              +--ro signaling-type?                   identityref
              +--ro origin-type?                      enumeration
              +--ro lsp-resource-status?              enumeration
              +--ro lockout-of-normal?                boolean
              +--ro freeze?                           boolean
              +--ro lsp-protection-role?              enumeration
              +--ro lsp-protection-state?             identityref
              +--ro protection-group-ingress-node-id?
              |       te-types:te-node-id
              +--ro protection-group-egress-node-id?
              |       te-types:te-node-id
              +--ro lsp-actual-route-information
                    ...

     rpcs:
       +---x tunnels-path-compute
       |  +---w input
       |  |  +---w path-compute-info
       |  +--ro output
       |     +--ro path-compute-result
       +---x tunnels-actions
          +---w input
          |  +---w tunnel-info
          |  |  +---w (filter-type)
          |  |        ...
          |  +---w action-info
          |     +---w action?       identityref
          |     +---w disruptive?   empty
          +--ro output
             +--ro action-result?   identityref
```

Figure 6: Tree diagram of depth-4 of TE Tunnel YANG data model

5.3. YANG Module

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

- ietf-te-types defined in [I-D.draft-ietf-teas-rfc8776-update]

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

- ietf-network and ietf-network-topology defined in [RFC8345]

This module references the following documents: [RFC4206], [RFC4427], [RFC4872], [RFC3209], [RFC6780], [RFC7471], [RFC9012], [RFC8570], [RFC8232], [RFC7271], [RFC8234], [RFC7308], and [ITU_G.808.1].

```
   <CODE BEGINS> file "ietf-te@2024-02-02.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-rfc8776-update: Common YANG Data Types
          for Traffic Engineering.";
     }
     import ietf-yang-types {
       prefix yang;
       reference
         "RFC6991: Common YANG Data Types.";
     }

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

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

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

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.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.ietf@gmail.com>

        Editor:   Igor Bryskin
                  <mailto:i_bryskin@yahoo.com>

        Editor:   Oscar Gonzalez de Dios
```

```
                  <mailto: oscar.gonzalezdedios@telefonica.com>";

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

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

        This version of this YANG module is part of RFC XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.";

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

     revision 2024-02-02 {
       description
         "Initial revision for the TE generic YANG module.";
       reference
         "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
          and Interfaces.";
     }

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

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

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

     /* This grouping is re-used in path-computation rpc */
     grouping path-forward-properties {
       description
         "The path preference.";
       leaf preference {
         type uint8 {
           range "1..255";
         }
         default "1";
         description
           "Specifies a preference for this path. The lower the number
```

```
            higher the preference.";
       }
       leaf co-routed {
         when "/te:te/te:tunnels/te:tunnel/te:bidirectional = 'true'" {
           description
             "Applicable to bidirectional tunnels only.";
         }
         type boolean;
         default "false";
         description
           "Indicates whether the reverse path must to be co-routed
            with the primary.";
       }
     }

     /* This grouping is re-used in path-computation rpc */
     grouping k-requested-paths {
       description
         "The k-shortest paths requests.";
       leaf k-requested-paths {
         type uint8;
         default "1";
         description
           "The number of k-shortest-paths requested from the path
            computation server and returned sorted by its optimization
            objective.";
       }
     }

     grouping path-state {
       description
         "TE per path state parameters.";
       uses path-computation-response;
       container lsp-provisioning-error-infos {
         config false;
         description
           "LSP provisioning error information.";
         list lsp-provisioning-error-info {
           description
             "List of LSP provisioning error info entries.";
           leaf error-reason {
             type identityref {
               base te-types:lsp-provisioning-error-reason;
             }
             description
               "LSP provision error type.";
           }
           leaf error-description {
             type string;
             description
               "The textual representation of the error occurred during
                path computation.";
           }
           leaf error-timestamp {
             type yang:date-and-time;
             description
               "Timestamp of when the reported error occurred.";
           }
           leaf error-node-id {
             type te-types:te-node-id;
             description
               "Node identifier of node where error occurred.";
           }
           leaf error-link-id {
             type te-types:te-tp-id;
             description
               "Link ID where the error occurred.";
           }
           leaf lsp-id {
             type uint16;
             description
               "The LSP-ID for which path computation was performed.";
           }
         }
       }
       container lsps {
         config false;
         description
           "The TE LSPs container.";
         list lsp {
           key "node lsp-id";
           description
             "List of LSPs associated with the tunnel.";
           leaf tunnel-name {
             type leafref {
               path "/te:te/te:lsps/te:lsp/te:tunnel-name";
             }
             description "TE tunnel name.";
           }
           leaf node {
             type leafref {
               path "/te:te/te:lsps/te:lsp[tunnel-name="
                  + "current()/../te:tunnel-name][lsp-id="
                  + "current()/../te:lsp-id]/te:node";
             }
             description "The node where the LSP state resides on.";
           }
           leaf lsp-id {
             type leafref {
               path "/te:te/te:lsps/te:lsp[tunnel-name="
                  + "current()/../tunnel-name]/te:lsp-id";
             }
             description "The TE LSP identifier.";
           }
         }
       }
     }

     /* This grouping is re-used in path-computation rpc */
     grouping path-computation-response {
       description
         "Attributes reported by path computation response.";
       container computed-paths-properties {
         config false;
         description
           "Computed path properties container.";
         list computed-path-properties {
           key "k-index";
           description
             "List of computed paths.";
           leaf k-index {
             type uint8;
             description
               "The k-th path returned from the computation server.
                A lower k value path is more optimal than higher k
                value path(s)";
           }
           uses te-types:generic-path-properties {
             augment "path-properties" {
               description
                 "additional path properties returned by path
                  computation.";
               uses te-types:te-bandwidth;
               leaf disjointness-type {
                 type te-types:te-path-disjointness;
                 config false;
                 description
                   "The type of resource disjointness.
                    When reported for a primary path, it represents the
                    minimum level of disjointness of all the secondary
                    paths. When reported for a secondary path, it
                    represents the disjointness of the secondary path.";
               }
             }
           }
         }
       }
       container computed-path-error-infos {
         config false;
         description
           "Path computation information container.";
         list computed-path-error-info {
           description
             "List of path computation info entries.";
           leaf error-description {
             type string;
             description
               "Textual representation of the error that occurred
                during path computation.";
           }
           leaf error-timestamp {
             type yang:date-and-time;
             description
               "Timestamp of last path computation attempt.";
           }
           leaf error-reason {
             type identityref {
               base te-types:path-computation-error-reason;
             }
             description
               "Reason for the path computation error.";
           }
         }
       }
     }

     grouping protection-restoration-properties {
       description
         "Protection and restoration parameters.";
       container protection {
         description
           "Protection parameters.";
         leaf protection-type {
           type identityref {
             base te-types:lsp-protection-type;
           }
           default "te-types:lsp-protection-unprotected";
           description
             "LSP protection type.";
         }
         leaf protection-reversion-disable {
           type boolean;
```

```
          default "false";
          description
            "Disable protection reversion to working path.";
        }
        leaf hold-off-time {
          type uint32;
          units "milli-seconds";
          description
            "The time between the declaration of an SF or SD condition
             and the initialization of the protection switching
             algorithm.";
          reference
            "RFC4427";
        }
        leaf wait-to-revert {
          type uint16;
          units "seconds";
          description
            "Time to wait before attempting LSP reversion.";
          reference
            "RFC4427";
        }
        leaf aps-signal-id {
          type uint8 {
            range "1..255";
          }
          default "1";
          description
            "The APS signal number used to reference the traffic of
             this tunnel. The default value for normal traffic is 1.
             The default value for extra-traffic is 255. If not
             specified, non-default values can be assigned by the
             server, if and only if, the server controls both
              endpoints.";
          reference
            "ITU_G.808.1";
        }
      }
      container restoration {
        description
          "Restoration parameters.";
        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;
           default "false";
           description
             "Disable restoration reversion to working path.";
         }
         leaf hold-off-time {
           type uint32;
           units "milli-seconds";
           description
             "The time between the declaration of an SF or SD condition
              and the initialization of the protection switching
              algorithm.";
           reference
             "RFC4427";
         }
         leaf wait-to-restore {
           type uint16;
           units "seconds";
           description
             "Time to wait before attempting LSP restoration.";
           reference
             "RFC4427";
         }
         leaf wait-to-revert {
           type uint16;
           units "seconds";
           description
             "Time to wait before attempting LSP reversion.";
           reference
             "RFC4427";
         }
       }
     }

     grouping tunnel-associations-properties {
       description
         "TE tunnel association grouping.";
       container association-objects {
         description
           "TE tunnel associations.";
         list association-object {
           key "association-key";
           unique "type id source/id source/type";
           description
             "List of association base objects.";
           reference
             "RFC4872";
           leaf association-key {
             type string;
             description
               "Association key used to identify a specific
                association in the list";
           }
           leaf type {
             type identityref {
               base te-types:association-type;
             }
             description
               "Association type.";
             reference
               "RFC4872";
           }
           leaf id {
             type uint16;
             description
               "Association identifier.";
             reference
               "RFC4872";
           }
           container source {
             uses te-types:te-generic-node-id;
             description
               "Association source.";
             reference
               "RFC4872";
           }
         }
         list association-object-extended {
           key "association-key";
           unique
             "type id source/id source/type global-source extended-id";
           description
             "List of extended association objects.";
           reference
             "RFC6780";
           leaf association-key {
             type string;
             description
```

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

     grouping tunnel-end-point {
       description
         "Common grouping used to specify the tunnel source and
         destination end-points.";
       leaf node-id {
         type nw:node-id;
         description
           "The TE tunnel end-point node identifier";
       }
       leaf te-node-id {
         type te-types:te-node-id;
         description
           "The TE tunnel end-point TE node identifier";
       }
       leaf tunnel-tp-id {
         when "../node-id or ../te-node-id" {
           description
             "The TE tunnel termination point identifier is local to
             a node";
         }
         type binary;
         description
           "The TE tunnel end-point TE tunnel termination point
           identifier";
       }
     }

     /* This grouping is re-used in path-computation rpc */
     grouping tunnel-common-attributes {
       description
         "Common grouping to define the TE tunnel parameters";
       container source {
         description
           "TE tunnel source end-point.";
         uses tunnel-end-point;
       }
       container destination {
         description
           "TE tunnel destination end-point.";
         uses tunnel-end-point;
       }
       leaf bidirectional {
         type boolean;
         default "false";
         description
           "Indicates a bidirectional tunnel";
       }
     }

     /* This grouping is re-used in path-computation rpc */
     grouping tunnel-hierarchy-properties {
       description
```

```
          "A grouping for TE tunnel hierarchy information.";
        container hierarchy {
          description
            "Container for TE hierarchy related information.";
          container dependency-tunnels {
            description
              "List of tunnels that this tunnel can be potentially
               dependent on.";
            list dependency-tunnel {
              key "name";
              description
                "A tunnel entry that this tunnel can potentially depend
                 on.";
              leaf name {
                type tunnel-ref;
                description
                  "Dependency tunnel name. The tunnel may not have been
                   instantiated yet.";
              }
              uses te-types:encoding-and-switching-type;
            }
          }
          container hierarchical-link {
            description
              "Identifies a hierarchical link (in client layer)
               that this tunnel is associated with. By default, the
               topology of the hierarchical link is the same topology of
               the tunnel;";
            reference
              "RFC4206";
            leaf enable {
              type boolean;
              default "false";
              description
                "Enables the hierarchical link properties supported by
                 this tunnel";
            }
            leaf local-node-id {
              type nw:node-id;
              description
                "The local node identifier.";
            }
            leaf local-te-node-id {
              type te-types:te-node-id;
              description
                "The local TE node identifier.";
            }
            leaf local-link-tp-id {
              type nt:tp-id;
              description
                "The local link termination point identifier.";
              reference
                "RFC8345";
            }
            leaf local-te-link-tp-id {
              type te-types:te-tp-id;
              description
                "The local TE link termination point identifier.";
            }
            leaf remote-node-id {
              type nw:node-id;
              description
                "The remote node identifier.";
            }
            leaf remote-link-tp-id {
              type nt:tp-id;
              description
                "The remote link termination point identifier.";
              reference
                "RFC8345";
            }
            leaf remote-te-link-tp-id {
              type te-types:te-tp-id;
              description
                "The remote TE link termination point identifier.";
            }
            leaf remote-te-node-id {
              type te-types:te-node-id;
              description
                "Remote TE node identifier.";
            }
            leaf link-id {
              type nt:link-id;
              config false;
              description
                "A network topology assigned identifier to the link";
              reference
                "RFC8345";
            }
            leaf network-id {
              type nw:network-id;
              description
                "The network topology identifier where the hierarchical
                link supported by this TE tunnel is instantiated.";
            }
            uses te-types:te-topology-identifier {
              description
                "The TE topology identifier where the hierarchical link
                 supported by this TE tunnel is instantiated.";
            }
          }
        }
      }

      grouping path-constraints-common {
        description
          "Global named path constraints configuration
           grouping.";
        uses te-types:common-path-constraints-attributes;
        uses te-types:generic-path-disjointness;
        uses te-types:path-constraints-route-objects;
        container path-in-segment {
          presence "The end-to-end tunnel starts in a previous domain;
                    this tunnel is a segment in the current domain.";
          description
            "If an end-to-end tunnel crosses multiple domains using
             the same technology, some additional constraints have to be
             taken in consideration in each domain.
             This TE tunnel segment is stitched to the upstream TE tunnel
             segment.";
          uses te-types:label-set-info;
        }
        container path-out-segment {
          presence
            "The end-to-end tunnel is not terminated in this domain;
             this tunnel is a segment in the current domain.";
          description
            "If an end-to-end tunnel crosses multiple domains using
             the same technology, some additional constraints have to be
             taken in consideration in each domain.
             This TE tunnel segment is stitched to the downstream TE
             tunnel segment.";
          uses te-types:label-set-info;
        }
      }

      /**
       * TE container
       */

      container te {
        description
          "TE global container.";
        leaf enable {
          type boolean;
          description
            "Enables the TE component features.";
        }

        /* TE Global Data */
        container globals {
          description
            "Globals TE system-wide configuration data container.";
          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.";
              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";
              }


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

        /* TE Tunnel Data */
        container tunnels {
          description
            "Tunnels TE configuration data container.";
          list tunnel {
            key "name";
            description
              "The list of TE tunnels.";
            leaf name {
              type string;
              description
                "TE tunnel name.";
            }
            leaf alias {
              type string;
              description
                "An alternate name of the TE tunnel that can be modified
                 anytime during its lifetime.";
            }
            leaf identifier {
              type uint32;
              description
                "TE tunnel Identifier.";
              reference
                "RFC3209";
            }
            leaf color {
              type uint32;
              description "The color associated with the TE tunnel.";
              reference "RFC9012";
            }
            leaf description {
              type string;
              default "None";
              description
                "Textual description for this TE tunnel.";
            }
            leaf admin-state {
              type identityref {
                base te-types:tunnel-admin-state-type;
              }
              default "te-types:tunnel-admin-state-up";
              description
                "TE tunnel administrative state.";
            }
            leaf operational-state {
              type identityref {
                base te-types:tunnel-state-type;
              }
              config false;
              description
                "TE tunnel operational state.";
            }
            uses te-types:encoding-and-switching-type;
            uses tunnel-common-attributes;
            container controller {
              description
                "Contains tunnel data relevant to external controller(s).
                 This target node may be augmented by external module(s),
                 for example, to add data for PCEP initiated and/or
                 delegated tunnels.";
              leaf protocol-origin {
                type identityref {
                  base te-types:protocol-origin-type;
                }
                description
                  "The protocol origin for instantiating the tunnel.";
              }
              leaf controller-entity-id {
                type string;
                description
                  "An identifier unique within the scope of visibility
                   that associated with the entity that controls the
                   tunnel.";
                reference "RFC8232";
              }
            }
            leaf reoptimize-timer {
              type uint16;
              units "seconds";
              description
                "Frequency of reoptimization of a traffic engineered
                 LSP.";
            }
            uses tunnel-associations-properties;
            uses protection-restoration-properties;
            uses te-types:tunnel-constraints;
            uses tunnel-hierarchy-properties;
            container primary-paths {
              description
                "The set of primary paths.";
              reference "RFC4872";
              list primary-path {
                key "name";
                description
                  "List of primary paths for this tunnel.";
                leaf active {
                  type boolean;
                  config false;
                  description
                    "Indicates an active path that
                     has been selected from the primary paths list.";
                }
                uses path-common-properties;
                uses path-forward-properties;
                uses k-requested-paths;
                uses path-compute-info;
                uses path-state;
                container primary-reverse-path {
                  when "../../../te:bidirectional = 'true'";
                  description
```

```
                   "The reverse primary path properties.";
                 uses path-common-properties;
                 uses path-compute-info;
                 uses path-state;
                 container candidate-secondary-reverse-paths {
                   description
                     "The set of referenced candidate reverse secondary
                      paths from the full set of secondary reverse paths
                      which may be used for this primary path.";
                   list candidate-secondary-reverse-path {
                     key "secondary-reverse-path";
                     ordered-by user;
                     description
                       "List of candidate secondary reverse path(s)";
                     leaf secondary-reverse-path {
                       type leafref {
                         path "../../../../../../"
                            + "te:secondary-reverse-paths/"
                            + "te:secondary-reverse-path/te:name";
                       }
                       description
                         "A reference to the secondary reverse path that
                          may be utilized when the containing primary
                          reverse path is in use.";
                     }
                     leaf active {
                       type boolean;
                       config false;
                       description
                         "Indicates an active path that has been
                          selected from the secondary reverse paths
                          list.";
                     }
                   }
                 }
               }
               container candidate-secondary-paths {
                 description
                   "The set of candidate secondary paths which may be
                    used for this primary path. When secondary paths are
                    specified in the list the path of the secondary LSP
                    in use must be restricted to those paths
                    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 may be
                    utilised when the current primary path is in use.";
                 list candidate-secondary-path {
                   key "secondary-path";
                   ordered-by user;
                   description
                     "List of candidate secondary paths for this
                      tunnel.";
                   leaf secondary-path {
                     type leafref {
                       path "../../../../../te:secondary-paths/"
                          + "te:secondary-path/te:name";
                     }
                     description
                       "A reference to the secondary path that may be
                        utilised when the containing primary path is
                        in use.";
                   }
                   leaf active {
                     type boolean;
                     config false;
                     description
                       "Indicates an active path that has been selected
                        from the candidate secondary paths.";
                   }
                 }
               }
             }
           }
           container secondary-paths {
             description
               "The set of secondary paths.";
             reference "RFC4872";
             list secondary-path {
               key "name";
               description
                 "List of secondary paths for this tunnel.";
               uses path-common-properties;
               leaf preference {
                 type uint8 {
                   range "1..255";
                 }
                 default "1";
                 description
                   "Specifies a preference for this path. The lower the
                    number higher the preference.";
               }
               leaf secondary-reverse-path {
                 type leafref {
                   path "../../../"
                      + "te:secondary-reverse-paths/"
                      + "te:secondary-reverse-path/te:name";
                 }
                 description
                   "A reference to the reverse secondary path when
                    co-routed with the secondary path.";
               }
               uses path-compute-info;
               uses protection-restoration-properties;
               uses path-state;
             }
           }
           container secondary-reverse-paths {
             description
               "The set of secondary reverse paths.";
             list secondary-reverse-path {
               key "name";
               description
                 "List of secondary paths for this tunnel.";
               uses path-common-properties;
               leaf preference {
                 type uint8 {
                   range "1..255";
                 }
                 default "1";
                 description
                   "Specifies a preference for this path. The lower the
                    number higher the preference. Paths that have the
                    same preference will be activated together.";
               }
               uses path-compute-info;
               uses protection-restoration-properties;
               uses path-state;
             }
           }
           action tunnel-action {
             description
               "Action commands to manipulate the TE tunnel state.";
             reference
               "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
                Section 2.5";
             input {
               leaf action-type {
                 type identityref {
                   base te-types:tunnel-action-type;
                 }
                 description
```

```
                   "The action to be invoked on the TE tunnel.";
               }
             }
             output {
               leaf action-result {
                 type identityref {
                   base te-types:te-action-result;
                 }
                 description
                   "The result of the tunnel action operation.";
               }
             }
           }
           action protection-external-commands {
             description
               "Actions to manipulate the protection external
                commands of the TE tunnel.";
             reference
               "RFC 4427: Recovery (Protection and Restoration)
                Terminology for Generalized Multi-Protocol Label
                Switching (GMPLS)";
             input {
               leaf protection-external-command {
                 type identityref {
                   base te-types:protection-external-commands;
                 }
                 description
                   "Protection external command.";
               }
               leaf protection-group-ingress-node {
                 type boolean;
                 default "true";
                 description
                   "When 'true', indicates that the action is
                    applied on ingress node.
                    By default, the action applies to the ingress node
                    only.";
               }
               leaf protection-group-egress-node {
                 type boolean;
                 default "false";
                 description
                   "When set to 'true', indicates that the action is
                    applied on egress node.
                    By default, the action applies to the ingress node
                    only.";
               }
               leaf path-name {
                 type string;
                 description
                   "The name of the path that the external command
                   applies to.";
               }
               leaf path-type {
                 type te-types:path-type;
                 description
                   "The type of the path that the external command
                   applies to.";
               }
               leaf traffic-type {
                 type enumeration {
                   enum normal-traffic {
                     description
                       "The manual-switch or forced-switch command
                        applies to the normal traffic (this Tunnel).";
                   }
                   enum null-traffic {
                     description
                       "The manual-switch or forced-switch command
                        applies to the null traffic.";
                   }
                   enum extra-traffic {
                     description
                       "The manual-switch or forced-switch command
                        applies to the extra traffic (the extra-traffic
                        Tunnel sharing protection bandwidth with this
                        Tunnel).";
                   }
                 }
                 description
                   "Indicates whether the manual-switch or forced-switch
                    commands applies to the normal traffic, the null
                    traffic or the extra-traffic.";
                 reference
                   "RFC4427";
               }
               leaf extra-traffic-tunnel-ref {
                 type tunnel-ref;
                 description
                   "In case there are multiple extra-traffic tunnels
                    sharing protection bandwidth with this Tunnel
                    (m:n protection), represents which extra-traffic
                    Tunnel the manual-switch or forced-switch to
                    extra-traffic command applies to.";
               }
             }
```

```
          }
        }
      }

      /* TE LSPs Data */
      container lsps {
        config false;
        description
          "TE LSPs state container.";
        list lsp {
          key "tunnel-name lsp-id node";
          unique "source destination tunnel-id lsp-id "
            + "extended-tunnel-id";
          description
            "List of LSPs associated with the tunnel.";
          leaf tunnel-name {
            type string;
            description "The TE tunnel name.";
          }
          leaf lsp-id {
            type uint16;
            description
              "Identifier used in the SENDER_TEMPLATE and the
               FILTER_SPEC that can be changed to allow a sender to
               share resources with itself.";
            reference
              "RFC3209";
          }
          leaf node {
            type te-types:te-node-id;
            description
              "The node where the TE LSP state resides on.";
          }
          leaf source {
            type te-types:te-node-id;
            description
              "Tunnel sender address extracted from
               SENDER_TEMPLATE  object.";
            reference
              "RFC3209";
          }
          leaf destination {
            type te-types:te-node-id;
            description
              "The tunnel endpoint address.";
            reference
              "RFC3209";
          }
```

```
           leaf tunnel-id {
             type uint16;
             description
               "The tunnel identifier that remains
                constant over the life of the tunnel.";
             reference
               "RFC3209";
           }
           leaf extended-tunnel-id {
             type yang:dotted-quad;
             description
               "The LSP Extended Tunnel ID.";
             reference
               "RFC3209";
           }
           leaf operational-state {
             type identityref {
               base te-types:lsp-state-type;
             }
             description
               "The LSP operational state.";
           }
           leaf signaling-type {
             type identityref {
               base te-types:path-signaling-type;
             }
             description
               "The signaling protocol used to set up this LSP.";
           }
           leaf origin-type {
             type enumeration {
               enum ingress {
                 description
                   "Origin ingress.";
               }
               enum egress {
                 description
                   "Origin egress.";
               }
               enum transit {
                 description
                   "Origin transit.";
               }
             }
             description
               "The origin of the LSP relative to the location of the
                local switch in the path.";
           }
           leaf lsp-resource-status {
             type enumeration {
               enum primary {
                 description
                   "A primary LSP is a fully established LSP for which
                    the resource allocation has been committed at the
                    data plane.";
               }
               enum secondary {
                 description
                   "A secondary LSP is an LSP that has been provisioned
                    in the control plane only; e.g. resource allocation
                    has not been committed at the data plane.";
               }
             }
             description
               "LSP resource allocation state.";
             reference
               "RFC4872, section 4.2.1";
           }
           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
               "RFC4427";
           }
           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 applies to
                all the Tunnels which are sharing the protection
                resources with this Tunnel.";
             reference
               "RFC4427";
           }
           leaf lsp-protection-role {
             type enumeration {
               enum working {
                 description
                   "A working LSP must be a primary LSP whilst a
                    protecting LSP can be either a primary or a
                    secondary LSP. Also, known as protected LSPs when
                    working LSPs are associated with protecting LSPs.";
               }
               enum protecting {
                 description
                   "A secondary LSP is an LSP that has been provisioned
                    in the control plane only; e.g. resource allocation
                    has not been committed at the data plane.";
               }
             }
             description
               "LSP role type.";
             reference
               "RFC4872, section 4.2.1";
           }
           leaf lsp-protection-state {
             type identityref {
               base te-types:lsp-protection-state;
             }
             config false;
             description
               "The reported protection state controlling which
                tunnel 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 external
                command (LoP, SF, MS) applied to it or a WTR timer
                running on it. If the external command is not applied to
                the ingress node or the WTR timer is not running on it,
                this attribute is not specified. A value 0.0.0.0 is used
                when the te-node-id of the protection group ingress node
                is unknown (e.g., because the ingress node is outside
                the scope of control of the server)";
           }
           leaf protection-group-egress-node-id {
             type te-types:te-node-id;
             description
               "Indicates the te-node-id of the protection group egress
                node when the APS state represents an external command
                (LoP, SF, MS) applied to it or a WTR timer running on
                it. If the external command is not applied to the
                ingress node or the WTR timer is not running on it, this
                attribute is not specified. A value 0.0.0.0 is used when
                the te-node-id of the protection group ingress node is
                unknown (e.g., because the ingress node is outside the
                scope of control of the server)";
           }
           container lsp-actual-route-information {
             description
               "RSVP recorded route object information.";
             list lsp-actual-route-information {
               when "../../origin-type = 'ingress'" {
                 description
                   "Applicable on ingress LSPs only.";
               }
               key "index";
               description
                 "Record route list entry.";
               uses te-types:record-route-state;
             }
           }
         }
       }
     }

     /* TE Tunnel RPCs/execution Data */

     rpc tunnels-path-compute {
       description
         "This RPC is a generic API whose
          input and output parameters are expected to be provided by
          augments to this module.";
       reference
         "RFC 4655: A Path Computation Element (PCE)-Based
          Architecture.";
       input {
         container path-compute-info {
           /*
            * An external path compute module may augment this
            * target.
            */
           description
             "RPC input information.";
         }
       }
       output {
         container path-compute-result {
           /*
            * An external path compute module may augment this
            * target.
            */
           description
             "RPC output information.";
         }
       }
     }

     rpc tunnels-actions {
       description
         "RPC that manipulates the state of a TE tunnel.";
       reference
         "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
          Section 2.5";
       input {
         container tunnel-info {
           description
             "TE tunnel information.";
           choice filter-type {
             mandatory true;
             description
               "Filter choice.";
             case all-tunnels {
               leaf all {
                 type empty;
                 mandatory true;
                 description
                   "When present, applies the action on all TE
                    tunnels.";
               }
             }
             case one-tunnel {
               leaf tunnel {
                 type tunnel-ref;
                 description
                   "Apply action on the specific TE tunnel.";
               }
             }
           }
         }
         container action-info {
           description
             "TE tunnel action information.";
           leaf action {
             type identityref {
               base te-types:tunnel-action-type;
             }
             description
               "The action type.";
           }
           leaf disruptive {
             when "derived-from-or-self(../action, "
                + "'te-types:tunnel-action-reoptimize')";
             type empty;
             description
               "When present, specifies whether or not the
                reoptimization
                action is allowed to be disruptive.";
           }
         }
       }
       output {
         leaf action-result {
           type identityref {
             base te-types:te-action-result;
           }
           description
             "The result of the tunnel action operation.";
         }
       }
     }
   }
   <CODE ENDS>
```

Figure 7: TE Tunnel data model YANG module

6. TE Device YANG Model

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

6.1. Module Structure

6.1.1. TE Interfaces

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

- Maximum reservable bandwidth, bandwidth constraints (BC)

- Flooding parameters

  - Flooding intervals and threshold values

- Interface attributes

  - (Extended) administrative groups

  - SRLG values

  - TE metric value

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

The derived state associated with interfaces is grouped under the interface "state" sub-container as shown in Figure 8. This covers state data such as:

* Bandwidth information: maximum bandwidth, available bandwidth at different priorities and for each class-type (CT)

* List of admitted LSPs

  - Name, bandwidth value and pool, time, priority

* Statistics: state counters, flooding counters, admission counters (accepted/rejected), preemption counters

* Adjacency information

  - Neighbor address

  - Metric value

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

Figure 8: TE interface state YANG subtree

6.2. Tree Diagram

Figure 9 shows the tree diagram of the device TE YANG model defined in modules 'ietf-te-device.yang'.

```
   module: ietf-te-device

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

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

Figure 9: TE Tunnel device model YANG tree diagram

6.3. YANG Module

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

- ietf-interfaces defined in [RFC8343]

- ietf-routing-types defined in [RFC8294]

- ietf-te-types defined in [I-D.draft-ietf-teas-rfc8776-update]

- ietf-te defined in this document

```
   <CODE BEGINS> file "ietf-te-device@2024-02-02.yang"
   module ietf-te-device {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

     /* Replace with IANA when assigned */

     prefix te-dev;

     /* Import TE module */

     import ietf-te {
       prefix te;
       reference
         "RFCXXXX: A YANG Data Model for Traffic Engineering
          Tunnels and Interfaces";
     }

     /* Import TE types */

     import ietf-te-types {
       prefix te-types;
       reference
         "draft-ietf-teas-rfc8776-update: Common YANG Data Types
              for Traffic Engineering.";
     }
     import ietf-interfaces {
       prefix if;
       reference
         "RFC8343: A YANG Data Model for Interface Management";
     }
     import ietf-routing-types {
       prefix rt-types;
       reference
         "RFC8294: Common YANG Data Types for the Routing Area";
     }

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

```
        WG List:  <mailto:teas@ietf.org>

        Editor:   Tarek Saad
                  <mailto:tsaad.net@gmail.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.ietf@gmail.com>

        Editor:   Igor Bryskin
                  <mailto:i_bryskin@yahoo.com>

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

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

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

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

        This version of this YANG module is part of RFC XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.";

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

     revision 2024-02-02 {
       description
         "Initial revision for the TE device YANG module.";
       reference
         "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
          and Interfaces";
     }

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

     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 device augmentations
      */
     augment "/te:te" {
       description
         "TE global container.";
       /* TE Interface Configuration Data */
       container interfaces {
         description
           "Configuration data model for TE interfaces.";
         uses te-igp-flooding-bandwidth-config;
         list interface {
           key "name";
           description
             "The list of interfaces enabled for TE.";
           leaf name {
             type if:interface-ref;
             description
               "The reference to interface enabled for TE.";
           }
           /* TE interface parameters */
           leaf te-metric {
             type te-types:te-metric;
             description
               "TE interface metric.";
           }
           choice admin-group-type {
             description
               "TE interface administrative groups
                representation type.";
             case value-admin-groups {
               choice value-admin-group-type {
                 description
                   "choice of admin-groups.";
                 case admin-groups {
                   description
                     "Administrative group/Resource
                      class/Color.";
                   leaf admin-group {
                     type te-types:admin-group;
                     description
                       "TE interface administrative group.";
                   }
                 }
                 case extended-admin-groups {
                   if-feature "te-types:extended-admin-groups";
                   description
                     "Extended administrative group/Resource
                      class/Color.";
                   leaf extended-admin-group {
                     type te-types:extended-admin-group;
                     description
                       "TE interface extended administrative group.";
                   }
                 }
               }
             }
             case named-admin-groups {
               list named-admin-groups {
                 if-feature "te-types:extended-admin-groups";
                 if-feature "te-types:named-extended-admin-groups";
                 key "named-admin-group";
                 description
                   "A list of named admin-group entries.";
                 leaf named-admin-group {
                   type leafref {
                     path "../../../../te:globals/"
                        + "te:named-admin-groups/te:named-admin-group/"
                        + "te:name";
                   }
                   description
                     "A named admin-group entry.";
                 }
               }
             }
           }
           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.";
                 }
               }
             }
           }
           uses te-igp-flooding-bandwidth-config;
           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.";
             }
           }
           container te-advertisements-state {
             config false;
             description
               "TE interface advertisements state container.";
             leaf flood-interval {
               type uint32;
               description
                 "The periodic flooding interval.";
             }
             leaf last-flooded-time {
               type uint32;
               units "seconds";
               description
                 "Time elapsed since last flooding in seconds.";
             }
             leaf next-flooded-time {
               type uint32;
               units "seconds";
               description
                 "Time remained for next flooding in seconds.";
             }
             leaf last-flooded-trigger {
               type enumeration {
                 enum link-up {
                   description
                     "Link-up flooding trigger.";
                 }
                 enum link-down {
                   description
                     "Link-down flooding trigger.";
                 }
                 enum threshold-up {
                   description
                     "Bandwidth reservation up threshold.";
                 }
                 enum threshold-down {
                   description
                     "Bandwidth reservation down threshold.";
                 }
                 enum bandwidth-change {
                   description
                     "Bandwidth capacity change.";
                 }
                 enum user-initiated {
                   description
```

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

     /* TE globals device augmentation */

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

     /* TE tunnels device configuration augmentation */

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

     /* TE LSPs device state augmentation */

     augment "/te:te/te:lsps/te:lsp" {
       description
         "TE LSP device dependent augmentation.";
       container lsp-timers {
         when "../te:origin-type = 'ingress'" {
           description
             "Applicable to ingress LSPs only.";
         }
         description
           "Ingress LSP timers.";
         leaf uptime {
           type uint32;
           units "seconds";
           description
             "The LSP uptime.";
         }
         leaf time-to-install {
           type uint32;
           units "seconds";
           description
             "The time remaining for a new LSP to be instantiated
              in forwarding to carry traffic.";
         }
         leaf time-to-destroy {
           type uint32;
           units "seconds";
           description
             "The time remaining for a existing LSP to be deleted
              from forwarding.";
         }
       }
       container downstream-info {
         when "../te:origin-type != 'egress'" {
           description
             "Downstream information of the LSP.";
         }
         description
           "downstream information.";
         leaf nhop {
           type te-types:te-tp-id;
           description
             "downstream next-hop address.";
         }
         leaf outgoing-interface {
           type if:interface-ref;
           description
             "downstream interface.";
         }
         container neighbor {
           uses te-types:te-generic-node-id;
           description
             "downstream neighbor address.";
         }
         leaf label {
           type rt-types:generalized-label;
           description
             "downstream label.";
         }
       }
       container upstream-info {
         when "../te:origin-type != ’ingress’" {
           description
             "Upstream information of the LSP.";
         }
         description
           "upstream information.";
         leaf phop {
           type te-types:te-tp-id;
           description
             "upstream next-hop or previous-hop address.";
         }
         container neighbor {
           uses te-types:te-generic-node-id;
           description
             "upstream neighbor address.";
         }
         leaf label {
           type rt-types:generalized-label;
           description
             "upstream label.";
         }
       }
     }

     /* TE interfaces RPCs/execution Data */

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

Figure 10: TE device data model YANG module

7. Notifications

Notifications are a key component of any topology data model.

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

- Subscribe to notifications on a per-client basis.

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

- Specify either periodic or on-demand notifications.

8. IANA Considerations

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

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

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

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

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

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

9. Security Considerations

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

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

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

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

"/te/tunnels": This list specifies the configuration and state of TE Tunnels present on the device or controller. Unauthorized access to this list could cause the device to ignore packets it should receive and process. An attacker may also use state to derive information about the network topology, and subsequently orchestrate further attacks.

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

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

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

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

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

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

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

10. Acknowledgement

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

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

11. Contributors

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

12. Appendix A: Data Tree Examples

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

For the example we will use a 4 node MPLS network were RSVP-TE MPLS Tunnels can be setup. The loopbacks of each router are shown. The network in Figure 11 will be used in the examples described in the following sections.

```
    192.0.2.1        192.0.2.2      192.0.2.4
    +-------+        +-------+      +-------+
    |       |        |       |      |       |
    |   A   +--------+   B   +------+   D   |
    +---+---+        +-------+      +---+---+
        |                               |
        |            +-------+          |
        |            |       |          |
        +------------+   C   +----------+
                     |       |
                     +-------+
                     192.0.2.3
```

Figure 11: TE network used in data tree examples

12.1. Basic Tunnel Setup

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

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

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

{
  "ietf-te:tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_2",
      "encoding": "te-types:lsp-encoding-packet",
      "admin-state": "te-types:tunnel-state-up",
      "source": "192.0.2.1",
      "destination": "192.0.2.4",
      "bidirectional": "false",
      "signaling-type": "te-types:path-setup-rsvp"
    }
  ]
}
```

12.2. Global Named Path Constraints

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

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


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

12.3. Tunnel with Global Path Constraint

In this example, the previously created 'named path constraint' is applied to the TE Tunnel created in Section 12.1.

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

{
  "ietf-te:ietf-tunnel": [
    {
      "name": "Example_LSP_Tunnel_A_4_1",
      "encoding": "te-types:lsp-encoding-packet",
      "description": "Simple_LSP_with_named_path",
      "admin-state": "te-types:tunnel-state-up",
      "source": "192.0.2.1",
      "destination": "192.0.2.4",
      "signaling-type": "path-setup-rsvp",
      "primary-paths": [
        {
          "primary-path": {
            "name": "Simple_LSP_1",
            "use-path-computation": "true",
            "named-path-constraint": "max-hop-3"
          }
        }
      ]
    }
  ]
}
```

12.4. Tunnel with Per-tunnel Path Constraint

In this example, the a per tunnel path constraint is explicitly indicated under the TE Tunnel created in Section 12.1 to constrain the computed path for the tunnel.

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

   {
     "ietf-te:tunnel": [
       {
         "name": "Example_LSP_Tunnel_A_4_2",
         "encoding": "te-types:lsp-encoding-packet",
         "admin-state": "te-types:tunnel-state-up",
         "source": "192.0.2.1",
         "destination": "192.0.2.4",
         "signaling-type": "te-types:path-setup-rsvp",
         "primary-paths": {
           "primary-path": [
             {
               "name": "path1",
               "path-metric-bounds": {
                 "path-metric-bound": [
                   {
                     "metric-type": "te-types:path-metric-hop",
                     "upper-bound": "3"
                   }
                 ]
               }
             }
           ]
         }
       }
     ]
   }
```

12.5. Tunnel State

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

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

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

```
   {
     "ietf-te:primary-paths": {
       "primary-path": [
         {
           "name": "path1",
           "path-computation-method": "te-types:path-locally-computed",
           "computed-paths-properties": {
             "computed-path-properties": [
               {
                 "k-index": "1",
                 "path-properties": {
                   "path-route-objects": {
                     "path-route-object": [
                       {
                         "index": "1",
                         "numbered-node-hop": {
                           "node-id": "192.0.2.2"
                         }
                       },
                       {
                         "index": "2",
                         "numbered-node-hop": {
                           "node-id": "192.0.2.4"
                         }
                       }
                     ]
                   }
                 }
               }
             ]
           },
           "lsps": {
             "lsp": [
               {
                 "tunnel-name": "Example_LSP_Tunnel_A_4_1",
                 "node": "192.0.2.1 ",
                 "lsp-id": "25356"
               }
             ]
           }
         }
       ]
     }
   }
```

12.6. Example TE Tunnel with Primary and Secondary Paths

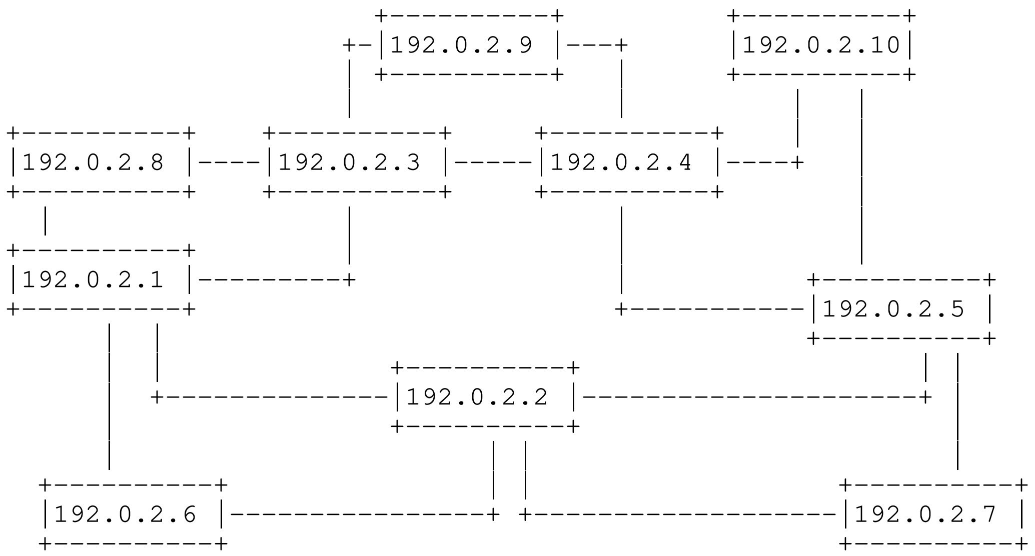

Figure 12: TE network used in data tree examples

Below is the state retrieved for a TE tunnel from source 192.0.2.1 to 192.0.2.5 with primary, secondary, reverse, and secondary reverse paths as shown in Figure 12.

```
{
  "ietf-te:te": {
    "tunnels": {
      "tunnel": [
        {
          "name": "example-1",
          "description": "Example in slide 1",
          "source": "192.0.2.1",
          "destination": "192.0.2.5",
          "bidirectional": false,
          "primary-paths": {
            "primary-path": [
              {
                "name": "primary-1 (fwd)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                },
                "primary-reverse-path": {
                  "name": "primary-2 (rev)",
                  "explicit-route-objects": {
                    "route-object-include-exclude": [
                      {
                        "index": 1,
                        "explicit-route-usage" : "route-include-object",
                        "numbered-node-hop": {
                          "node-id": "192.0.2.3",
                          "hop-type": "loose"
                        }
                      }
                    ]
                  },
                  "candidate-secondary-reverse-paths": {
                    "candidate-secondary-reverse-path": [
                      "secondary-3 (rev)",
                      "secondary-4 (rev)",
                      "secondary-5 (rev)"
                    ]
                  }
                },
                "candidate-secondary-paths": {
                  "candidate-secondary-path": [
                    "secondary-1 (fwd)",
                    "secondary-2 (fwd)"
                  ]
                }
              }
            ]
          },
          "secondary-paths": {
            "secondary-path": [
              {
                "name": "secondary-1 (fwd)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.1"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              },
              {
                "name": "secondary-2 (fwd)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.5",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              }
            ]
          },
          "secondary-reverse-paths": {
            "secondary-reverse-path": [
              {
                "name": "secondary-3 (rev)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.5"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.4",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              },
              {
                "name": "secondary-4 (rev)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.4"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.3",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              },
              {
                "name": "secondary-5 (rev)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.3"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.1",
                        "hop-type":"loose"
                      }
                    }
                  ]
                }
              }
            ]
          }
        },
        {
          "name": "example-3",
          "description": "Example in slide 3",
          "source": "192.0.2.1",
          "destination": "192.0.2.5",
          "bidirectional": true,
          "primary-paths": {
            "primary-path": [
              {
                "name": "primary-1 (bidir)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                },
                "candidate-secondary-paths": {
                  "candidate-secondary-path": [
                    "secondary-1 (bidir)",
                    "secondary-2 (bidir)"
                  ]
                }
              }
            ]
          },
          "secondary-paths": {
            "secondary-path": [
              {
                "name": "secondary-1 (bidir)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.1"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              },
              {
                "name": "secondary-2 (bidir)",
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.5",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              }
            ]
          }
        },
        {
          "name": "example-4",
          "description": "Example in slide 4",
          "source": "192.0.2.1",
          "destination": "192.0.2.5",
          "bidirectional": false,
          "primary-paths": {
            "primary-path": [
              {
                "name": "primary-1 (fwd)",
                "co-routed": [null],
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                },
                "primary-reverse-path": {
                  "name": "primary-2 (rev)",
                  "candidate-secondary-reverse-paths": {
                    "candidate-secondary-reverse-path": [
                      "secondary-3 (rev)",
                      "secondary-4 (rev)"
                    ]
                  }
                },
                "candidate-secondary-paths": {
                  "candidate-secondary-path": [
                    "secondary-1 (fwd)",
                    "secondary-2 (fwd)"
                  ]
                }
              }
            ]
          },
          "secondary-paths": {
            "secondary-path": [
              {
                "name": "secondary-1 (fwd)",
                "co-routed": [null],
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.1"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              },
              {
                "name": "secondary-2 (fwd)",
                "co-routed": [null],
                "explicit-route-objects": {
                  "route-object-include-exclude": [
                    {
                      "index": 1,
                      "explicit-route-usage" : "route-include-object",
                      "numbered-node-hop": {
                        "node-id": "192.0.2.2"
                      }
                    },
                    {
                      "index": 2,
                      "numbered-node-hop": {
                        "node-id": "192.0.2.5",
                        "hop-type": "loose"
                      }
                    }
                  ]
                }
              }
            ]
          },
          "secondary-reverse-paths": {
            "secondary-reverse-path": [
              {
                "name": "secondary-3 (rev)"
              },
              {
                "name": "secondary-4 (rev)"
              }
            ]
          }
        }
      ]
    }
  }
}
```

13. Appendix B: Full Model Tree Diagram

Figure 13 shows the full tree diagram of the TE YANG model defined in module 'ietf-te.yang'.

```
   module: ietf-te
     +--rw te
        +--rw enable?    boolean
        +--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 value?   te-types:srlg
        |  |     +--rw cost?    uint32
        |  +--rw named-path-constraints
        |     +--rw named-path-constraint* [name]
        |             {te-types:named-path-constraints}?
        |        +--rw name                       string
        |        +---u path-constraints-common
        +--rw tunnels
        |  +--rw tunnel* [name]
        |     +--rw name                                    string
        |     +--rw alias?                                  string
        |     +--rw identifier?                             uint32
        |     +--rw color?                                  uint32
        |     +--rw description?                            string
        |     +--rw admin-state?                            identityref
        |     +--ro operational-state?                      identityref
        |     +---u te-types:encoding-and-switching-type
        |     +---u tunnel-common-attributes
        |     +--rw controller
        |     |  +--rw protocol-origin?        identityref
        |     |  +--rw controller-entity-id?   string
        |     +--rw reoptimize-timer?                       uint16
        |     +---u tunnel-associations-properties
        |     +---u protection-restoration-properties
        |     +---u te-types:tunnel-constraints
        |     +---u tunnel-hierarchy-properties
        |     +--rw primary-paths
        |     |  +--rw primary-path* [name]
        |     |     +--ro active?                      boolean
        |     |     +---u path-common-properties
        |     |     +---u path-forward-properties
        |     |     +---u k-requested-paths
        |     |     +---u path-compute-info
        |     |     +---u path-state
        |     |     +--rw primary-reverse-path
        |     |     |  +---u path-common-properties
        |     |     |  +---u path-compute-info
        |     |     |  +---u path-state
        |     |     |  +--rw candidate-secondary-reverse-paths
        |     |     |     +--rw candidate-secondary-reverse-path*
        |     |     |             [secondary-reverse-path]
        |     |     |        +--rw secondary-reverse-path    leafref
        |     |     |        +--ro active?                   boolean
        |     |     +--rw candidate-secondary-paths
        |     |        +--rw candidate-secondary-path* [secondary-path]
        |     |           +--rw secondary-path    leafref
        |     |           +--ro active?           boolean
        |     +--rw secondary-paths
        |     |  +--rw secondary-path* [name]
        |     |     +---u path-common-properties
        |     |     +--rw preference?                          uint8
        |     |     +--rw secondary-reverse-path?              leafref
        |     |     +---u path-compute-info
        |     |     +---u protection-restoration-properties
        |     |     +---u path-state
        |     +--rw secondary-reverse-paths
        |     |  +--rw secondary-reverse-path* [name]
        |     |     +---u path-common-properties
        |     |     +--rw preference?                          uint8
        |     |     +---u path-compute-info
        |     |     +---u protection-restoration-properties
        |     |     +---u path-state
        |     +---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?   boolean
        |           +---w protection-group-egress-node?    boolean
        |           +---w path-name?                       string
        |           +---w path-type?
        |           |       te-types:path-type
        |           +---w traffic-type?                    enumeration
        |           +---w extra-traffic-tunnel-ref?        tunnel-ref
        +--ro lsps
           +--ro lsp* [tunnel-name lsp-id node]
              +--ro tunnel-name                         string
              +--ro lsp-id                              uint16
              +--ro node
              |       te-types:te-node-id
              +--ro source?
              |       te-types:te-node-id
              +--ro destination?
              |       te-types:te-node-id
              +--ro tunnel-id?                          uint16
              +--ro extended-tunnel-id?                 yang:dotted-quad
              +--ro operational-state?                  identityref
              +--ro signaling-type?                     identityref
              +--ro origin-type?                        enumeration
              +--ro lsp-resource-status?                enumeration
              +--ro lockout-of-normal?                  boolean
              +--ro freeze?                             boolean
              +--ro lsp-protection-role?                enumeration
              +--ro lsp-protection-state?               identityref
              +--ro protection-group-ingress-node-id?
              |       te-types:te-node-id
              +--ro protection-group-egress-node-id?
              |       te-types:te-node-id
              +--ro lsp-actual-route-information
                 +--ro lsp-actual-route-information* [index]
                    +---u te-types:record-route-state

     rpcs:
       +---x tunnels-path-compute
       |  +---w input
       |  |  +---w path-compute-info
       |  +--ro output
       |     +--ro path-compute-result
       +---x tunnels-actions
          +---w input
          |  +---w tunnel-info
          |  |  +---w (filter-type)
          |  |     +--:(all-tunnels)
          |  |     |  +---w all       empty
          |  |     +--:(one-tunnel)
          |  |        +---w tunnel?   tunnel-ref
          |  +---w action-info
          |     +---w action?       identityref
          |     +---w disruptive?   empty
          +--ro output
             +--ro action-result?   identityref

     grouping path-common-properties:
       +-- name?                      string
       +-- path-computation-method?   identityref
       +-- path-computation-server
       |  +---u te-types:te-generic-node-id
       +-- compute-only?              empty
       +-- use-path-computation?      boolean
       +-- lockdown?                  empty
       +--ro path-scope?                identityref
     grouping path-compute-info:
       +---u tunnel-associations-properties
       +---u te-types:generic-path-optimization
       +-- named-path-constraint?                leafref
       |       {te-types:named-path-constraints}?
       +---u path-constraints-common
     grouping path-forward-properties:
       +-- preference?   uint8
       +-- co-routed?    boolean
     grouping k-requested-paths:
       +-- k-requested-paths?   uint8
     grouping path-state:
       +---u path-computation-response
       +--ro lsp-provisioning-error-infos
       |  +--ro lsp-provisioning-error-info* []
       |     +--ro error-reason?        identityref
       |     +--ro error-description?   string
       |     +--ro error-timestamp?     yang:date-and-time
       |     +--ro error-node-id?       te-types:te-node-id
       |     +--ro error-link-id?       te-types:te-tp-id
       |     +--ro lsp-id?              uint16
       +--ro lsps
          +--ro lsp* [node lsp-id]
             +--ro tunnel-name?   -> /te/lsps/lsp/tunnel-name
             +--ro node?          leafref
             +--ro lsp-id?        leafref
     grouping path-computation-response:
       +--ro computed-paths-properties
       |  +--ro computed-path-properties* [k-index]
       |     +--ro k-index?                            uint8
       |     +---u te-types:generic-path-properties
       +--ro computed-path-error-infos
          +--ro computed-path-error-info* []
             +--ro error-description?   string
             +--ro error-timestamp?     yang:date-and-time
             +--ro error-reason?        identityref
     grouping protection-restoration-properties:
       +-- protection
       |  +-- protection-type?                identityref
       |  +-- protection-reversion-disable?   boolean
       |  +-- hold-off-time?                  uint32
       |  +-- wait-to-revert?                 uint16
       |  +-- aps-signal-id?                  uint8
       +-- restoration
          +-- restoration-type?                identityref
          +-- restoration-scheme?              identityref
          +-- restoration-reversion-disable?   boolean
          +-- hold-off-time?                   uint32
          +-- wait-to-restore?                 uint16
          +-- wait-to-revert?                  uint16
     grouping tunnel-associations-properties:
       +-- association-objects
          +-- association-object* [association-key]
          |  +-- association-key?   string
          |  +-- type?              identityref
          |  +-- id?                uint16
          |  +-- source
          |     +---u te-types:te-generic-node-id
          +-- association-object-extended* [association-key]
             +-- association-key?   string
             +-- type?              identityref
             +-- id?                uint16
             +-- source
             |  +---u te-types:te-generic-node-id
             +-- global-source?     uint32
             +-- extended-id?       yang:hex-string
     grouping tunnel-end-point:
       +-- node-id?        nw:node-id
       +-- te-node-id?     te-types:te-node-id
       +-- tunnel-tp-id?   binary
     grouping tunnel-common-attributes:
       +-- source
       |  +---u tunnel-end-point
       +-- destination
       |  +---u tunnel-end-point
       +-- bidirectional?   boolean
     grouping tunnel-hierarchy-properties:
       +-- hierarchy
          +-- dependency-tunnels
          |  +-- dependency-tunnel* [name]
          |     +-- name?                                   tunnel-ref
          |     +---u te-types:encoding-and-switching-type
          +-- hierarchical-link
             +-- enable?                            boolean
             +-- local-node-id?                     nw:node-id
             +-- local-te-node-id?                  te-types:te-node-id
             +-- local-link-tp-id?                  nt:tp-id
             +-- local-te-link-tp-id?               te-types:te-tp-id
             +-- remote-node-id?                    nw:node-id
             +-- remote-link-tp-id?                 nt:tp-id
             +-- remote-te-link-tp-id?              te-types:te-tp-id
             +-- remote-te-node-id?                 te-types:te-node-id
             +--ro link-id?                           nt:link-id
             +-- network-id?                        nw:network-id
             +---u te-types:te-topology-identifier
     grouping path-constraints-common:
       +---u te-types:common-path-constraints-attributes
       +---u te-types:generic-path-disjointness
       +---u te-types:path-constraints-route-objects
       +-- path-in-segment!
       │  +---u te-types:label-set-info
       +-- path-out-segment!
          +---u te-types:label-set-info
```

Figure 13: Full tree diagram of TE Tunnel YANG data model

14. References

14.1. Normative References

[I-D.draft-ietf-teas-rfc8776-update]
Busi, I., Guo, A., Liu, X., Saad, T., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", Work in Progress, Internet-Draft, draft-ietf-teas-rfc8776-update-09, 29 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-rfc8776-update-09>.

[ITU_G.808.1]
ITU-T Recommendation G.808.1, "Generic protection switching - Linear trail and subnetwork protection", ITU-T G.808.1 , May 2014.

[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>.

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

[RFC4427] Mannie, E., Ed. and D. Papadimitriou, Ed., "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4427, DOI 10.17487/RFC4427, March 2006, <https://www.rfc-editor.org/info/rfc4427>.

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

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

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

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

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

[RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP ASSOCIATION Object Extensions", RFC 6780, DOI 10.17487/RFC6780, October 2012, <https://www.rfc-editor.org/info/rfc6780>.

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

[RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <https://www.rfc-editor.org/info/rfc7271>.

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

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

[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>.

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

[RFC8234] Ryoo, J., Cheung, T., van Helvoort, H., Busi, I., and G. Wen, "Updates to MPLS Transport Profile (MPLS-TP) Linear Protection in Automatic Protection Switching (APS) Mode", RFC 8234, DOI 10.17487/RFC8234, August 2017, <https://www.rfc-editor.org/info/rfc8234>.

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

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

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

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

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

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

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

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <https://www.rfc-editor.org/info/rfc8570>.

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

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

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

[RFC9012] Patel, K., Van de Velde, G., Sangli, S., and J. Scudder, "The BGP Tunnel Encapsulation Attribute", RFC 9012, DOI 10.17487/RFC9012, April 2021, <https://www.rfc-editor.org/info/rfc9012>.

14.2. Informative References

[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>.

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

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

Authors' Addresses

Tarek Saad
Cisco Systems Inc
Email: tsaad.net@gmail.com

Rakesh Gandhi
Cisco Systems Inc
Email: rgandhi@cisco.com

Xufeng Liu
Alef Edge
Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin
Individual
Email: i_bryskin@yahoo.com

TEAS Working Group
Internet Draft
Intended status: Standards Track

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

Expires: December 19, 2019 June 19, 2019

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

Abstract

This document defines a YANG data model for representing, retrieving and manipulating Traffic Engineering (TE) Topologies. The model serves as a base model that other technology specific TE Topology models can augment.

Conventions used in this document

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

Table of Contents

1. Introduction

The Traffic Engineering Database (TED) is an essential component of Traffic Engineered (TE) systems that are based on MPLS-TE [RFC2702] and GMPLS [RFC3945]. The TED is a collection of all TE information about all TE nodes and TE links in the network. The TE Topology is a schematic arrangement of TE nodes and TE links present in a given TED. There could be one or more TE Topologies present in a given Traffic Engineered system. A TE Topology is the topology on which path computational algorithms are run to compute Traffic Engineered Paths (TE Paths).

This document defines a YANG [RFC7950] data model for representing and manipulating TE Topologies. This model contains technology agnostic TE Topology building blocks that can be augmented and used by other technology-specific TE Topology models.

1.1. Terminology

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

The reader is assumed to be familiar with general body of work captured in currently available TE related RFCs. [RFC7926] serves as a good starting point for those who may be less familiar with Traffic Engineering related RFCs.

Some of the key terms used in this document are:

TED: The Traffic Engineering Database is a collection of all TE information about all TE nodes and TE links in a given network.

TE-Topology: The TE Topology is a schematic arrangement of TE nodes and TE links in a given TED. It forms the basis for a graph suitable for TE path computations.

Native TE Topology: Native TE Topology is a topology that is native to a given provider network. Native TE topology could be discovered via various routing protocols and/or subscribe/publish techniques. This is the topology on which path computational algorithms are run to compute TE Paths.

Customized TE Topology: Customized TE Topology is a custom topology that is produced by a provider for a given client. This topology typically makes abstractions on the provider’s Native TE Topology, and is provided to the client. The client receives the Customized TE Topology, and merges it into the client’s Native TE Topology. The client’s path computational algorithms aren’t typically run on the Customized TE Topology; they are run on the client’s Native TE Topology after the merge.

1.2. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A. of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
nw	ietf-network	[RFC6991]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te-types]

Table 1: Prefixes and corresponding YANG modules

2. Characterizing TE Topologies

The data model proposed by this document takes the following characteristics of TE Topologies into account:

- TE Topology is an abstract control-plane representation of the data-plane topology. Hence attributes specific to the data-plane must make their way into the corresponding TE Topology modeling. The TE Topology comprises of dynamic auto-discovered data as well as fairly static data associated with data-plane nodes and links. The dynamic data may change frequently, such as unreserved bandwidth available on data-plane links. The static data rarely changes, such as layer network identification, switching and adaptation capabilities and limitations, fate sharing, and administrative colors. It is possible for a single TE Topology to encompass TE information at multiple switching layers.

- TE Topologies are protocol independent. Information about topological elements may be learnt via link-state protocols, but the topology can exist without being dependent on any particular protocol.

- TE Topology may not be congruent to the routing topology in a given TE System. The routing topology is constructed based on routing adjacencies. There isn’t always a one-to-one association between a TE-link and a routing adjacency. For example, the presence of a TE link between a pair of nodes doesn’t necessarily imply the existence of a routing-adjacency between these nodes. To learn more, see [I-D.ietf-teas-te-topo-and-tunnel-modeling] and [I-D.ietf-teas-yang-l3-te-topo].

- Each TE Topological element has at least one information source associated with it. In some scenarios, there could be more than one information source associated with any given topological element.

- TE Topologies can be hierarchical. Each node and link of a given TE Topology can be associated with respective underlay topology. This means that each node and link of a given TE Topology can be associated with an independent stack of supporting TE Topologies.

- TE Topologies can be customized. TE topologies of a given network presented by the network provider to its client could be customized on per-client request basis. This customization could be performed by provider, by client or by provider/client negotiation. The relationship between a customized topology and provider’s native topology could be captured as hierarchical (overlay-underlay), but otherwise the two topologies are decoupled from each other. A customized topology is presented to the client, while provider’s native topology is known in its entirety to the provider itself.

3. Modeling Abstractions and Transformations

```
                                | +---+          __
                                | |   | TE Node  \/ TTP  o LTP
                                | +---+
                                |
                                | ----- TE Link
                                | ***** Node Connectivity Matrix,
                                |       TTP Local Link Connectivity
                                | @@@@@ TE Tunnel
                                o----------------------------------

      Node-1                                            Node-3
   +------------+                                    +------------+
   |    TTP-1   |                                    |    TTP-1   |
   |LTP  __     |           TE-Tunel-1               |     __     |
   |-6   \/@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\/     |
   o    *   *   oLTP-1          Node-2         LTP-6o    *   *   o
   |   *    *   |           +------------+           |   *    *   |
   |  * TTP-2*  |           |            |           |  * TTP-2*  |
   | *   __   * |LTP-2 LTP-6|            |LTP-1 LTP-5| *   __   * |
   o*    \/    *o-----------o************o-----------o*    \/    *o
   |LTP *  *    | Link-12   |          * | Link-23   |    *  *    |
   |-5 *    *   |      LTP-5|        *   |LTP-2      |   *    *   |
   +--o------o--+           o************o           +--o------o--+
    LTP-4  LTP-3            | *   *    * |            LTP-4  LTP-3
                            |  **      * |
                            +--o------o--+
                             LTP-4  LTP-3
```

Figure 1: TE Topology Modeling Abstractions

3.1. TE Topology

TE topology is a traffic engineering representation of one or more layers of network topologies. TE topology is comprised of TE nodes (TE graph vertices) interconnected via TE links (TE graph edges). A TE topology is mapped to a TE graph.

3.2. TE Node

TE node is an element of a TE topology, presented as a vertex on TE graph. TE node represents one or several nodes, or a fraction of a node, which can be a switch or router that is physical or virtual. TE node belongs to and is fully defined in exactly one TE topology. TE node is assigned a unique ID within the TE topology scope. TE node attributes include information related to the data plane aspects of the associated node(s) (e.g. connectivity matrix), as well as configuration data (such as TE node name). A given TE node can be reached on the TE graph over one of TE links terminated by the TE node.

Multi-layer TE nodes providing switching functions at multiple network layers are an example where a physical node can be decomposed into multiple logical TE nodes, which are fractions of the physical node. Some of these (logical) TE nodes may reside in the client layer TE topology while the remaining TE nodes belong to the server layer TE topology.

In Figure 1, Node-1, Node-2, and Node-3 are TE nodes.

3.3. TE Link

TE link is an element of a TE topology, presented as an edge on TE graph. The arrows on an edge indicate one or both directions of the TE link. When there are a pair of parallel links of opposite directions, an edge without arrows is also used. TE link represents one or several (physical) links or a fraction of a link. TE link belongs to and is fully defined in exactly one TE topology. TE link is assigned a unique ID within the TE topology scope. TE link attributes include parameters related to the data plane aspects of the associated link(s) (e.g. unreserved bandwidth, resource maps/pools, etc.), as well as the configuration data (such as remote node/link IDs, SRLGs, administrative colors, etc.). TE link is connected to TE node, terminating the TE link via exactly one TE link termination point (LTP).

In Figure 1, Link-12 and Link-23 are TE links.

3.4. Transitional TE Link for Multi-Layer Topologies

Networks are typically composed of multiple network layers where one or multiple signals in the client layer network can be multiplexed and encapsulated into a server layer signal [RFC5212] [G.805]. The server layer signal can be carried in the server layer network across multiple nodes until the server layer signal is terminated and the client layer signals reappear in the node that terminates the server layer signal. Examples of multi-layer networks are: IP over MPLS over Ethernet, low order Optical Data Unit-k (ODUk) signals multiplexed into a high order ODUl (l>k) carried over an Optical Channel (OCh) signal in an optical transport network as defined in [G.872] and [G.709].

TE links as defined in Section 3.3. can be used to represent links within a network layer. In case of a multi-layer network, TE nodes and TE links only allow representation of each network layer as a separate TE topology. Each of these single layer TE topologies would be isolated from their client and their server layer TE topology, if present. The highest and the lowest network layer in the hierarchy only have a single adjacent layer below or above, respectively. Multiplexing of client layer signals and encapsulating them into a server layer signal requires a function that is provided inside a node (typically realized in hardware). This function is also called layer transition.

One of the key requirements for path computation is to be able to calculate a path between two endpoints across a multi-layer network based on the TE topology representing this multi-layer network. This means that an additional TE construct is needed that represents potential layer transitions in the multi-layer TE-topology that connects the TE-topologies representing each separate network layer. The so-called transitional TE link is such a construct and it represents the layer transition function residing inside a node that is decomposed into multiple logical nodes that are represented as TE nodes (see also the transitional link definition in [G.8080] for the optical transport network). Hence, a transitional TE link connects a client layer node with a server layer node. A TE link as defined in 3.3. has LTPs of exactly the same kind on each link end whereas the transitional TE link has client layer LTPs on the client side of the transitional link and in most cases a single server layer LTP on the server side. It should be noted that transitional links are a helper construct in the multi-layer TE topology and they only exist as long as they are not in use, as they represent potential connectivity. When the server layer trail has been established between the server layer LTP of two transitional links in the server layer network, the resulting client layer link in the data plane will be represented as a normal TE link in the client layer topology. The transitional TE links will re-appear when the server layer trail has been torn down.

```
                                |                 |
                                | +---+          ---
                                | |   | TE Node  \ / Transitional
                                | +---+           |  Link
                                |
                                | ----- Client Layer Link
                                | ===== Server Layer Link
                                | ***** Layer Boundary
                                o----------------------------------

      +------------------+
      | +------+         |                          +------+
   -----|Client|------+  |              Client -----|Client|
      | |Layer |---+  |  |              Layer       |Layer |
   -----|Switch|-+ |  |  |              Links  -----|Node  |
      | +------+ |  |  |  |                          +------+
      |          | |  |  |  Client                    | |  |
      |          | |  ---_| Layer                     --- ---
   ***|**********|*| \ /*|***************************\ /*\ /****
      |          ---  |  | Server       Transitional  |   |
      |    Layer \ /  |  | Layer               Links  |   |
      |    Term.  |   |  |                             |   |
      |           |   |  |                             |   |
      |         +------+ |                          +------+
   =============|Server|=====            Server ====|Server|====
      |         |Layer | |               Layer      |Layer |
   =============|Switch|=====            Links  ====|Node  |====
      |         +------+ |                          +------+
      +------------------+

       Physical Node View                      TE-Topology View
```

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

3.5. TE Link Termination Point (LTP)

TE link termination point (LTP) is a conceptual point of connection of a TE node to one of the TE links, terminated by the TE node. Cardinality between an LTP and the associated TE link is 1:0..1.

In Figure 1, Node-2 has six LTPs: LTP-1 to LTP-6.

3.6. TE Tunnel Termination Point (TTP)

TE tunnel termination point (TTP) is an element of TE topology representing one or several of potential transport service termination points (i.e. service client adaptation points such as

WDM/OCh transponder). TTP is associated with (hosted by) exactly one TE node. TTP is assigned a unique ID within the TE node scope. Depending on the TE node's internal constraints, a given TTP hosted by the TE node could be accessed via one, several or all TE links terminated by the TE node.

In Figure 1, Node-1 has two TTPs: TTP-1 and TTP-2.

3.7. TE Node Connectivity Matrix

TE node connectivity matrix is a TE node's attribute describing the TE node's switching limitations in a form of valid switching combinations of the TE node's LTPs (see below). From the point of view of a potential TE path arriving at the TE node at a given inbound LTP, the node's connectivity matrix describes valid (permissible) outbound LTPs for the TE path to leave the TE node from.

In Figure 1, the connectivity matrix on Node-2 is:
{<LTP-6, LTP-1>, <LTP-5, LTP-2>, <LTP-5, LTP-4>, <LTP-4, LTP-1>,
<LTP-3, LTP-2>}

3.8. TTP Local Link Connectivity List (LLCL)

TTP Local Link Connectivity List (LLCL) is a List of TE links terminated by the TTP hosting TE node (i.e. list of the TE link LTPs), which the TTP could be connected to. From the point of view of a potential TE path, LLCL provides a list of valid TE links the TE path needs to start/stop on for the connection, taking the TE path, to be successfully terminated on the TTP in question.

In Figure 1, the LLCL on Node-1 is:
{<TTP-1, LTP-5>, <TTP-1, LTP-2>, <TTP-2, LTP-3>, <TTP-2, LTP4>}

3.9. TE Path

TE path is an ordered list of TE links and/or TE nodes on the TE topology graph, inter-connecting a pair of TTPs to be taken by a potential connection. TE paths, for example, could be a product of successful path computation performed for a given transport service.

In Figure 1, the TE Path for TE-Tunnel-1 is:
{Node-1:TTP-1, Link-12, Node-2, Link-23, Node-3:TTP1}

3.10. TE Inter-Layer Lock

TE inter-layer lock is a modeling concept describing client-server layer adaptation relationships and hence important for the multi-layer traffic engineering. It is an association of M client layer LTPs and N server layer TTPs, within which data arriving at any of the client layer LTPs could be adopted onto any of the server layer TTPs. TE inter-layer lock is identified by inter-layer lock ID, which is unique across all TE topologies provided by the same provider. The client layer LTPs and the server layer TTPs associated within a given TE inter-layer lock are annotated with the same inter-layer lock ID attribute.

```
                                | +---+          __
                                | |   | TE Node  \/ TTP  o LTP
                                | +---+
                                |
                                | ----- TE Link
                                | ***** TTP Local Link Connectivity
                                o-----------------------------------

             (IL-1) C-LTP-1 +------------+   C-LTP-2 (IL-1)
                    --------O   (IL-1)   O--------
             (IL-1) C-LTP-3 |   S-TTP-1  |   C-LTP-4 (IL-1)
                    --------O     __     0--------
             (IL-1) C-LTP-5 |    *\/*    |   C-LTP-5 (IL-1)
                    --------O   *    *   O--------
                            |  *(IL-1)*  |
                    S-LTP-3 | * S-TTP-2* |   S-LTP-4
                    --------o*    __    *o--------
                            |    *\/*    |
                            |   *    *   |
                            +--o------o--+
                       S-LTP-1 |      | S-LTP-2
```

Figure 3: TE Inter-Layer Lock ID Associations

On the picture above a TE inter-layer lock with IL_1 ID associates 6 client layer LTPs (C-LTP-1 - C-LTP-6) with two server layer TTPs (S-TTP-1 and S-TTP-2). They all have the same attribute - TE inter-layer lock ID: IL-1, which is the only thing that indicates the association. A given LTP may have 0, 1 or more inter-layer lock IDs. In the latter case this means that the data arriving at the LTP may be adopted onto any of TTPs associated with all specified inter-layer locks. For example, C-LTP-1 could have two inter-layer lock IDs - IL-1 and IL-2. This would mean that C-LTP-1 for adaptation purposes could use not just TTPs associated with inter-layer lock IL-1 (i.e.

S-TTP-1 and S-TTP-2 on the picture), but any of TTPs associated with inter-layer lock IL-2 as well. Likewise, a given TTP may have one or more inter-layer lock IDs, meaning that it can offer the adaptation service to any of client layer LTPs with inter-layer lock ID matching one of its own. Additionally, each TTP has an attribute - Unreserved Adaptation Bandwidth, which announces its remaining adaptation resources sharable between all potential client LTPs.

LTPs and TTPs associated within the same TE inter-layer lock may be hosted by the same (hybrid, multi-layer) TE node or multiple TE nodes located in the same or separate TE topologies. The latter is especially important since TE topologies of different layer networks could be modeled by separate augmentations of the basic (common to all layers) TE topology model.

3.11. Underlay TE topology

Underlay TE topology is a TE topology that serves as a base for constructing of overlay TE topologies

3.12. Overlay TE topology

Overlay TE topology is a TE topology constructed based on one or more underlay TE topologies. Each TE node of the overlay TE topology represents an arbitrary segment of an underlay TE topology; each TE link of the overlay TE topology represents an arbitrary TE path in one of the underlay TE topologies. The overlay TE topology and the supporting underlay TE topologies may represent distinct layer networks (e.g. OTN/ODUk and WDM/OCh respectively) or the same layer network.

3.13. Abstract TE topology

Abstract TE topology is a topology that contains abstract topological elements (nodes, links, tunnel termination points). Abstract TE topology is an overlay TE topology created by a topology provider and customized for a topology provider's client based on one or more of the provider's native TE topologies (underlay TE topologies), the provider's policies and the client's preferences. For example, a first level topology provider (such as Domain Controller) can create an abstract TE topology for its client (e.g. Multi-Domain Service Coordinator) based on the provider's one or more native TE topologies, local policies/profiles and the client's TE topology configuration requests

Figure 4 shows an example of abstract TE topology.

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

     +---+              +---+
     |s31|--------------|S5 |
     +---+\           / +---+
           \         /
            \       /
             \+---+/                +---+
             /|AN1|\----------------|S8 |
            / +---+ \               +---+
     +---+ /         \ +---+
     |S9 |-------------|S11|
     +---+             +---+
           Abstract TE Topology

            +---+                    +---+
            |S1 |--------------------|S2 |
            +---+                    +---+
             /                          \
            /                            \
     +---+ /                  +---+       \ +---+
     |s3 |--------------------|S4 |---------|S5 |
     +---+\                   +---+         +---+
           \                      \             \
            \                      \             \
             \+---+                 +---+         +---+
             /|S6 |\                |S7 |---------|S8 |
            / +---+ \               +---+\       /+---+
     +---+ /         \ +---+              +---+ /
     |S9 |-------------|S10|--------------|S11|/
     +---+             +---+              +---+
           Native TE Topology
```

Figure 4: Abstract TE Topology

4. Model Applicability

4.1. Native TE Topologies

The model discussed in this draft can be used to represent and retrieve native TE topologies on a given TE system.

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

     +---+       +---+        +---+         +---+         +---+
     | R1|-------| R2|--------| R3|---------| R4|---------| R5|
     +---+       +---+        +---+         +---+         +---+
       |                      /   \         /   \         /
       |                     /     \       /     \       /
       |                    /       \     /       \     /
       |                   /         \   /         \   /
       |                  /           \ /           \ /
     +---+             +---+         +---+         +---+
     | R6|-------------| R7|         | R8|---------| R9|
     +---+             +---+         +---+         +---+
```

Figure 5a: Example Network Topology

Consider the network topology depicted in Figure 5a. R1 .. R9 are nodes representing routers. An implementation MAY choose to construct a native TE Topology using all nodes and links present in the given TED as depicted in Figure 5b. The data model proposed in this document can be used to retrieve/represent this TE topology.

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

      [R1] ++++ [R2] ++++ [R3] ++++ [R4] ++++ [R5]
        +                +    +    +    +    +
        +               +      +  +      +  +
        +              +        ++        ++
      [R6] +++++++++ [R7]      [R8] ++++ [R9]
```

Figure 5b: Native TE Topology as seen on Node R3

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

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

```
                                     :
   TE info distributed via ISIS-TE   :  TE info distributed via OSPF-TE
                                     :
          +---+       +---+        +---+         +---+         +---+
          | R1|-------| R2|--------| R3|---------| R4|---------| R5|
          +---+       +---+        +---+         +---+         +---+
            |                      / : \         /   \         /
            |                     /  :  \       /     \       /
            |                    /   :   \     /       \     /
            |                   /    :    \   /         \   /
            |                  /     :     \ /           \ /
          +---+             +---+    :    +---+         +---+
          | R6|-------------| R7|    :    | R8|---------| R9|
          +---+             +---+    :    +---+         +---+
                                     :
```

Figure 6a: Example Network Topology

```
           -----------------------   :  -----------------------
           |Native TE Topology   |   :  |Native TE Topology   |
           |Info-Source: ISIS-TE |   :  |Info-Source: OSPF-TE |
           -----------------------   :  -----------------------
                                     :
            [R1] ++++ [R2] ++++ [R3] : [R3’] ++++ [R4] ++++ [R5]
             +                +      :      +    +    +    +
             +               +       :       +  +      +  +
             +              +        :        ++        ++
            [R6] +++++++++ [R7]      :       [R8] ++++ [R9]
```

Figure 6b: Native TE Topologies as seen on Node R3

4.2. Customized TE Topologies

Customized TE topology is a topology that was modified by the provider to honor a particular client’s requirements or preferences. The model discussed in this draft can be used to represent, retrieve and manipulate customized TE Topologies. The model allows the provider to present the network in abstract TE Terms on a per client basis. These customized topologies contain sufficient information for the path computing client to select paths according to its policies.

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

     +---+        /-\          /-\           /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/          \-/           \-/          +---+
                              /   \         /   \
                             /     \       /     \
                            /       \     /       \
                           /         \   /         \
                          /           \ /           \
         +---+          /-\           /-\           /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

Figure 7: Example packet optical topology

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

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

     +---+        /-\ $$$$$$$$ /-\ $$$$$$$$$ /-\          +---+
     | R1|-------( A )--------( C )---------( E )---------| R3|
     +---+        \-/         @\-/ @@@@@@@@@ \-/          +---+
                             @/   \         /   \
                            @/     \       /     \
                           @/       \     /       \
                          @/         \   /         \
                         @/           \ /           \
         +---+          /-\ ********* /-\ ********* /-\          +---+
         | R2|---------( B )---------( D )---------( F )---------| R4|
         +---+          \-/           \-/           \-/          +---+
```

Figure 8a: Paths within the provider domain

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

Figure 8b: Customized TE Topology provided to the Client

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

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

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

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

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

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

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

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

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

4.3. Merging TE Topologies Provided by Multiple Providers

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

In particular, the process of merging TE topologies includes:

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

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

        Domain 1 TE Topology              Domain 2 TE Topology

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

Figure 9: Merging Domain TE Topologies

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

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

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

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

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

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

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

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

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

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


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

             +-------------+                 +-------------+
        -----|             |----          ----|             |----
             |             |                  |             |
             |    AN1      |----          ----|    AN1      |----
             |             |                  |             |
        -----|             |----          ----|             |----
             +-------------+                 +-------------+
```

Figure 10: Merging Domain TE Topologies

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

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

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

```
                Client's Merged TE Topology 2

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


                Client's Merged TE Topology 3

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

Figure 11: Multiple Native (Merged) Client's TE Topologies

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

5. Modeling Considerations

5.1. Network topology building blocks

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

```
                  +------------------------+
                  |                        |
                  | Network Topology Model |
                  | (ietf-network-topology)|
                  +------------------------+
                               |
                               |
                               |
                               V
                  +------------------------+
                  |       TE Topology      |
                  |         Model          |
                  |                        |
                  +------------------------+
```

Figure 12: Augmenting the Network Topology Model

5.2. Technology agnostic TE Topology model

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

```
                  +------------------------------+
                  |      TE Topology Model       |
                  |  (Defined in This Document)  |
                  +------------------------------+
                                 |
            +-------------+-------------+-------------+
            |             |             |             |
            V             V             V             V
      +------------+                              +------------+
      | Technology |                              | Technology |
      |  Specific  |   ......................     |  Specific  |
      | TE Topology|                              | TE Topology|
      |   Model 1  |                              |   Model n  |
      +------------+                              +------------+
```

Figure 13: Augmenting the Technology agnostic TE Topology model

5.3. Model Structure

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

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

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

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

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

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


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

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

5.4. Topology Identifiers

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

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

5.5. Generic TE Link Attributes

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

```
     +--rw te-link-attributes
        .....................
        +--rw admin-status?                   te-admin-status
        |  .....................
        +--rw link-index?                     uint64
        +--rw administrative-group?           te-types:admin-groups
        +--rw link-protection-type?           enumeration
        +--rw max-link-bandwidth?             te-bandwidth
        +--rw max-resv-link-bandwidth?          te-bandwidth
        +--rw unreserved-bandwidth* [priority]
        |  .....................
        +--rw te-default-metric?                uint32
        |  .....................
        +--rw te-srlgs
        +--rw te-nsrlgs {nsrlg}?        .....................
```

5.6. Generic TE Node Attributes

The model covers the definitions for generic TE Node attributes.

The definition of a generic connectivity matrix is shown below:

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

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

```
     +--rw tunnel-termination-point* [tunnel-tp-id]
        +--rw tunnel-tp-id               binary
        +--rw admin-status?              te-types:te-admin-status
        +--rw name?                      string
        +--rw switching-capability?      identityref
        +--rw encoding?                  identityref
        +--rw inter-layer-lock-id*       uint32
        +--rw protection-type?           Identityref
        +--rw client-layer-adaptation
        ...........
        +--rw local-link-connectivities
        ...........
        |  +--rw local-link-connectivity* [link-tp-ref]
        |     +--rw link-tp-ref                leafref
        |     +--rw label-restrictions
        ...........
        |     +--rw is-allowed?                 boolean
        |     +--rw underlay {te-topology-hierarchy}?
        ...........
        |     +--rw path-constraints
        ...........
        |     +--rw optimizations
        ...........
        |     +--ro path-properties
        ...........
        +--rw supporting-tunnel-termination-point* [node-ref tunnel-tp-
   ref]
           +--rw node-ref         inet:uri
           +--rw tunnel-tp-ref    binary
```

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

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

5.7. TED Information Sources

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

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

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

5.8. Overlay/Underlay Relationship

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

This relationship is captured via the "underlay-topology" field for the node and via the "underlay" field for the link. The use of these fields is optional and this functionality is tagged as a "feature" ("te-topology-hierarchy").

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

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

5.9. Templates

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

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

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

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

5.10. Scheduling Parameters

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

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

5.11. Notifications

Notifications are a key component of any topology data model.

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

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

6. Guidance for Writing Technology Specific TE Topology Augmentations

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

Consider the following technology specific example-topology model:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7. TE Topology YANG Module

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

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

  prefix "tet";

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

        This version of this YANG module is part of RFC XXXX; see the
        RFC itself for full legal notices.";

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

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

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

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

     /*
      * Typedefs
      */
     typedef geographic-coordinate-degree {
         type decimal64 {
           fraction-digits 8;
         }
         description
           "Decimal degree (DD) used to express latitude and longitude
            geographic coordinates.";
     } // geographic-coordinate-degree
     typedef te-info-source {
       type enumeration {
         enum "unknown" {
           description "The source is unknown.";
         }
         enum "locally-configured" {
           description "Configured entity.";
         }
         enum "ospfv2" {
           description "OSPFv2.";
         }
         enum "ospfv3" {
           description "OSPFv3.";
         }
         enum "isis" {
           description "ISIS.";
         }
         enum "bgp-ls" {
           description "BGP-LS.";
           reference
             "RFC 7752: North-Bound Distribution of Link-State and
              Traffic Engineering (TE) Information Using BGP";
         }
         enum "system-processed" {
           description "System processed entity.";
         }
         enum "other" {
           description "Other source.";
         }
       }
       description
         "Describining the type of source that has provided the
          related information, and the source credibility.";
     } // te-info-source

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

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

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

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

      grouping geolocation-container {
        description
          "A container containing a GPS location.";
        container geolocation{
          config false;
          description
            "A container containing a GPS location.";
          leaf altitude {
            type int64;
            units millimeter;
            description
              "Distance above the sea level.";
          }
          leaf latitude {
            type geographic-coordinate-degree {
              range "-90..90";
            }
            description
              "Relative position north or south on the Earth's surface.";
          }
          leaf longitude {
            type geographic-coordinate-degree {
              range "-180..180";
            }
            description
              "Angular distance east or west on the Earth's surface.";
          }
        } // gps-location
      } // geolocation-container

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

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

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

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

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

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

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

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

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

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

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

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

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

       container local-link-connectivity {
         description
           "Containing TE LLCL (Local Link Connectivity List) level
            statistics attributes.";
         leaf creates {
           type yang:counter32;
           description
             "Number of times that an LLCL entry was created.";
           reference
             "RFC 6241. Section 7.2 for 'create' operation.";
         }
         leaf deletes {
           type yang:counter32;
           description
             "Number of times that an LLCL entry was deleted.";
           reference
             "RFC 6241. Section 7.2 for 'delete' operation.";
         }
         leaf disables {
           type yang:counter32;
           description
             "Number of times that an LLCL  entry was disabled.";
         }
         leaf enables {
           type yang:counter32;
           description
             "Number of times that an LLCL  entry was enabled.";
         }
         leaf modifies {
           type yang:counter32;
           description
             "Number of times that an LLCL  entry was modified.";
         }
       } // local-link-connectivity
     } // statistics-per-ttp

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

     grouping te-link-config {
       description
```

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

               leaf sequence {
                 type uint32;
                 description
                   "Identify the sequence in the bundle.";
               }
               leaf src-interface-ref {
                 type string;
                 description
                   "Reference to component link interface on the
                    source node.";
               }
               leaf des-interface-ref {
                 type string;
                 description
                   "Reference to component link interface on the
                    destinatioin node.";
               }
             }
           }
         }
       } // bundle-stack-level

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

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

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

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

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

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

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

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

       container max-link-bandwidth {
         uses te-types:te-bandwidth;
         description
           "Maximum bandwidth that can be seen on this link in this
            direction. Units in bytes per second.";
         reference
           "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
            Version 2.
            RFC 5305: IS-IS Extensions for Traffic Engineering.";
       }
       container max-resv-link-bandwidth {
         uses te-types:te-bandwidth;
         description
           "Maximum amount of bandwidth that can be reserved in this
            direction in this link. Units in bytes per second.";
         reference
           "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
            Version 2.
            RFC 5305: IS-IS Extensions for Traffic Engineering.";
       }
       list unreserved-bandwidth {
         key "priority";
         max-elements "8";
         description
           "Unreserved bandwidth for 0-7 priority levels. Units in
            bytes per second.";
         reference
           "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
            Version 2.
            RFC 5305: IS-IS Extensions for Traffic Engineering.";
         leaf priority {
           type uint8 {
             range "0..7";
           }
           description "Priority.";
         }
         uses te-types:te-bandwidth;
       }
       leaf te-default-metric {
         type uint32;
         description
           "Traffic engineering metric.";
         reference
           "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
            Version 2.
            RFC 5305: IS-IS Extensions for Traffic Engineering.";
       }
       leaf te-delay-metric {
         type uint32;
         description
           "Traffic engineering delay metric.";
         reference
           "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.";
       }
       leaf te-igp-metric {
         type uint32;
         description
           "IGP metric used for traffic engineering.";
         reference
           "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
            Second MPLS Traffic Engineering (TE) Metric.";
       }
       container te-srlgs {
         description
           "Containing a list of SLRGs.";
         leaf-list value {
           type te-types:srlg;
           description "SRLG value.";
           reference
             "RFC 4202: Routing Extensions in Support of
              Generalized Multi-Protocol Label Switching (GMPLS).";
         }
       }
       container te-nsrlgs {
         if-feature nsrlg;
         description
           "Containing a list of NSRLGs (Not Sharing Risk Link
            Groups).
            When an abstract TE link is configured, this list specifies
            the request that underlay TE paths need to be mutually
            disjoint with other TE links in the same groups.";
         leaf-list id {
           type uint32;
```

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

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

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

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

     grouping te-link-underlay-attributes {
       description "Attributes for  te-link underlay.";
       reference
         "RFC 4206: Label Switched Paths (LSP) Hierarchy with
          Generalized Multi-Protocol Label Switching (GMPLS)
          Traffic Engineering (TE)";
       leaf enabled {
         type boolean;
         description
           "'true' if the underlay is enabled.
            'false' if the underlay is disabled.";
       }
       container primary-path {
         description
           "The service path on the underlay topology that
            supports this link.";
         uses nw:network-ref;
         list path-element {
           key "path-element-id";
           description
             "A list of path elements describing the service path.";
           leaf path-element-id {
             type uint32;
             description "To identify the element in a path.";
           }
           uses te-path-element;
         }
       } // primary-path
       list backup-path {
         key "index";
         description
           "A list of backup service paths on the underlay topology that
            protect the underlay primary path. If the primary path is
            not protected, the list contains zero elements. If the
            primary path is protected, the list contains one or more
            elements.";
         leaf index {
           type uint32;
           description
             "A sequence number to identify a backup path.";
         }
         uses nw:network-ref;
         list path-element {
           key "path-element-id";
           description
             "A list of path elements describing the backup service
              path";
           leaf path-element-id {
             type uint32;
             description "To identify the element in a path.";
           }
           uses te-path-element;
         }
       } // underlay-backup-path
       leaf protection-type {
         type identityref {
           base te-types:lsp-protection-type;
         }
         description
           "Underlay protection type desired for this link.";
       }
       container tunnel-termination-points {
         description
           "Underlay TTP(Tunnel Termination Points) desired for this
            link.";
         leaf source {
           type binary;
```

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

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

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

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

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

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

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

```
     } // te-node-config

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

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

     grouping te-node-connectivity-matrices {
       description "Connectivity matrix on a TE node.";
       container connectivity-matrices {
         description
           "Containing connectivity matrix on a TE node.";
         leaf number-of-entries {
           type uint16;
           description
             "The number of connectivity matrix entries.
              If this number is specified in the configuration request,
              the number is requested number of entries, which may not
              all be listed in the list;
              if this number is reported in the state data,
              the number is the current number of operational entries.";
         }
         uses te-types:label-set-info;
         uses connectivity-matrix-entry-path-attributes;
         list connectivity-matrix {
           key "id";
           description
             "Represents node’s switching limitations, i.e. limitations
              in interconnecting network TE links across the node.";
           reference
             "RFC 7579: General Network Element Constraint Encoding
              for GMPLS-Controlled Networks.";
           leaf id {
             type uint32;
             description "Identifies the connectivity-matrix entry.";
           }
         } // connectivity-matrix
       } // connectivity-matrices
     } // te-node-connectivity-matrices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

     /*
      * Data nodes
      */
     augment "/nw:networks/nw:network/nw:network-types" {
       description
         "Introduce new network type for TE topology.";
       container te-topology {
         presence "Indicates TE topology.";
         description
           "Its presence identifies the TE topology type.";
       }
     }

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

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

     augment "/nw:networks/nw:network/nw:node" {
       when "../nw:network-types/tet:te-topology" {
         description
           "Augmentation parameters apply only for networks with
            TE topology type.";
       }
       description
         "Configuration parameters for TE at node level.";
       leaf te-node-id {
         type te-types:te-node-id;
         description
           "The identifier of a node in the TE topology.
            A node is specific to a topology to which it belongs.";
       }
       container te {
         must "../te-node-id" {
           description
             "te-node-id is mandatory.";
         }
         must "count(../nw:supporting-node)<=1" {
           description
             "For a node in a TE topology, there cannot be more
              than 1 supporting node. If multiple nodes are abstracted,
              the underlay-topology is used.";
         }
         presence "TE support.";
         description
           "Indicates TE support.";
         uses te-node-augment;
       } // te
     }

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

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

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

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

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

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

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

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

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

8. Security Considerations

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

9. IANA Considerations

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

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

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

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

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

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

10. References

10.1. Normative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10.2. Informative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

March 2018.

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

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

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

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

11. Acknowledgments

The authors would like to thank Lou Berger, Sue Hares, Mazen Khaddam, Cyril Margaria and Zafar Ali for participating in design discussions and providing valuable insights.

Appendix A. Complete Model Tree Structure

```
   module: ietf-te-topology
     augment /nw:networks/nw:network/nw:network-types:
       +--rw te-topology!
     augment /nw:networks:
       +--rw te!
          +--rw templates
             +--rw node-template* [name] {template}?
             |  +--rw name
             |  |       te-types:te-template-name
             |  +--rw priority?                  uint16
             |  +--rw reference-change-policy?   enumeration
             |  +--rw te-node-attributes
             |     +--rw admin-status?        te-types:te-admin-status
             |     +--rw domain-id?           uint32
             |     +--rw is-abstract?         empty
             |     +--rw name?                string
             |     +--rw signaling-address*   inet:ip-address
             |     +--rw underlay-topology {te-topology-hierarchy}?
             |        +--rw network-ref?
             |                -> /nw:networks/network/network-id
             +--rw link-template* [name] {template}?
                +--rw name
                |       te-types:te-template-name
                +--rw priority?                  uint16
                +--rw reference-change-policy?   enumeration
                +--rw te-link-attributes
                   +--rw access-type?
                   |       te-types:te-link-access-type
                   +--rw external-domain
                   |  +--rw network-ref?
                   |  |       -> /nw:networks/network/network-id
                   |  +--rw remote-te-node-id?      te-types:te-node-id
                   |  +--rw remote-te-link-tp-id?   te-types:te-tp-id
                   +--rw is-abstract?                          empty
                   +--rw name?                                 string
                   +--rw underlay {te-topology-hierarchy}?
                   |  +--rw enabled?                         boolean
                   |  +--rw primary-path
                   |  |  +--rw network-ref?
                   |  |  |       -> /nw:networks/network/network-id
                   |  |  +--rw path-element* [path-element-id]
                   |  |     +--rw path-element-id               uint32
                   |  |     +--rw (type)?
                   |  |        +--:(numbered-node-hop)
                   |  |        |  +--rw numbered-node-hop
                   |  |        |     +--rw node-id     te-node-id
                   |  |        |     +--rw hop-type?   te-hop-type
                   |  |        +--:(numbered-link-hop)
                   |  |        |  +--rw numbered-link-hop
                   |  |        |     +--rw link-tp-id    te-tp-id
                   |  |        |     +--rw hop-type?     te-hop-type
                   |  |        |     +--rw direction?
                   |  |        |             te-link-direction
                   |  |        +--:(unnumbered-link-hop)
                   |  |        |  +--rw unnumbered-link-hop
                   |  |        |     +--rw link-tp-id    te-tp-id
                   |  |        |     +--rw node-id       te-node-id
                   |  |        |     +--rw hop-type?     te-hop-type
                   |  |        |     +--rw direction?
                   |  |        |             te-link-direction
                   |  |        +--:(as-number)
                   |  |        |  +--rw as-number-hop
                   |  |        |     +--rw as-number    inet:as-number
                   |  |        |     +--rw hop-type?    te-hop-type
                   |  |        +--:(label)
                   |  |           +--rw label-hop
                   |  |              +--rw te-label
                   |  |                 +--rw (technology)?
                   |  |                 |  +--:(generic)
                   |  |                 |     +--rw generic?
                   |  |                 |             rt-
   types:generalized-label
                   |  |                 +--rw direction?
                   |  |                         te-label-direction
                   |  +--rw backup-path* [index]
                   |  |  +--rw index           uint32
                   |  |  +--rw network-ref?
                   |  |  |       -> /nw:networks/network/network-id
                   |  |  +--rw path-element* [path-element-id]
                   |  |     +--rw path-element-id               uint32
                   |  |     +--rw (type)?
                   |  |        +--:(numbered-node-hop)
                   |  |        |  +--rw numbered-node-hop
                   |  |        |     +--rw node-id     te-node-id
                   |  |        |     +--rw hop-type?   te-hop-type
                   |  |        +--:(numbered-link-hop)
                   |  |        |  +--rw numbered-link-hop
                   |  |        |     +--rw link-tp-id    te-tp-id
                   |  |        |     +--rw hop-type?     te-hop-type
                   |  |        |     +--rw direction?
                   |  |        |             te-link-direction
                   |  |        +--:(unnumbered-link-hop)
                   |  |        |  +--rw unnumbered-link-hop
                   |  |        |     +--rw link-tp-id    te-tp-id
                   |  |        |     +--rw node-id       te-node-id
                   |  |        |     +--rw hop-type?     te-hop-type
                   |  |        |     +--rw direction?
                   |  |        |             te-link-direction
                   |  |        +--:(as-number)
                   |  |        |  +--rw as-number-hop
                   |  |        |     +--rw as-number    inet:as-number
                   |  |        |     +--rw hop-type?    te-hop-type
                   |  |        +--:(label)
                   |  |           +--rw label-hop
                   |  |              +--rw te-label
                   |  |                 +--rw (technology)?
                   |  |                 |  +--:(generic)
                   |  |                 |     +--rw generic?
                   |  |                 |             rt-
   types:generalized-label
                   |  |                 +--rw direction?
                   |  |                         te-label-direction
                   |  +--rw protection-type?             identityref
                   |  +--rw tunnel-termination-points
                   |  |  +--rw source?        binary
                   |  |  +--rw destination?   binary
                   |  +--rw tunnels
                   |     +--rw sharing?   boolean
                   |     +--rw tunnel* [tunnel-name]
                   |        +--rw tunnel-name    string
                   |        +--rw sharing?       boolean
                   |
                   +--rw admin-status?
                   |       te-types:te-admin-status
                   +--rw link-index?                       uint64
                   +--rw administrative-group?
                   |       te-types:admin-groups
                   +--rw interface-switching-capability*
                   |       [switching-capability encoding]
                   |  +--rw switching-capability    identityref
                   |  +--rw encoding                identityref
                   |  +--rw max-lsp-bandwidth* [priority]
                   |     +--rw priority        uint8
                   |     +--rw te-bandwidth
                   |        +--rw (technology)?
                   |           +--:(generic)
                   |              +--rw generic?   te-bandwidth
                   +--rw label-restrictions
                   |  +--rw label-restriction* [index]
                   |     +--rw restriction?    enumeration
                   |     +--rw index           uint32
                   |     +--rw label-start
                   |     |  +--rw te-label
                   |     |     +--rw (technology)?
                   |     |     |  +--:(generic)
                   |     |     |     +--rw generic?
                   |     |     |             rt-types:generalized-label
                   |     |     +--rw direction?       te-label-direction
                   |     +--rw label-end
                   |     |  +--rw te-label
                   |     |     +--rw (technology)?
                   |     |     |  +--:(generic)
                   |     |     |     +--rw generic?
                   |     |     |             rt-types:generalized-label
                   |     |     +--rw direction?       te-label-direction
                   |     +--rw label-step
                   |     |  +--rw (technology)?
                   |     |     +--:(generic)
                   |     |        +--rw generic?   int32
                   |     +--rw range-bitmap?   yang:hex-string
                   +--rw link-protection-type?             identityref
                   +--rw max-link-bandwidth
                   |  +--rw te-bandwidth
                   |     +--rw (technology)?
                   |        +--:(generic)
                   |           +--rw generic?   te-bandwidth
                   +--rw max-resv-link-bandwidth
                   |  +--rw te-bandwidth
                   |     +--rw (technology)?
                   |        +--:(generic)
                   |           +--rw generic?   te-bandwidth
                   +--rw unreserved-bandwidth* [priority]
                   |  +--rw priority        uint8
                   |  +--rw te-bandwidth
                   |     +--rw (technology)?
                   |        +--:(generic)
                   |           +--rw generic?   te-bandwidth
                   +--rw te-default-metric?                uint32
                   +--rw te-delay-metric?                  uint32
                   +--rw te-igp-metric?                    uint32
                   +--rw te-srlgs
                   |  +--rw value*   te-types:srlg
                   +--rw te-nsrlgs {nsrlg}?
                      +--rw id*   uint32
     augment /nw:networks/nw:network:
       +--rw te-topology-identifier
       |  +--rw provider-id?   te-global-id
       |  +--rw client-id?     te-global-id
       |  +--rw topology-id?   te-topology-id
       +--rw te!
          +--rw name?                     string
          +--rw preference?               uint8
          +--rw optimization-criterion?   identityref
          +--rw nsrlg* [id] {nsrlg}?
          |  +--rw id              uint32
          |  +--rw disjointness?   te-types:te-path-disjointness
          +--ro geolocation
             +--ro altitude?    int64
             +--ro latitude?    geographic-coordinate-degree
             +--ro longitude?   geographic-coordinate-degree
     augment /nw:networks/nw:network/nw:node:
       +--rw te-node-id?   te-types:te-node-id
       +--rw te!
          +--rw te-node-template*
          |       -> ../../../../te/templates/node-template/name
          |       {template}?
          +--rw te-node-attributes
          |  +--rw admin-status?             te-types:te-admin-status
          |  +--rw connectivity-matrices
          |  |  +--rw number-of-entries?     uint16
          |  |  +--rw label-restrictions
          |  |  |  +--rw label-restriction* [index]
          |  |  |     +--rw restriction?    enumeration
          |  |  |     +--rw index           uint32
          |  |  |     +--rw label-start
          |  |  |     |  +--rw te-label
          |  |  |     |     +--rw (technology)?
          |  |  |     |     |  +--:(generic)
          |  |  |     |     |     +--rw generic?
          |  |  |     |     |             rt-types:generalized-label
          |  |  |     |     +--rw direction?       te-label-direction
          |  |  |     +--rw label-end
          |  |  |     |  +--rw te-label
          |  |  |     |     +--rw (technology)?
          |  |  |     |     |  +--:(generic)
          |  |  |     |     |     +--rw generic?
          |  |  |     |     |             rt-types:generalized-label
          |  |  |     |     +--rw direction?       te-label-direction
          |  |  |     +--rw label-step
          |  |  |     |  +--rw (technology)?
          |  |  |     |     +--:(generic)
          |  |  |     |        +--rw generic?   int32
          |  |  |     +--rw range-bitmap?   yang:hex-string
          |  |  +--rw is-allowed?            boolean
          |  |  +--rw underlay {te-topology-hierarchy}?
          |  |  |  +--rw enabled?                     boolean
          |  |  |  +--rw primary-path
          |  |  |  |  +--rw network-ref?
          |  |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  |  +--rw path-element* [path-element-id]
          |  |  |  |     +--rw path-element-id              uint32
          |  |  |  |     +--rw (type)?
          |  |  |  |        +--:(numbered-node-hop)
          |  |  |  |        |  +--rw numbered-node-hop
          |  |  |  |        |     +--rw node-id     te-node-id
          |  |  |  |        |     +--rw hop-type?   te-hop-type
          |  |  |  |        +--:(numbered-link-hop)
          |  |  |  |        |  +--rw numbered-link-hop
          |  |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |  |        |     +--rw direction?    te-link-direction
          |  |  |  |        +--:(unnumbered-link-hop)
          |  |  |  |        |  +--rw unnumbered-link-hop
          |  |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |  |        |     +--rw node-id       te-node-id
          |  |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |  |        |     +--rw direction?    te-link-direction
          |  |  |  |        +--:(as-number)
          |  |  |  |        |  +--rw as-number-hop
          |  |  |  |        |     +--rw as-number    inet:as-number
          |  |  |  |        |     +--rw hop-type?    te-hop-type
          |  |  |  |        +--:(label)
          |  |  |  |           +--rw label-hop
          |  |  |  |              +--rw te-label
          |  |  |  |                 +--rw (technology)?
          |  |  |  |                 |  +--:(generic)
          |  |  |  |                 |     +--rw generic?
          |  |  |  |                 |             rt-types:generalized-
   label
          |  |  |  |                 +--rw direction?
          |  |  |  |                         te-label-direction
          |  |  |  +--rw backup-path* [index]
          |  |  |  |  +--rw index           uint32
          |  |  |  |  +--rw network-ref?
          |  |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  |  +--rw path-element* [path-element-id]
          |  |  |  |     +--rw path-element-id              uint32
          |  |  |  |     +--rw (type)?
          |  |  |  |        +--:(numbered-node-hop)
          |  |  |  |        |  +--rw numbered-node-hop
          |  |  |  |        |     +--rw node-id     te-node-id
          |  |  |  |        |     +--rw hop-type?   te-hop-type
          |  |  |  |        +--:(numbered-link-hop)
          |  |  |  |        |  +--rw numbered-link-hop
          |  |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |  |        |     +--rw direction?    te-link-direction
          |  |  |  |        +--:(unnumbered-link-hop)
          |  |  |  |        |  +--rw unnumbered-link-hop
          |  |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |  |        |     +--rw node-id       te-node-id
          |  |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |  |        |     +--rw direction?    te-link-direction
          |  |  |  |        +--:(as-number)
          |  |  |  |        |  +--rw as-number-hop
          |  |  |  |        |     +--rw as-number    inet:as-number
          |  |  |  |        |     +--rw hop-type?    te-hop-type
          |  |  |  |        +--:(label)
          |  |  |  |           +--rw label-hop
          |  |  |  |              +--rw te-label
          |  |  |  |                 +--rw (technology)?
          |  |  |  |                 |  +--:(generic)
          |  |  |  |                 |     +--rw generic?
          |  |  |  |                 |             rt-types:generalized-
   label
          |  |  |  |                 +--rw direction?
          |  |  |  |                         te-label-direction
          |  |  |  +--rw protection-type?             identityref
          |  |  |  +--rw tunnel-termination-points
          |  |  |  |  +--rw source?        binary
          |  |  |  |  +--rw destination?   binary
          |  |  |  +--rw tunnels
          |  |  |     +--rw sharing?   boolean
          |  |  |     +--rw tunnel* [tunnel-name]
          |  |  |        +--rw tunnel-name    string
          |  |  |        +--rw sharing?       boolean
          |  |  +--rw path-constraints
          |  |  |  +--rw te-bandwidth
          |  |  |  |  +--rw (technology)?
          |  |  |  |     +--:(generic)
          |  |  |  |        +--rw generic?   te-bandwidth
          |  |  |  +--rw link-protection?          identityref
          |  |  |  +--rw setup-priority?           uint8
          |  |  |  +--rw hold-priority?            uint8
          |  |  |  +--rw signaling-type?           identityref
          |  |  |  +--rw path-metric-bounds
          |  |  |  |  +--rw path-metric-bound* [metric-type]
          |  |  |  |     +--rw metric-type    identityref
          |  |  |  |     +--rw upper-bound?   uint64
          |  |  |  +--rw path-affinities-values
          |  |  |  |  +--rw path-affinities-value* [usage]
          |  |  |  |     +--rw usage    identityref
          |  |  |  |     +--rw value?   admin-groups
          |  |  |  +--rw path-affinity-names
          |  |  |  |  +--rw path-affinity-name* [usage]
          |  |  |  |     +--rw usage            identityref
          |  |  |  |     +--rw affinity-name* [name]
          |  |  |  |        +--rw name    string
          |  |  |  +--rw path-srlgs-lists
          |  |  |  |  +--rw path-srlgs-list* [usage]
          |  |  |  |     +--rw usage     identityref
          |  |  |  |     +--rw values*   srlg
          |  |  |  +--rw path-srlgs-names
          |  |  |  |  +--rw path-srlgs-name* [usage]
          |  |  |  |     +--rw usage    identityref
          |  |  |  |     +--rw names*   string
          |  |  |  +--rw disjointness?             te-path-disjointness
          |  |  +--rw optimizations
          |  |  |  +--rw (algorithm)?
          |  |  |     +--:(metric) {path-optimization-metric}?
          |  |  |     |  +--rw optimization-metric* [metric-type]
          |  |  |     |  |  +--rw metric-type
          |  |  |     |  |  |       identityref
          |  |  |     |  |  +--rw weight?
          |  |  |     |  |  |       uint8
          |  |  |     |  |  +--rw explicit-route-exclude-objects
          |  |  |     |  |  |  +--rw route-object-exclude-object*
          |  |  |     |  |  |          [index]
          |  |  |     |  |  |     +--rw index
          |  |  |     |  |  |     |       uint32
          |  |  |     |  |  |     +--rw (type)?
          |  |  |     |  |  |        +--:(numbered-node-hop)
          |  |  |     |  |  |        |  +--rw numbered-node-hop
          |  |  |     |  |  |        |     +--rw node-id     te-node-id
          |  |  |     |  |  |        |     +--rw hop-type?   te-hop-type
          |  |  |     |  |  |        +--:(numbered-link-hop)
          |  |  |     |  |  |        |  +--rw numbered-link-hop
          |  |  |     |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |     |  |  |        |     +--rw hop-type?
          |  |  |     |  |  |        |     |       te-hop-type
          |  |  |     |  |  |        |     +--rw direction?
          |  |  |     |  |  |        |             te-link-direction
          |  |  |     |  |  |        +--:(unnumbered-link-hop)
          |  |  |     |  |  |        |  +--rw unnumbered-link-hop
          |  |  |     |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |     |  |  |        |     +--rw node-id
          |  |  |     |  |  |        |     |       te-node-id
          |  |  |     |  |  |        |     +--rw hop-type?
          |  |  |     |  |  |        |     |       te-hop-type
          |  |  |     |  |  |        |     +--rw direction?
          |  |  |     |  |  |        |             te-link-direction
          |  |  |     |  |  |        +--:(as-number)
          |  |  |     |  |  |        |  +--rw as-number-hop
          |  |  |     |  |  |        |     +--rw as-number
          |  |  |     |  |  |        |     |       inet:as-number
          |  |  |     |  |  |        |     +--rw hop-type?
          |  |  |     |  |  |        |             te-hop-type
          |  |  |     |  |  |        +--:(label)
          |  |  |     |  |  |        |  +--rw label-hop
          |  |  |     |  |  |        |     +--rw te-label
          |  |  |     |  |  |        |        +--rw (technology)?
          |  |  |     |  |  |        |        |  +--:(generic)
          |  |  |     |  |  |        |        |     +--rw generic?
          |  |  |     |  |  |        |        |              rt-
   types:generalized-label
          |  |  |     |  |  |        |        +--rw direction?
          |  |  |     |  |  |        |                te-label-direction
          |  |  |     |  |  |        +--:(srlg)
          |  |  |     |  |  |           +--rw srlg
          |  |  |     |  |  |              +--rw srlg?   uint32
          |  |  |     |  |  +--rw explicit-route-include-objects
          |  |  |     |  |     +--rw route-object-include-object*
          |  |  |     |  |             [index]
          |  |  |     |  |        +--rw index
          |  |  |     |  |        |       uint32
          |  |  |     |  |        +--rw (type)?
          |  |  |     |  |           +--:(numbered-node-hop)
          |  |  |     |  |           |  +--rw numbered-node-hop
          |  |  |     |  |           |     +--rw node-id     te-node-id
          │  │  │     │  │           │     +--rw hop-type?   te-hop-type
          │  │  │     │  │           +--:(numbered-link-hop)
          │  │  │     │  │           │  +--rw numbered-link-hop
          │  │  │     │  │           │     +--rw link-tp-id    te-tp-id
          │  │  │     │  │           │     +--rw hop-type?
          │  │  │     │  │           │     │       te-hop-type
          │  │  │     │  │           │     +--rw direction?
          │  │  │     │  │           │             te-link-direction
          │  │  │     │  │           +--:(unnumbered-link-hop)
          │  │  │     │  │           │  +--rw unnumbered-link-hop
          │  │  │     │  │           │     +--rw link-tp-id    te-tp-id
          │  │  │     │  │           │     +--rw node-id
          │  │  │     │  │           │     │       te-node-id
          │  │  │     │  │           │     +--rw hop-type?
          │  │  │     │  │           │     │       te-hop-type
          │  │  │     │  │           │     +--rw direction?
          │  │  │     │  │           │             te-link-direction
          │  │  │     │  │           +--:(as-number)
          │  │  │     │  │           │  +--rw as-number-hop
          │  │  │     │  │           │     +--rw as-number
          │  │  │     │  │           │     │       inet:as-number
          │  │  │     │  │           │     +--rw hop-type?
          │  │  │     │  │           │             te-hop-type
          │  │  │     │  │           +--:(label)
          │  │  │     │  │              +--rw label-hop
          │  │  │     │  │                 +--rw te-label
          │  │  │     │  │                    +--rw (technology)?
          │  │  │     │  │                    │  +--:(generic)
          │  │  │     │  │                    │     +--rw generic?
          │  │  │     │  │                    │             rt-
   types:generalized-label
          │  │  │     │  │                    +--rw direction?
          │  │  │     │  │                            te-label-direction
          │  │  │     │  +--rw tiebreakers
          │  │  │     │     +--rw tiebreaker* [tiebreaker-type]
          │  │  │     │        +--rw tiebreaker-type    identityref
          │  │  │     +--:(objective-function)
          │  │  │              {path-optimization-objective-function}?
          │  │  │        +--rw objective-function
          │  │  │           +--rw objective-function-type?   identityref
          │  │  +--ro path-properties
          |  |  |  +--ro path-metric* [metric-type]
          |  |  |  |  +--ro metric-type           identityref
          |  |  |  |  +--ro accumulative-value?   uint64
          |  |  |  +--ro path-affinities-values
          |  |  |  |  +--ro path-affinities-value* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro value?   admin-groups
          |  |  |  +--ro path-affinity-names
          |  |  |  |  +--ro path-affinity-name* [usage]
          |  |  |  |     +--ro usage            identityref
          |  |  |  |     +--ro affinity-name* [name]
          |  |  |  |        +--ro name    string
          |  |  |  +--ro path-srlgs-lists
          |  |  |  |  +--ro path-srlgs-list* [usage]
          |  |  |  |     +--ro usage     identityref
          |  |  |  |     +--ro values*   srlg
          |  |  |  +--ro path-srlgs-names
          |  |  |  |  +--ro path-srlgs-name* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro names*   string
          |  |  |  +--ro path-route-objects
          |  |  |     +--ro path-route-object* [index]
          |  |  |        +--ro index                        uint32
          |  |  |        +--ro (type)?
          |  |  |           +--:(numbered-node-hop)
          |  |  |           |  +--ro numbered-node-hop
          |  |  |           |     +--ro node-id     te-node-id
          |  |  |           |     +--ro hop-type?   te-hop-type
          |  |  |           +--:(numbered-link-hop)
          |  |  |           |  +--ro numbered-link-hop
          |  |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |           |     +--ro hop-type?     te-hop-type
          |  |  |           |     +--ro direction?    te-link-direction
          |  |  |           +--:(unnumbered-link-hop)
          |  |  |           |  +--ro unnumbered-link-hop
          |  |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |           |     +--ro node-id       te-node-id
          |  |  |           |     +--ro hop-type?     te-hop-type
          |  |  |           |     +--ro direction?    te-link-direction
          |  |  |           +--:(as-number)
          |  |  |           |  +--ro as-number-hop
          |  |  |           |     +--ro as-number    inet:as-number
          |  |  |           |     +--ro hop-type?    te-hop-type
          |  |  |           +--:(label)
          |  |  |              +--ro label-hop
          |  |  |                 +--ro te-label
          |  |  |                    +--ro (technology)?
          |  |  |                    |  +--:(generic)
          |  |  |                    |     +--ro generic?
          |  |  |                    |             rt-types:generalized-
   label
          |  |  |                    +--ro direction?
          |  |  |                            te-label-direction
          |  |  +--rw connectivity-matrix* [id]
          |  |     +--rw id                  uint32
          |  |     +--rw from
          |  |     |  +--rw tp-ref?               leafref
          |  |     |  +--rw label-restrictions
          |  |     |     +--rw label-restriction* [index]
          |  |     |        +--rw restriction?    enumeration
          |  |     |        +--rw index           uint32
          |  |     |        +--rw label-start
          |  |     |        |  +--rw te-label
          |  |     |        |     +--rw (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--rw generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--rw direction?
          |  |     |        |             te-label-direction
          |  |     |        +--rw label-end
          |  |     |        |  +--rw te-label
          |  |     |        |     +--rw (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--rw generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--rw direction?
          |  |     |        |             te-label-direction
          |  |     |        +--rw label-step
          |  |     |        |  +--rw (technology)?
          |  |     |        |     +--:(generic)
          |  |     |        |        +--rw generic?   int32
          |  |     |        +--rw range-bitmap?   yang:hex-string
          |  |     +--rw to
          |  |     |  +--rw tp-ref?               leafref
          |  |     |  +--rw label-restrictions
          |  |     |     +--rw label-restriction* [index]
          |  |     |        +--rw restriction?    enumeration
          |  |     |        +--rw index           uint32
          |  |     |        +--rw label-start
          |  |     |        |  +--rw te-label
          |  |     |        |     +--rw (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--rw generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--rw direction?
          |  |     |        |             te-label-direction
          |  |     |        +--rw label-end
          |  |     |        |  +--rw te-label
          |  |     |        |     +--rw (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--rw generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--rw direction?
          |  |     |        |             te-label-direction
          |  |     |        +--rw label-step
          |  |     |        |  +--rw (technology)?
          |  |     |        |     +--:(generic)
          |  |     |        |        +--rw generic?   int32
          |  |     |        +--rw range-bitmap?   yang:hex-string
          |  |     +--rw is-allowed?         boolean
          |  |     +--rw underlay {te-topology-hierarchy}?
          |  |     |  +--rw enabled?                     boolean
          |  |     |  +--rw primary-path
          |  |     |  |  +--rw network-ref?
          |  |     |  |  |       -> /nw:networks/network/network-id
          |  |     |  |  +--rw path-element* [path-element-id]
          |  |     |  |     +--rw path-element-id              uint32
          |  |     |  |     +--rw (type)?
          |  |     |  |        +--:(numbered-node-hop)
          |  |     |  |        |  +--rw numbered-node-hop
          |  |     |  |        |     +--rw node-id     te-node-id
          |  |     |  |        |     +--rw hop-type?   te-hop-type
          |  |     |  |        +--:(numbered-link-hop)
          |  |     |  |        |  +--rw numbered-link-hop
          |  |     |  |        |     +--rw link-tp-id    te-tp-id
          |  |     |  |        |     +--rw hop-type?     te-hop-type
          |  |     |  |        |     +--rw direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(unnumbered-link-hop)
          |  |     |  |        |  +--rw unnumbered-link-hop
          |  |     |  |        |     +--rw link-tp-id    te-tp-id
          |  |     |  |        |     +--rw node-id       te-node-id
          |  |     |  |        |     +--rw hop-type?     te-hop-type
          |  |     |  |        |     +--rw direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(as-number)
          |  |     |  |        |  +--rw as-number-hop
          |  |     |  |        |     +--rw as-number    inet:as-number
          |  |     |  |        |     +--rw hop-type?    te-hop-type
          |  |     |  |        +--:(label)
          |  |     |  |           +--rw label-hop
          |  |     |  |              +--rw te-label
          |  |     |  |                 +--rw (technology)?
          |  |     |  |                 |  +--:(generic)
          |  |     |  |                 |     +--rw generic?
          |  |     |  |                 |             rt-
   types:generalized-label
          |  |     |  |                 +--rw direction?
          |  |     |  |                         te-label-direction
          |  |     |  +--rw backup-path* [index]
          |  |     |  |  +--rw index           uint32
          |  |     |  |  +--rw network-ref?
          |  |     |  |  |       -> /nw:networks/network/network-id
          |  |     |  |  +--rw path-element* [path-element-id]
          |  |     |  |     +--rw path-element-id               uint32
          |  |     |  |     +--rw (type)?
          |  |     |  |        +--:(numbered-node-hop)
          |  |     |  |        |  +--rw numbered-node-hop
          |  |     |  |        |     +--rw node-id     te-node-id
          |  |     |  |        |     +--rw hop-type?   te-hop-type
          |  |     |  |        +--:(numbered-link-hop)
          |  |     |  |        |  +--rw numbered-link-hop
          |  |     |  |        |     +--rw link-tp-id    te-tp-id
          |  |     |  |        |     +--rw hop-type?     te-hop-type
          |  |     |  |        |     +--rw direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(unnumbered-link-hop)
          |  |     |  |        |  +--rw unnumbered-link-hop
          |  |     |  |        |     +--rw link-tp-id    te-tp-id
          |  |     |  |        |     +--rw node-id       te-node-id
          |  |     |  |        |     +--rw hop-type?     te-hop-type
          |  |     |  |        |     +--rw direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(as-number)
          |  |     |  |        |  +--rw as-number-hop
          |  |     |  |        |     +--rw as-number    inet:as-number
          |  |     |  |        |     +--rw hop-type?    te-hop-type
          |  |     |  |        +--:(label)
          |  |     |  |           +--rw label-hop
          |  |     |  |              +--rw te-label
          |  |     |  |                 +--rw (technology)?
          |  |     |  |                 |  +--:(generic)
          |  |     |  |                 |     +--rw generic?
          |  |     |  |                 |             rt-
   types:generalized-label
          |  |     |  |                 +--rw direction?
          |  |     |  |                         te-label-direction
          |  |     |  +--rw protection-type?             identityref
          |  |     |  +--rw tunnel-termination-points
          |  |     |  |  +--rw source?        binary
          |  |     |  |  +--rw destination?   binary
          |  |     |  +--rw tunnels
          |  |     |     +--rw sharing?   boolean
          |  |     |     +--rw tunnel* [tunnel-name]
          |  |     |        +--rw tunnel-name    string
          |  |     |        +--rw sharing?       boolean
          |  |     +--rw path-constraints
          |  |     |  +--rw te-bandwidth
          |  |     |  |  +--rw (technology)?
          |  |     |  |     +--:(generic)
          |  |     |  |        +--rw generic?   te-bandwidth
          |  |     |  +--rw link-protection?          identityref
          |  |     |  +--rw setup-priority?           uint8
          |  |     |  +--rw hold-priority?            uint8
          |  |     |  +--rw signaling-type?           identityref
          |  |     |  +--rw path-metric-bounds
          |  |     |  |  +--rw path-metric-bound* [metric-type]
          |  |     |  |     +--rw metric-type    identityref
          |  |     |  |     +--rw upper-bound?   uint64
          |  |     |  +--rw path-affinities-values
          |  |     |  |  +--rw path-affinities-value* [usage]
          |  |     |  |     +--rw usage    identityref
          |  |     |  |     +--rw value?   admin-groups
          |  |     |  +--rw path-affinity-names
          |  |     |  |  +--rw path-affinity-name* [usage]
          |  |     |  |     +--rw usage            identityref
          |  |     |  |     +--rw affinity-name* [name]
          |  |     |  |        +--rw name    string
          |  |     |  +--rw path-srlgs-lists
          |  |     |  |  +--rw path-srlgs-list* [usage]
          |  |     |  |     +--rw usage     identityref
          |  |     |  |     +--rw values*   srlg
          |  |     |  +--rw path-srlgs-names
          |  |     |  |  +--rw path-srlgs-name* [usage]
          |  |     |  |     +--rw usage    identityref
          |  |     |  |     +--rw names*   string
          |  |     |  +--rw disjointness?
          |  |     |          te-path-disjointness
          |  |     +--rw optimizations
          |  |     |  +--rw (algorithm)?
          |  |     |     +--:(metric) {path-optimization-metric}?
          |  |     |     |  +--rw optimization-metric* [metric-type]
          |  |     |     |  |  +--rw metric-type
          |  |     |     |  |  |       identityref
          |  |     |     |  |  +--rw weight?
          |  |     |     |  |  |       uint8
          |  |     |     |  |  +--rw explicit-route-exclude-objects
          |  |     |     |  |  |  +--rw route-object-exclude-object*
          |  |     |     |  |  |          [index]
          |  |     |     |  |  |     +--rw index
          |  |     |     |  |  |     |       uint32
          |  |     |     |  |  |     +--rw (type)?
          |  |     |     |  |  |        +--:(numbered-node-hop)
          |  |     |     |  |  |        |  +--rw numbered-node-hop
          |  |     |     |  |  |        |     +--rw node-id
          |  |     |     |  |  |        |     |       te-node-id
          |  |     |     |  |  |        |     +--rw hop-type?
          |  |     |     |  |  |        |             te-hop-type
          |  |     |     |  |  |        +--:(numbered-link-hop)
          |  |     |     |  |  |        |  +--rw numbered-link-hop
          |  |     |     |  |  |        |     +--rw link-tp-id
          |  |     |     |  |  |        |     |       te-tp-id
          |  |     |     |  |  |        |     +--rw hop-type?
          |  |     |     |  |  |        |     |       te-hop-type
          |  |     |     |  |  |        |     +--rw direction?
          |  |     |     |  |  |        |             te-link-direction
          |  |     |     |  |  |        +--:(unnumbered-link-hop)
          |  |     |     |  |  |        |  +--rw unnumbered-link-hop
          |  |     |     |  |  |        |     +--rw link-tp-id
          |  |     |     |  |  |        |     |       te-tp-id
          |  |     |     |  |  |        |     +--rw node-id
          |  |     |     |  |  |        |     |       te-node-id
          |  |     |     |  |  |        |     +--rw hop-type?
          |  |     |     |  |  |        |     |       te-hop-type
          |  |     |     |  |  |        |     +--rw direction?
          |  |     |     |  |  |        |             te-link-direction
          |  |     |     |  |  |        +--:(as-number)
          |  |     |     |  |  |        |  +--rw as-number-hop
          |  |     |     |  |  |        |     +--rw as-number
          |  |     |     |  |  |        |     |       inet:as-number
          |  |     |     |  |  |        |     +--rw hop-type?
          |  |     |     |  |  |        |             te-hop-type
          |  |     |     |  |  |        +--:(label)
          |  |     |     |  |  |        |  +--rw label-hop
          |  |     |     |  |  |        |     +--rw te-label
          |  |     |     |  |  |        |        +--rw (technology)?
          |  |     |     |  |  |        |        |  +--:(generic)
          |  |     |     |  |  |        |        |     +--rw generic?
          |  |     |     |  |  |        |        |            rt-
   types:generalized-label
          |  |     |     |  |  |        |        +--rw direction?
          |  |     |     |  |  |        |                te-label-
   direction
```

```
          |  |     |     |  |  |        +--:(srlg)
          |  |     |     |  |  |           +--rw srlg
          |  |     |     |  |  |              +--rw srlg?   uint32
          |  |     |     |  |  +--rw explicit-route-include-objects
          |  |     |     |  |     +--rw route-object-include-object*
          |  |     |     |  |             [index]
          |  |     |     |  |        +--rw index
          |  |     |     |  |        |       uint32
          |  |     |     |  |        +--rw (type)?
          |  |     |     |  |           +--:(numbered-node-hop)
          |  |     |     |  |           |  +--rw numbered-node-hop
          |  |     |     |  |           |     +--rw node-id
          |  |     |     |  |           |     |       te-node-id
          |  |     |     |  |           |     +--rw hop-type?
          |  |     |     |  |           |             te-hop-type
          |  |     |     |  |           +--:(numbered-link-hop)
          |  |     |     |  |           |  +--rw numbered-link-hop
          |  |     |     |  |           |     +--rw link-tp-id
          |  |     |     |  |           |     |       te-tp-id
          |  |     |     |  |           |     +--rw hop-type?
          |  |     |     |  |           |     |       te-hop-type
          |  |     |     |  |           |     +--rw direction?
          |  |     |     |  |           |             te-link-direction
          |  |     |     |  |           +--:(unnumbered-link-hop)
          |  |     |     |  |           |  +--rw unnumbered-link-hop
          |  |     |     |  |           |     +--rw link-tp-id
          |  |     |     |  |           |     |       te-tp-id
          |  |     |     |  |           |     +--rw node-id
          |  |     |     |  |           |     |       te-node-id
          |  |     |     |  |           |     +--rw hop-type?
          |  |     |     |  |           |     |       te-hop-type
          |  |     |     |  |           |     +--rw direction?
          |  |     |     |  |           |             te-link-direction
          |  |     |     |  |           +--:(as-number)
          |  |     |     |  |           |  +--rw as-number-hop
          |  |     |     |  |           |     +--rw as-number
          |  |     |     |  |           |     |       inet:as-number
          |  |     |     |  |           |     +--rw hop-type?
          |  |     |     |  |           |             te-hop-type
          |  |     |     |  |           +--:(label)
          |  |     |     |  |              +--rw label-hop
          |  |     |     |  |                 +--rw te-label
          |  |     |     |  |                    +--rw (technology)?
          |  |     |     |  |                    |  +--:(generic)
          |  |     |     |  |                    |     +--rw generic?
          |  |     |     |  |                    |             rt-
   types:generalized-label
          |  |     |     |  |                    +--rw direction?
          |  |     |     |  |                            te-label-
   direction
          |  |     |     |  +--rw tiebreakers
          |  |     |     |     +--rw tiebreaker* [tiebreaker-type]
          |  |     |     |        +--rw tiebreaker-type    identityref
          |  |     |     +--:(objective-function)
          |  |     |              {path-optimization-objective-
   function}?
          |  |     |        +--rw objective-function
          |  |     |           +--rw objective-function-type?
          |  |     |                   identityref
          |  |     +--ro path-properties
          |  |        +--ro path-metric* [metric-type]
          |  |        |  +--ro metric-type           identityref
          |  |        |  +--ro accumulative-value?   uint64
          |  |        +--ro path-affinities-values
          |  |        |  +--ro path-affinities-value* [usage]
          |  |        |     +--ro usage    identityref
          |  |        |     +--ro value?   admin-groups
          |  |        +--ro path-affinity-names
          |  |        |  +--ro path-affinity-name* [usage]
          |  |        |     +--ro usage            identityref
          |  |        |     +--ro affinity-name* [name]
          |  |        |        +--ro name    string
          |  |        +--ro path-srlgs-lists
          |  |        |  +--ro path-srlgs-list* [usage]
          |  |        |     +--ro usage     identityref
          |  |        |     +--ro values*   srlg
          |  |        +--ro path-srlgs-names
          |  |        |  +--ro path-srlgs-name* [usage]
          |  |        |     +--ro usage    identityref
          |  |        |     +--ro names*   string
          |  |        +--ro path-route-objects
          |  |           +--ro path-route-object* [index]
          |  |              +--ro index                         uint32
          |  |              +--ro (type)?
          |  |                 +--:(numbered-node-hop)
          |  |                 |  +--ro numbered-node-hop
          |  |                 |     +--ro node-id     te-node-id
          |  |                 |     +--ro hop-type?   te-hop-type
          |  |                 +--:(numbered-link-hop)
          |  |                 |  +--ro numbered-link-hop
          |  |                 |     +--ro link-tp-id    te-tp-id
          |  |                 |     +--ro hop-type?     te-hop-type
          |  |                 |     +--ro direction?
          |  |                 |             te-link-direction
          |  |                 +--:(unnumbered-link-hop)
          |  |                 |  +--ro unnumbered-link-hop
          |  |                 |     +--ro link-tp-id    te-tp-id
          |  |                 |     +--ro node-id       te-node-id
          |  |                 |     +--ro hop-type?     te-hop-type
          |  |                 |     +--ro direction?
          |  |                 |             te-link-direction
          |  |                 +--:(as-number)
          |  |                 |  +--ro as-number-hop
          |  |                 |     +--ro as-number    inet:as-number
          |  |                 |     +--ro hop-type?    te-hop-type
          |  |                 +--:(label)
          |  |                    +--ro label-hop
          |  |                       +--ro te-label
          |  |                          +--ro (technology)?
          |  |                          |  +--:(generic)
          |  |                          |     +--ro generic?
          |  |                          |             rt-
   types:generalized-label
          |  |                          +--ro direction?
          |  |                                  te-label-direction
          |  +--rw domain-id?               uint32
          |  +--rw is-abstract?             empty
          |  +--rw name?                    string
          |  +--rw signaling-address*       inet:ip-address
          |  +--rw underlay-topology {te-topology-hierarchy}?
          |     +--rw network-ref?   -> /nw:networks/network/network-id
          +--ro oper-status?                   te-types:te-oper-status
          +--ro geolocation
          |  +--ro altitude?    int64
          |  +--ro latitude?    geographic-coordinate-degree
          |  +--ro longitude?   geographic-coordinate-degree
          +--ro is-multi-access-dr?            empty
          +--ro information-source?            te-info-source
          +--ro information-source-instance?   string
          +--ro information-source-state
          |  +--ro credibility-preference?    uint16
          |  +--ro logical-network-element?   string
          |  +--ro network-instance?          string
          |  +--ro topology
          |     +--ro node-ref?      leafref
          |     +--ro network-ref?   -> /nw:networks/network/network-id
          +--ro information-source-entry*
          |       [information-source information-source-instance]
          |  +--ro information-source             te-info-source
          |  +--ro information-source-instance    string
          |  +--ro information-source-state
          |  |  +--ro credibility-preference?    uint16
          |  |  +--ro logical-network-element?   string
          |  |  +--ro network-instance?          string
          |  |  +--ro topology
          |  |     +--ro node-ref?      leafref
          |  |     +--ro network-ref?
          |  |             -> /nw:networks/network/network-id
          |  +--ro connectivity-matrices
          |  |  +--ro number-of-entries?     uint16
          |  |  +--ro label-restrictions
          |  |  |  +--ro label-restriction* [index]
          |  |  |     +--ro restriction?    enumeration
          |  |  |     +--ro index           uint32
          |  |  |     +--ro label-start
          |  |  |     |  +--ro te-label
          |  |  |     |     +--ro (technology)?
          |  |  |     |     |  +--:(generic)
          |  |  |     |     |     +--ro generic?
          |  |  |     |     |             rt-types:generalized-label
          |  |  |     |     +--ro direction?       te-label-direction
          |  |  |     +--ro label-end
          |  |  |     |  +--ro te-label
          |  |  |     |     +--ro (technology)?
          |  |  |     |     |  +--:(generic)
          |  |  |     |     |     +--ro generic?
          |  |  |     |     |             rt-types:generalized-label
          |  |  |     |     +--ro direction?       te-label-direction
          |  |  |     +--ro label-step
          |  |  |     |  +--ro (technology)?
          |  |  |     |     +--:(generic)
          |  |  |     |        +--ro generic?   int32
          |  |  |     +--ro range-bitmap?   yang:hex-string
          |  |  +--ro is-allowed?            boolean
          |  |  +--ro underlay {te-topology-hierarchy}?
          |  |  |  +--ro enabled?                     boolean
          |  |  |  +--ro primary-path
          |  |  |  |  +--ro network-ref?
          |  |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  |  +--ro path-element* [path-element-id]
          |  |  |  |     +--ro path-element-id              uint32
          |  |  |  |     +--ro (type)?
          |  |  |  |        +--:(numbered-node-hop)
          |  |  |  |        |  +--ro numbered-node-hop
          |  |  |  |        |     +--ro node-id     te-node-id
          |  |  |  |        |     +--ro hop-type?   te-hop-type
          |  |  |  |        +--:(numbered-link-hop)
          |  |  |  |        |  +--ro numbered-link-hop
          |  |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |  |        |     +--ro hop-type?     te-hop-type
          |  |  |  |        |     +--ro direction?    te-link-direction
          |  |  |  |        +--:(unnumbered-link-hop)
          |  |  |  |        |  +--ro unnumbered-link-hop
          |  |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |  |        |     +--ro node-id       te-node-id
          |  |  |  |        |     +--ro hop-type?     te-hop-type
          |  |  |  |        |     +--ro direction?    te-link-direction
          |  |  |  |        +--:(as-number)
          |  |  |  |        |  +--ro as-number-hop
          |  |  |  |        |     +--ro as-number    inet:as-number
          |  |  |  |        |     +--ro hop-type?    te-hop-type
          |  |  |  |        +--:(label)
          |  |  |  |           +--ro label-hop
          |  |  |  |              +--ro te-label
          |  |  |  |                 +--ro (technology)?
          |  |  |  |                 |  +--:(generic)
          |  |  |  |                 |     +--ro generic?
          |  |  |  |                 |             rt-types:generalized-
   label
          |  |  |  |                 +--ro direction?
          |  |  |  |                         te-label-direction
          |  |  |  +--ro backup-path* [index]
          |  |  |  |  +--ro index           uint32
          |  |  |  |  +--ro network-ref?
          |  |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  |  +--ro path-element* [path-element-id]
          |  |  |  |     +--ro path-element-id              uint32
          |  |  |  |     +--ro (type)?
          |  |  |  |        +--:(numbered-node-hop)
          |  |  |  |        |  +--ro numbered-node-hop
          |  |  |  |        |     +--ro node-id     te-node-id
          |  |  |  |        |     +--ro hop-type?   te-hop-type
          |  |  |  |        +--:(numbered-link-hop)
          |  |  |  |        |  +--ro numbered-link-hop
          |  |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |  |        |     +--ro hop-type?     te-hop-type
          |  |  |  |        |     +--ro direction?    te-link-direction
          |  |  |  |        +--:(unnumbered-link-hop)
          |  |  |  |        |  +--ro unnumbered-link-hop
          |  |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |  |        |     +--ro node-id       te-node-id
          |  |  |  |        |     +--ro hop-type?     te-hop-type
          |  |  |  |        |     +--ro direction?    te-link-direction
          |  |  |  |        +--:(as-number)
          |  |  |  |        |  +--ro as-number-hop
          |  |  |  |        |     +--ro as-number    inet:as-number
          |  |  |  |        |     +--ro hop-type?    te-hop-type
          |  |  |  |        +--:(label)
          |  |  |  |           +--ro label-hop
          |  |  |  |              +--ro te-label
          |  |  |  |                 +--ro (technology)?
          |  |  |  |                 |  +--:(generic)
          |  |  |  |                 |     +--ro generic?
          |  |  |  |                 |             rt-types:generalized-
   label
          |  |  |  |                 +--ro direction?
          |  |  |  |                         te-label-direction
          |  |  |  +--ro protection-type?             identityref
          |  |  |  +--ro tunnel-termination-points
          |  |  |  |  +--ro source?        binary
          |  |  |  |  +--ro destination?   binary
          |  |  |  +--ro tunnels
          |  |  |     +--ro sharing?   boolean
          |  |  |     +--ro tunnel* [tunnel-name]
          |  |  |        +--ro tunnel-name    string
          |  |  |        +--ro sharing?       boolean
          |  |  +--ro path-constraints
          |  |  |  +--ro te-bandwidth
          |  |  |  |  +--ro (technology)?
          |  |  |  |     +--:(generic)
          |  |  |  |        +--ro generic?   te-bandwidth
          |  |  |  +--ro link-protection?          identityref
          |  |  |  +--ro setup-priority?           uint8
          |  |  |  +--ro hold-priority?            uint8
          |  |  |  +--ro signaling-type?           identityref
          |  |  |  +--ro path-metric-bounds
          |  |  |  |  +--ro path-metric-bound* [metric-type]
          |  |  |  |     +--ro metric-type    identityref
          |  |  |  |     +--ro upper-bound?   uint64
          |  |  |  +--ro path-affinities-values
          |  |  |  |  +--ro path-affinities-value* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro value?   admin-groups
          |  |  |  +--ro path-affinity-names
          |  |  |  |  +--ro path-affinity-name* [usage]
          |  |  |  |     +--ro usage            identityref
          |  |  |  |     +--ro affinity-name* [name]
          |  |  |  |        +--ro name    string
          |  |  |  +--ro path-srlgs-lists
          |  |  |  |  +--ro path-srlgs-list* [usage]
          |  |  |  |     +--ro usage     identityref
          |  |  |  |     +--ro values*   srlg
          |  |  |  +--ro path-srlgs-names
          |  |  |  |  +--ro path-srlgs-name* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro names*   string
          |  |  |  +--ro disjointness?             te-path-disjointness
          |  |  +--ro optimizations
          |  |  |  +--ro (algorithm)?
          |  |  |     +--:(metric) {path-optimization-metric}?
          |  |  |     |  +--ro optimization-metric* [metric-type]
          |  |  |     |  |  +--ro metric-type
          |  |  |     |  |  |       identityref
          |  |  |     |  |  +--ro weight?
          |  |  |     |  |  |       uint8
          |  |  |     |  |  +--ro explicit-route-exclude-objects
          |  |  |     |  |  |  +--ro route-object-exclude-object*
          |  |  |     |  |  |          [index]
          |  |  |     |  |  |     +--ro index
          |  |  |     |  |  |     |       uint32
          |  |  |     |  |  |     +--ro (type)?
          |  |  |     |  |  |        +--:(numbered-node-hop)
          |  |  |     |  |  |        |  +--ro numbered-node-hop
          |  |  |     |  |  |        |     +--ro node-id     te-node-id
          |  |  |     |  |  |        |     +--ro hop-type?   te-hop-type
          |  |  |     |  |  |        +--:(numbered-link-hop)
          |  |  |     |  |  |        |  +--ro numbered-link-hop
          |  |  |     |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |     |  |  |        |     +--ro hop-type?
          |  |  |     |  |  |        |     |       te-hop-type
          |  |  |     |  |  |        |     +--ro direction?
          |  |  |     |  |  |        |             te-link-direction
          |  |  |     |  |  |        +--:(unnumbered-link-hop)
          |  |  |     |  |  |        |  +--ro unnumbered-link-hop
          |  |  |     |  |  |        |     +--ro link-tp-id    te-tp-id
          |  |  |     |  |  |        |     +--ro node-id
          |  |  |     |  |  |        |     |       te-node-id
          |  |  |     |  |  |        |     +--ro hop-type?
          |  |  |     |  |  |        |     |       te-hop-type
          |  |  |     |  |  |        |     +--ro direction?
          |  |  |     |  |  |        |             te-link-direction
          |  |  |     |  |  |        +--:(as-number)
          |  |  |     |  |  |        |  +--ro as-number-hop
          |  |  |     |  |  |        |     +--ro as-number
          |  |  |     |  |  |        |     |       inet:as-number
          |  |  |     |  |  |        |     +--ro hop-type?
          |  |  |     |  |  |        |             te-hop-type
          |  |  |     |  |  |        +--:(label)
          |  |  |     |  |  |        |  +--ro label-hop
          |  |  |     |  |  |        |     +--ro te-label
          |  |  |     |  |  |        |        +--ro (technology)?
          |  |  |     |  |  |        |        |  +--:(generic)
          |  |  |     |  |  |        |        |     +--ro generic?
          |  |  |     |  |  |        |        |             rt-
   types:generalized-label
          |  |  |     |  |  |        |        +--ro direction?
          |  |  |     |  |  |        |                te-label-direction
          |  |  |     |  |  |        +--:(srlg)
          |  |  |     |  |  |           +--ro srlg
          |  |  |     |  |  |              +--ro srlg?   uint32
          |  |  |     |  |  +--ro explicit-route-include-objects
          |  |  |     |  |     +--ro route-object-include-object*
          |  |  |     |  |             [index]
          |  |  |     |  |        +--ro index
          |  |  |     |  |        |       uint32
          |  |  |     |  |        +--ro (type)?
          |  |  |     |  |           +--:(numbered-node-hop)
          |  |  |     |  |           |  +--ro numbered-node-hop
          |  |  |     |  |           |     +--ro node-id     te-node-id
          |  |  |     |  |           |     +--ro hop-type?   te-hop-type
          |  |  |     |  |           +--:(numbered-link-hop)
          |  |  |     |  |           |  +--ro numbered-link-hop
          |  |  |     |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |     |  |           |     +--ro hop-type?
          |  |  |     |  |           |     |       te-hop-type
          |  |  |     |  |           |     +--ro direction?
          |  |  |     |  |           |             te-link-direction
          |  |  |     |  |           +--:(unnumbered-link-hop)
          |  |  |     |  |           |  +--ro unnumbered-link-hop
          |  |  |     |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |     |  |           |     +--ro node-id
          |  |  |     |  |           |     |       te-node-id
          |  |  |     |  |           |     +--ro hop-type?
          |  |  |     |  |           |     |       te-hop-type
          |  |  |     |  |           |     +--ro direction?
          |  |  |     |  |           |             te-link-direction
          |  |  |     |  |           +--:(as-number)
          |  |  |     |  |           |  +--ro as-number-hop
          |  |  |     |  |           |     +--ro as-number
          |  |  |     |  |           |     |       inet:as-number
          |  |  |     |  |           |     +--ro hop-type?
          |  |  |     |  |           |             te-hop-type
          |  |  |     |  |           +--:(label)
          |  |  |     |  |              +--ro label-hop
          |  |  |     |  |                 +--ro te-label
          |  |  |     |  |                    +--ro (technology)?
          |  |  |     |  |                    |  +--:(generic)
          |  |  |     |  |                    |     +--ro generic?
          |  |  |     |  |                    |             rt-
   types:generalized-label
          |  |  |     |  |                    +--ro direction?
          |  |  |     |  |                            te-label-direction
          |  |  |     |  +--ro tiebreakers
          |  |  |     |     +--ro tiebreaker* [tiebreaker-type]
          |  |  |     |        +--ro tiebreaker-type    identityref
          |  |  |     +--:(objective-function)
          |  |  |              {path-optimization-objective-function}?
          |  |  |        +--ro objective-function
          |  |  |           +--ro objective-function-type?   identityref
          |  |  +--ro path-properties
          |  |  |  +--ro path-metric* [metric-type]
          |  |  |  |  +--ro metric-type           identityref
          |  |  |  |  +--ro accumulative-value?   uint64
          |  |  |  +--ro path-affinities-values
          |  |  |  |  +--ro path-affinities-value* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro value?   admin-groups
          |  |  |  +--ro path-affinity-names
          |  |  |  |  +--ro path-affinity-name* [usage]
          |  |  |  |     +--ro usage            identityref
          |  |  |  |     +--ro affinity-name* [name]
          |  |  |  |        +--ro name    string
          |  |  |  +--ro path-srlgs-lists
          |  |  |  |  +--ro path-srlgs-list* [usage]
          |  |  |  |     +--ro usage     identityref
          |  |  |  |     +--ro values*   srlg
          |  |  |  +--ro path-srlgs-names
          |  |  |  |  +--ro path-srlgs-name* [usage]
          |  |  |  |     +--ro usage    identityref
          |  |  |  |     +--ro names*   string
          |  |  |  +--ro path-route-objects
          |  |  |     +--ro path-route-object* [index]
          |  |  |        +--ro index                        uint32
          |  |  |        +--ro (type)?
          |  |  |           +--:(numbered-node-hop)
          |  |  |           |  +--ro numbered-node-hop
          |  |  |           |     +--ro node-id     te-node-id
          |  |  |           |     +--ro hop-type?   te-hop-type
          |  |  |           +--:(numbered-link-hop)
          |  |  |           |  +--ro numbered-link-hop
          |  |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |           |     +--ro hop-type?     te-hop-type
          |  |  |           |     +--ro direction?    te-link-direction
          |  |  |           +--:(unnumbered-link-hop)
          |  |  |           |  +--ro unnumbered-link-hop
          |  |  |           |     +--ro link-tp-id    te-tp-id
          |  |  |           |     +--ro node-id       te-node-id
          |  |  |           |     +--ro hop-type?     te-hop-type
          |  |  |           |     +--ro direction?    te-link-direction
          |  |  |           +--:(as-number)
          |  |  |           |  +--ro as-number-hop
          |  |  |           |     +--ro as-number    inet:as-number
          |  |  |           |     +--ro hop-type?    te-hop-type
          |  |  |           +--:(label)
          |  |  |              +--ro label-hop
          |  |  |                 +--ro te-label
          |  |  |                    +--ro (technology)?
          |  |  |                    |  +--:(generic)
          |  |  |                    |     +--ro generic?
          |  |  |                    |             rt-types:generalized-
   label
          |  |  |                    +--ro direction?
          |  |  |                            te-label-direction
          |  |  +--ro connectivity-matrix* [id]
          |  |     +--ro id                  uint32
          |  |     +--ro from
          |  |     |  +--ro tp-ref?              leafref
          |  |     |  +--ro label-restrictions
          |  |     |     +--ro label-restriction* [index]
          |  |     |        +--ro restriction?    enumeration
          |  |     |        +--ro index           uint32
          |  |     |        +--ro label-start
          |  |     |        |  +--ro te-label
          |  |     |        |     +--ro (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--ro generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--ro direction?
          |  |     |        |             te-label-direction
          |  |     |        +--ro label-end
          |  |     |        |  +--ro te-label
          |  |     |        |     +--ro (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--ro generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--ro direction?
          |  |     |        |             te-label-direction
          |  |     |        +--ro label-step
          |  |     |        |  +--ro (technology)?
          |  |     |        |     +--:(generic)
          |  |     |        |        +--ro generic?   int32
          |  |     |        +--ro range-bitmap?   yang:hex-string
          |  |     +--ro to
          |  |     |  +--ro tp-ref?              leafref
          |  |     |  +--ro label-restrictions
          |  |     |     +--ro label-restriction* [index]
          |  |     |        +--ro restriction?    enumeration
          |  |     |        +--ro index           uint32
          |  |     |        +--ro label-start
          |  |     |        |  +--ro te-label
          |  |     |        |     +--ro (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--ro generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--ro direction?
          |  |     |        |             te-label-direction
          |  |     |        +--ro label-end
          |  |     |        |  +--ro te-label
          |  |     |        |     +--ro (technology)?
          |  |     |        |     |  +--:(generic)
          |  |     |        |     |     +--ro generic?
          |  |     |        |     |             rt-types:generalized-
   label
          |  |     |        |     +--ro direction?
          |  |     |        |             te-label-direction
          |  |     |        +--ro label-step
          |  |     |        |  +--ro (technology)?
          |  |     |        |     +--:(generic)
          |  |     |        |        +--ro generic?   int32
          |  |     |        +--ro range-bitmap?   yang:hex-string
          |  |     +--ro is-allowed?         boolean
          |  |     +--ro underlay {te-topology-hierarchy}?
          |  |     |  +--ro enabled?                     boolean
          |  |     |  +--ro primary-path
          |  |     |  |  +--ro network-ref?
          |  |     |  |  |       -> /nw:networks/network/network-id
          |  |     |  |  +--ro path-element* [path-element-id]
          |  |     |  |     +--ro path-element-id              uint32
          |  |     |  |     +--ro (type)?
          |  |     |  |        +--:(numbered-node-hop)
          |  |     |  |        |  +--ro numbered-node-hop
          |  |     |  |        |     +--ro node-id     te-node-id
          |  |     |  |        |     +--ro hop-type?   te-hop-type
          |  |     |  |        +--:(numbered-link-hop)
          |  |     |  |        |  +--ro numbered-link-hop
          |  |     |  |        |     +--ro link-tp-id    te-tp-id
          |  |     |  |        |     +--ro hop-type?     te-hop-type
          |  |     |  |        |     +--ro direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(unnumbered-link-hop)
          |  |     |  |        |  +--ro unnumbered-link-hop
          |  |     |  |        |     +--ro link-tp-id    te-tp-id
          |  |     |  |        |     +--ro node-id       te-node-id
          |  |     |  |        |     +--ro hop-type?     te-hop-type
          |  |     |  |        |     +--ro direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(as-number)
          |  |     |  |        |  +--ro as-number-hop
          |  |     |  |        |     +--ro as-number    inet:as-number
          |  |     |  |        |     +--ro hop-type?    te-hop-type
          |  |     |  |        +--:(label)
          |  |     |  |           +--ro label-hop
          |  |     |  |              +--ro te-label
          |  |     |  |                 +--ro (technology)?
          |  |     |  |                 |  +--:(generic)
          |  |     |  |                 |     +--ro generic?
          |  |     |  |                 |             rt-
   types:generalized-label
          |  |     |  |                 +--ro direction?
          |  |     |  |                         te-label-direction
          |  |     |  +--ro backup-path* [index]
          |  |     |  |  +--ro index           uint32
          |  |     |  |  +--ro network-ref?
          |  |     |  |  |       -> /nw:networks/network/network-id
          |  |     |  |  +--ro path-element* [path-element-id]
          |  |     |  |     +--ro path-element-id              uint32
          |  |     |  |     +--ro (type)?
          |  |     |  |        +--:(numbered-node-hop)
          |  |     |  |        |  +--ro numbered-node-hop
          |  |     |  |        |     +--ro node-id     te-node-id
          |  |     |  |        |     +--ro hop-type?   te-hop-type
          |  |     |  |        +--:(numbered-link-hop)
          |  |     |  |        |  +--ro numbered-link-hop
          |  |     |  |        |     +--ro link-tp-id    te-tp-id
          |  |     |  |        |     +--ro hop-type?     te-hop-type
          |  |     |  |        |     +--ro direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(unnumbered-link-hop)
          |  |     |  |        |  +--ro unnumbered-link-hop
          |  |     |  |        |     +--ro link-tp-id    te-tp-id
          |  |     |  |        |     +--ro node-id       te-node-id
          |  |     |  |        |     +--ro hop-type?     te-hop-type
          |  |     |  |        |     +--ro direction?
          |  |     |  |        |             te-link-direction
          |  |     |  |        +--:(as-number)
          |  |     |  |        |  +--ro as-number-hop
          |  |     |  |        |     +--ro as-number    inet:as-number
          |  |     |  |        |     +--ro hop-type?    te-hop-type
          |  |     |  |        +--:(label)
          |  |     |  |           +--ro label-hop
          |  |     |  |              +--ro te-label
          |  |     |  |                 +--ro (technology)?
          |  |     |  |                 |  +--:(generic)
          |  |     |  |                 |     +--ro generic?
          |  |     |  |                 |             rt-
   types:generalized-label
          |  |     |  |                 +--ro direction?
          |  |     |  |                         te-label-direction
          |  |     |  +--ro protection-type?             identityref
          |  |     |  +--ro tunnel-termination-points
          |  |     |  |  +--ro source?        binary
          |  |     |  |  +--ro destination?   binary
          |  |     |  +--ro tunnels
          |  |     |     +--ro sharing?   boolean
          |  |     |     +--ro tunnel* [tunnel-name]
          |  |     |        +--ro tunnel-name    string
          |  |     |        +--ro sharing?       boolean
          |  |     +--ro path-constraints
          |  |     |  +--ro te-bandwidth
          |  |     |  |  +--ro (technology)?
          |  |     |  |     +--:(generic)
          |  |     |  |        +--ro generic?   te-bandwidth
          |  |     |  +--ro link-protection?          identityref
          |  |     |  +--ro setup-priority?           uint8
          |  |     |  +--ro hold-priority?            uint8
          |  |     |  +--ro signaling-type?           identityref
          |  |     |  +--ro path-metric-bounds
          |  |     |  |  +--ro path-metric-bound* [metric-type]
          |  |     |  |     +--ro metric-type    identityref
          |  |     |  |     +--ro upper-bound?   uint64
          |  |     |  +--ro path-affinities-values
          |  |     |  |  +--ro path-affinities-value* [usage]
          |  |     |  |     +--ro usage    identityref
          |  |     |  |     +--ro value?   admin-groups
          |  |     |  +--ro path-affinity-names
          |  |     |  |  +--ro path-affinity-name* [usage]
          |  |     |  |     +--ro usage            identityref
          |  |     |  |     +--ro affinity-name* [name]
          |  |     |  |        +--ro name    string
          |  |     |  +--ro path-srlgs-lists
          |  |     |  |  +--ro path-srlgs-list* [usage]
          |  |     |  |     +--ro usage     identityref
          |  |     |  |     +--ro values*   srlg
          |  |     |  +--ro path-srlgs-names
          |  |     |  |  +--ro path-srlgs-name* [usage]
          |  |     |  |     +--ro usage    identityref
          |  |     |  |     +--ro names*   string
          |  |     |  +--ro disjointness?
          |  |     |          te-path-disjointness
          |  |     +--ro optimizations
          |  |     |  +--ro (algorithm)?
          |  |     |     +--:(metric) {path-optimization-metric}?
          |  |     |     |  +--ro optimization-metric* [metric-type]
          |  |     |     |  |  +--ro metric-type
          |  |     |     |  |  |       identityref
          |  |     |     |  |  +--ro weight?
          |  |     |     |  |  |       uint8
          |  |     |     |  |  +--ro explicit-route-exclude-objects
          |  |     |     |  |  |  +--ro route-object-exclude-object*
          |  |     |     |  |  |          [index]
          |  |     |     |  |  |     +--ro index
          |  |     |     |  |  |     |       uint32
          |  |     |     |  |  |     +--ro (type)?
          |  |     |     |  |  |        +--:(numbered-node-hop)
          |  |     |     |  |  |        |  +--ro numbered-node-hop
          |  |     |     |  |  |        |     +--ro node-id
          |  |     |     |  |  |        |     |       te-node-id
          |  |     |     |  |  |        |     +--ro hop-type?
          |  |     |     |  |  |        |             te-hop-type
          |  |     |     |  |  |        +--:(numbered-link-hop)
          |  |     |     |  |  |        |  +--ro numbered-link-hop
          |  |     |     |  |  |        |     +--ro link-tp-id
          |  |     |     |  |  |        |     |       te-tp-id
          |  |     |     |  |  |        |     +--ro hop-type?
          |  |     |     |  |  |        |     |       te-hop-type
          |  |     |     |  |  |        |     +--ro direction?
          |  |     |     |  |  |        |             te-link-direction
          |  |     |     |  |  |        +--:(unnumbered-link-hop)
          |  |     |     |  |  |        |  +--ro unnumbered-link-hop
          |  |     |     |  |  |        |     +--ro link-tp-id
          |  |     |     |  |  |        |     |       te-tp-id
          |  |     |     |  |  |        |     +--ro node-id
          |  |     |     |  |  |        |     |       te-node-id
          |  |     |     |  |  |        |     +--ro hop-type?
          |  |     |     |  |  |        |     |       te-hop-type
          |  |     |     |  |  |        |     +--ro direction?
          |  |     |     |  |  |        |             te-link-direction
          |  |     |     |  |  |        +--:(as-number)
          |  |     |     |  |  |        |  +--ro as-number-hop
          |  |     |     |  |  |        |     +--ro as-number
          |  |     |     |  |  |        |     |       inet:as-number
          |  |     |     |  |  |        |     +--ro hop-type?
          |  |     |     |  |  |        |             te-hop-type
          |  |     |     |  |  |        +--:(label)
          |  |     |     |  |  |        |  +--ro label-hop
          |  |     |     |  |  |        |     +--ro te-label
          |  |     |     |  |  |        |        +--ro (technology)?
          |  |     |     |  |  |        |        |  +--:(generic)
          |  |     |     |  |  |        |        |     +--ro generic?
          |  |     |     |  |  |        |        |             rt-
   types:generalized-label
          |  |     |     |  |  |        |        +--ro direction?
          |  |     |     |  |  |        |                te-label-
   direction
          |  |     |     |  |  |        +--:(srlg)
          |  |     |     |  |  |           +--ro srlg
          |  |     |     |  |  |              +--ro srlg?   uint32
          |  |     |     |  |  +--ro explicit-route-include-objects
          |  |     |     |  |     +--ro route-object-include-object*
          |  |     |     |  |             [index]
          |  |     |     |  |        +--ro index
          |  |     |     |  |        |       uint32
          |  |     |     |  |        +--ro (type)?
          |  |     |     |  |           +--:(numbered-node-hop)
          |  |     |     |  |           |  +--ro numbered-node-hop
          |  |     |     |  |           |     +--ro node-id
          |  |     |     |  |           |     |       te-node-id
          |  |     |     |  |           |     +--ro hop-type?
          |  |     |     |  |           |             te-hop-type
          |  |     |     |  |           +--:(numbered-link-hop)
          |  |     |     |  |           |  +--ro numbered-link-hop
          |  |     |     |  |           |     +--ro link-tp-id
          |  |     |     |  |           |     |       te-tp-id
          |  |     |     |  |           |     +--ro hop-type?
          |  |     |     |  |           |     |       te-hop-type
          |  |     |     |  |           |     +--ro direction?
          |  |     |     |  |           |             te-link-direction
          |  |     |     |  |           +--:(unnumbered-link-hop)
          |  |     |     |  |           |  +--ro unnumbered-link-hop
          |  |     |     |  |           |     +--ro link-tp-id
          |  |     |     |  |           |     |       te-tp-id
          |  |     |     |  |           |     +--ro node-id
          |  |     |     |  |           |     |       te-node-id
          |  |     |     |  |           |     +--ro hop-type?
          |  |     |     |  |           |     |       te-hop-type
          |  |     |     |  |           |     +--ro direction?
          |  |     |     |  |           |             te-link-direction
          |  |     |     |  |           +--:(as-number)
          |  |     |     |  |           |  +--ro as-number-hop
          |  |     |     |  |           |     +--ro as-number
          |  |     |     |  |           |     |       inet:as-number
          |  |     |     |  |           |     +--ro hop-type?
          |  |     |     |  |           |             te-hop-type
          |  |     |     |  |           +--:(label)
          |  |     |     |  |              +--ro label-hop
          |  |     |     |  |                 +--ro te-label
          |  |     |     |  |                    +--ro (technology)?
          |  |     |     |  |                    |  +--:(generic)
          |  |     |     |  |                    |     +--ro generic?
          |  |     |     |  |                    |             rt-
   types:generalized-label
          |  |     |     |  |                    +--ro direction?
          |  |     |     |  |                            te-label-
   direction
          |  |     |     |  +--ro tiebreakers
          |  |     |     |     +--ro tiebreaker* [tiebreaker-type]
          |  |     |     |        +--ro tiebreaker-type    identityref
          |  |     |     +--:(objective-function)
          |  |     |              {path-optimization-objective-
   function}?
          |  |     |        +--ro objective-function
          |  |     |           +--ro objective-function-type?
          |  |     |                   identityref
          |  |     +--ro path-properties
          |  |        +--ro path-metric* [metric-type]
          |  |        |  +--ro metric-type           identityref
          |  |        |  +--ro accumulative-value?   uint64
          |  |        +--ro path-affinities-values
          |  |        |  +--ro path-affinities-value* [usage]
          |  |        |     +--ro usage    identityref
          |  |        |     +--ro value?   admin-groups
          |  |        +--ro path-affinity-names
          |  |        |  +--ro path-affinity-name* [usage]
          |  |        |     +--ro usage            identityref
          |  |        |     +--ro affinity-name* [name]
          |  |        |        +--ro name    string
          |  |        +--ro path-srlgs-lists
          |  |        |  +--ro path-srlgs-list* [usage]
          |  |        |     +--ro usage     identityref
          |  |        |     +--ro values*   srlg
          |  |        +--ro path-srlgs-names
          |  |        |  +--ro path-srlgs-name* [usage]
          |  |        |     +--ro usage    identityref
          |  |        |     +--ro names*   string
          |  |        +--ro path-route-objects
          |  |           +--ro path-route-object* [index]
          |  |              +--ro index                        uint32
          |  |              +--ro (type)?
          |  |                 +--:(numbered-node-hop)
          |  |                 |  +--ro numbered-node-hop
          |  |                 |     +--ro node-id     te-node-id
          |  |                 |     +--ro hop-type?   te-hop-type
          |  |                 +--:(numbered-link-hop)
          |  |                 |  +--ro numbered-link-hop
          |  |                 |     +--ro link-tp-id    te-tp-id
          |  |                 |     +--ro hop-type?     te-hop-type
          |  |                 |     +--ro direction?
          |  |                 |             te-link-direction
          |  |                 +--:(unnumbered-link-hop)
          |  |                 |  +--ro unnumbered-link-hop
          |  |                 |     +--ro link-tp-id    te-tp-id
          |  |                 |     +--ro node-id       te-node-id
          |  |                 |     +--ro hop-type?     te-hop-type
          |  |                 |     +--ro direction?
          |  |                 |             te-link-direction
          |  |                 +--:(as-number)
          |  |                 |  +--ro as-number-hop
          |  |                 |     +--ro as-number    inet:as-number
          |  |                 |     +--ro hop-type?    te-hop-type
          |  |                 +--:(label)
          |  |                    +--ro label-hop
          |  |                       +--ro te-label
          |  |                          +--ro (technology)?
          |  |                          |  +--:(generic)
          |  |                          |     +--ro generic?
          |  |                          |             rt-
   types:generalized-label
          |  |                          +--ro direction?
          |  |                                  te-label-direction
          |  +--ro domain-id?                     uint32
          |  +--ro is-abstract?                   empty
          |  +--ro name?                          string
          |  +--ro signaling-address*             inet:ip-address
          |  +--ro underlay-topology {te-topology-hierarchy}?
          |     +--ro network-ref?   -> /nw:networks/network/network-id
          +--ro statistics
          |  +--ro discontinuity-time?          yang:date-and-time
          |  +--ro node
          |  |  +--ro disables?             yang:counter32
          |  |  +--ro enables?              yang:counter32
          |  |  +--ro maintenance-sets?     yang:counter32
          |  |  +--ro maintenance-clears?   yang:counter32
          |  |  +--ro modifies?             yang:counter32
          |  +--ro connectivity-matrix-entry
          |     +--ro creates?    yang:counter32
          |     +--ro deletes?    yang:counter32
          |     +--ro disables?   yang:counter32
          |     +--ro enables?    yang:counter32
          |     +--ro modifies?   yang:counter32
          +--rw tunnel-termination-point* [tunnel-tp-id]
             +--rw tunnel-tp-id                           binary
             +--rw admin-status?
             |       te-types:te-admin-status
             +--rw name?                                  string
             +--rw switching-capability?                  identityref
             +--rw encoding?                              identityref
             +--rw inter-layer-lock-id*                   uint32
             +--rw protection-type?                        identityref
             +--rw client-layer-adaptation
             |  +--rw switching-capability*
             |          [switching-capability encoding]
             |     +--rw switching-capability    identityref
             |     +--rw encoding                identityref
             |     +--rw te-bandwidth
             |        +--rw (technology)?
             |           +--:(generic)
             |              +--rw generic?   te-bandwidth
             +--rw local-link-connectivities
             |  +--rw number-of-entries?         uint16
             |  +--rw label-restrictions
             |  |  +--rw label-restriction* [index]
             |  |     +--rw restriction?    enumeration
             |  |     +--rw index           uint32
             |  |     +--rw label-start
             |  |     |  +--rw te-label
             |  |     |     +--rw (technology)?
             |  |     |     |  +--:(generic)
             |  |     |     |     +--rw generic?
             |  |     |     |             rt-types:generalized-label
             |  |     |     +--rw direction?       te-label-direction
             |  |     +--rw label-end
             |  |     |  +--rw te-label
             |  |     |     +--rw (technology)?
             |  |     |     |  +--:(generic)
             |  |     |     |     +--rw generic?
             |  |     |     |             rt-types:generalized-label
             |  |     |     +--rw direction?       te-label-direction
             |  |     +--rw label-step
             |  |     |  +--rw (technology)?
             |  |     |     +--:(generic)
             |  |     |        +--rw generic?   int32
             |  |     +--rw range-bitmap?   yang:hex-string
             |  +--rw is-allowed?                  boolean
             |  +--rw underlay {te-topology-hierarchy}?
             |  |  +--rw enabled?                       boolean
             |  |  +--rw primary-path
             |  |  |  +--rw network-ref?
             |  |  |  |       -> /nw:networks/network/network-id
             |  |  |  +--rw path-element* [path-element-id]
             |  |  |     +--rw path-element-id              uint32
             |  |  |     +--rw (type)?
             |  |  |        +--:(numbered-node-hop)
             |  |  |        |  +--rw numbered-node-hop
             |  |  |        |     +--rw node-id     te-node-id
             |  |  |        |     +--rw hop-type?   te-hop-type
             |  |  |        +--:(numbered-link-hop)
             |  |  |        |  +--rw numbered-link-hop
             |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |  |        |     +--rw hop-type?     te-hop-type
             |  |  |        |     +--rw direction?    te-link-direction
             |  |  |        +--:(unnumbered-link-hop)
             |  |  |        |  +--rw unnumbered-link-hop
             |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |  |        |     +--rw node-id       te-node-id
             |  |  |        |     +--rw hop-type?     te-hop-type
             |  |  |        |     +--rw direction?    te-link-direction
             |  |  |        +--:(as-number)
             |  |  |        |  +--rw as-number-hop
             |  |  |        |     +--rw as-number    inet:as-number
             |  |  |        |     +--rw hop-type?    te-hop-type
             |  |  |        +--:(label)
             |  |  |           +--rw label-hop
             |  |  |              +--rw te-label
             |  |  |                 +--rw (technology)?
             |  |  |                 |  +--:(generic)
             |  |  |                 |     +--rw generic?
             |  |  |                 |             rt-types:generalized-
   label
             |  |  |                 +--rw direction?
             |  |  |                         te-label-direction
             |  |  +--rw backup-path* [index]
             |  |  |  +--rw index           uint32
             |  |  |  +--rw network-ref?
             |  |  |  |       -> /nw:networks/network/network-id
             |  |  |  +--rw path-element* [path-element-id]
             |  |  |     +--rw path-element-id              uint32
             |  |  |     +--rw (type)?
             |  |  |        +--:(numbered-node-hop)
             |  |  |        |  +--rw numbered-node-hop
             |  |  |        |     +--rw node-id     te-node-id
             |  |  |        |     +--rw hop-type?   te-hop-type
             |  |  |        +--:(numbered-link-hop)
             |  |  |        |  +--rw numbered-link-hop
             |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |  |        |     +--rw hop-type?     te-hop-type
             |  |  |        |     +--rw direction?    te-link-direction
             |  |  |        +--:(unnumbered-link-hop)
             |  |  |        |  +--rw unnumbered-link-hop
             |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |  |        |     +--rw node-id       te-node-id
             |  |  |        |     +--rw hop-type?     te-hop-type
             |  |  |        |     +--rw direction?    te-link-direction
             |  |  |        +--:(as-number)
             |  |  |        |  +--rw as-number-hop
             |  |  |        |     +--rw as-number    inet:as-number
             |  |  |        |     +--rw hop-type?    te-hop-type
             |  |  |        +--:(label)
             |  |  |           +--rw label-hop
             |  |  |              +--rw te-label
             |  |  |                 +--rw (technology)?
             |  |  |                 |  +--:(generic)
             |  |  |                 |     +--rw generic?
             |  |  |                 |             rt-types:generalized-
   label
             |  |  |                 +--rw direction?
             |  |  |                         te-label-direction
             |  |  +--rw protection-type?             identityref
             |  |  +--rw tunnel-termination-points
             |  |  |  +--rw source?        binary
             |  |  |  +--rw destination?   binary
             |  |  +--rw tunnels
             |  |     +--rw sharing?   boolean
             |  |     +--rw tunnel* [tunnel-name]
             |  |        +--rw tunnel-name    string
             |  |        +--rw sharing?       boolean
             |  +--rw path-constraints
             |  |  +--rw te-bandwidth
             |  |  |  +--rw (technology)?
             |  |  |     +--:(generic)
             |  |  |        +--rw generic?   te-bandwidth
             |  |  +--rw link-protection?          identityref
             |  |  +--rw setup-priority?           uint8
             |  |  +--rw hold-priority?            uint8
             |  |  +--rw signaling-type?           identityref
             |  |  +--rw path-metric-bounds
             |  |  |  +--rw path-metric-bound* [metric-type]
             |  |  |     +--rw metric-type    identityref
             |  |  |     +--rw upper-bound?   uint64
             |  |  +--rw path-affinities-values
             |  |  |  +--rw path-affinities-value* [usage]
             |  |  |     +--rw usage    identityref
             |  |  |     +--rw value?   admin-groups
             |  |  +--rw path-affinity-names
             |  |  |  +--rw path-affinity-name* [usage]
             |  |  |     +--rw usage            identityref
             |  |  |     +--rw affinity-name* [name]
             |  |  |        +--rw name    string
             |  |  +--rw path-srlgs-lists
             |  |  |  +--rw path-srlgs-list* [usage]
             |  |  |     +--rw usage     identityref
             |  |  |     +--rw values*   srlg
             |  |  +--rw path-srlgs-names
             |  |  |  +--rw path-srlgs-name* [usage]
             |  |  |     +--rw usage    identityref
             |  |  |     +--rw names*   string
             |  |  +--rw disjointness?             te-path-disjointness
             |  +--rw optimizations
             |  |  +--rw (algorithm)?
             |  |     +--:(metric) {path-optimization-metric}?
             |  |     |  +--rw optimization-metric* [metric-type]
             |  |     |  |  +--rw metric-type
             |  |     |  |  |       identityref
             |  |     |  |  +--rw weight?
             |  |     |  |  |       uint8
             |  |     |  |  +--rw explicit-route-exclude-objects
             |  |     |  |  |  +--rw route-object-exclude-object*
             |  |     |  |  |          [index]
             |  |     |  |  |     +--rw index
             |  |     |  |  |     |       uint32
             |  |     |  |  |     +--rw (type)?
             |  |     |  |  |        +--:(numbered-node-hop)
             |  |     |  |  |        |  +--rw numbered-node-hop
             |  |     |  |  |        |     +--rw node-id     te-node-id
             |  |     |  |  |        |     +--rw hop-type?   te-hop-type
             |  |     |  |  |        +--:(numbered-link-hop)
             |  |     |  |  |        |  +--rw numbered-link-hop
             |  |     |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |     |  |  |        |     +--rw hop-type?
             |  |     |  |  |        |     |       te-hop-type
             |  |     |  |  |        |     +--rw direction?
             |  |     |  |  |        |             te-link-direction
             |  |     |  |  |        +--:(unnumbered-link-hop)
             |  |     |  |  |        |  +--rw unnumbered-link-hop
             |  |     |  |  |        |     +--rw link-tp-id    te-tp-id
             |  |     |  |  |        |     +--rw node-id
             |  |     |  |  |        |     |       te-node-id
             |  |     |  |  |        |     +--rw hop-type?
             |  |     |  |  |        |     |       te-hop-type
             |  |     |  |  |        |     +--rw direction?
             |  |     |  |  |        |             te-link-direction
             |  |     |  |  |        +--:(as-number)
             |  |     |  |  |        |  +--rw as-number-hop
             |  |     |  |  |        |     +--rw as-number
             |  |     |  |  |        |     |       inet:as-number
             |  |     |  |  |        |     +--rw hop-type?
             |  |     |  |  |        |             te-hop-type
             |  |     |  |  |        +--:(label)
             |  |     |  |  |        |  +--rw label-hop
             |  |     |  |  |        |     +--rw te-label
             |  |     |  |  |        |        +--rw (technology)?
             |  |     |  |  |        |        |  +--:(generic)
             |  |     |  |  |        |        |     +--rw generic?
             |  |     |  |  |        |        |             rt-
   types:generalized-label
             |  |     |  |  |        |        +--rw direction?
             |  |     |  |  |        |                te-label-direction
             |  |     |  |  |        +--:(srlg)
             |  |     |  |  |           +--rw srlg
             |  |     |  |  |              +--rw srlg?   uint32
             |  |     |  |  +--rw explicit-route-include-objects
             |  |     |  |     +--rw route-object-include-object*
             |  |     |  |             [index]
             |  |     |  |        +--rw index
             |  |     |  |        |       uint32
             |  |     |  |        +--rw (type)?
             |  |     |  |           +--:(numbered-node-hop)
             |  |     |  |           |  +--rw numbered-node-hop
             |  |     |  |           |     +--rw node-id     te-node-id
             |  |     |  |           |     +--rw hop-type?   te-hop-type
             |  |     |  |           +--:(numbered-link-hop)
             |  |     |  |           |  +--rw numbered-link-hop
             |  |     |  |           |     +--rw link-tp-id    te-tp-id
             |  |     |  |           |     +--rw hop-type?
             |  |     |  |           |     |       te-hop-type
             |  |     |  |           |     +--rw direction?
             |  |     |  |           |             te-link-direction
             |  |     |  |           +--:(unnumbered-link-hop)
             |  |     |  |           |  +--rw unnumbered-link-hop
             |  |     |  |           |     +--rw link-tp-id    te-tp-id
             |  |     |  |           |     +--rw node-id
             |  |     |  |           |     |       te-node-id
             |  |     |  |           |     +--rw hop-type?
             |  |     |  |           |     |       te-hop-type
             |  |     |  |           |     +--rw direction?
             |  |     |  |           |             te-link-direction
             |  |     |  |           +--:(as-number)
             |  |     |  |           |  +--rw as-number-hop
             |  |     |  |           |     +--rw as-number
             |  |     |  |           |     |       inet:as-number
             |  |     |  |           |     +--rw hop-type?
             |  |     |  |           |             te-hop-type
             |  |     |  |           +--:(label)
             |  |     |  |              +--rw label-hop
             |  |     |  |                 +--rw te-label
             |  |     |  |                    +--rw (technology)?
             |  |     |  |                    |  +--:(generic)
             |  |     |  |                    |     +--rw generic?
             |  |     |  |                    |             rt-
   types:generalized-label
             |  |     |  |                    +--rw direction?
             |  |     |  |                            te-label-direction
             |  |     |  +--rw tiebreakers
             |  |     |     +--rw tiebreaker* [tiebreaker-type]
             |  |     |        +--rw tiebreaker-type    identityref
             |  |     +--:(objective-function)
             |  |              {path-optimization-objective-function}?
             |  |        +--rw objective-function
             |  |           +--rw objective-function-type?   identityref
             |  +--ro path-properties
             |  |  +--ro path-metric* [metric-type]
             |  |  |  +--ro metric-type           identityref
             |  |  |  +--ro accumulative-value?   uint64
             |  |  +--ro path-affinities-values
             |  |  |  +--ro path-affinities-value* [usage]
             |  |  |     +--ro usage    identityref
             |  |  |     +--ro value?   admin-groups
             |  |  +--ro path-affinity-names
             |  |  |  +--ro path-affinity-name* [usage]
             |  |  |     +--ro usage            identityref
             |  |  |     +--ro affinity-name* [name]
             |  |  |        +--ro name    string
             |  |  +--ro path-srlgs-lists
             |  |  |  +--ro path-srlgs-list* [usage]
             |  |  |     +--ro usage     identityref
             |  |  |     +--ro values*   srlg
             |  |  +--ro path-srlgs-names
             |  |  |  +--ro path-srlgs-name* [usage]
             |  |  |     +--ro usage    identityref
             |  |  |     +--ro names*   string
             |  |  +--ro path-route-objects
             |  |     +--ro path-route-object* [index]
             |  |        +--ro index                        uint32
             |  |        +--ro (type)?
             |  |           +--:(numbered-node-hop)
             |  |           |  +--ro numbered-node-hop
             |  |           |     +--ro node-id     te-node-id
             |  |           |     +--ro hop-type?   te-hop-type
             |  |           +--:(numbered-link-hop)
             |  |           |  +--ro numbered-link-hop
             |  |           |     +--ro link-tp-id    te-tp-id
             |  |           |     +--ro hop-type?     te-hop-type
             |  |           |     +--ro direction?    te-link-direction
             |  |           +--:(unnumbered-link-hop)
             |  |           |  +--ro unnumbered-link-hop
             |  |           |     +--ro link-tp-id    te-tp-id
             |  |           |     +--ro node-id       te-node-id
             |  |           |     +--ro hop-type?     te-hop-type
             |  |           |     +--ro direction?    te-link-direction
             |  |           +--:(as-number)
             |  |           |  +--ro as-number-hop
             |  |           |     +--ro as-number    inet:as-number
             |  |           |     +--ro hop-type?    te-hop-type
             |  |           +--:(label)
             |  |              +--ro label-hop
             |  |                 +--ro te-label
             |  |                    +--ro (technology)?
             |  |                    |  +--:(generic)
             |  |                    |     +--ro generic?
             |  |                    |             rt-types:generalized-
   label
             |  |                    +--ro direction?
             |  |                            te-label-direction
             |  +--rw local-link-connectivity* [link-tp-ref]
             |     +--rw link-tp-ref
             |     |       -> ../../../../../nt:termination-point/tp-id
             |     +--rw label-restrictions
             |     |  +--rw label-restriction* [index]
             |     |     +--rw restriction?    enumeration
             |     |     +--rw index           uint32
             |     |     +--rw label-start
             |     |     |  +--rw te-label
             |     |     |     +--rw (technology)?
             |     |     |     |  +--:(generic)
             |     |     |     |     +--rw generic?
             |     |     |     |             rt-types:generalized-label
             |     |     |     +--rw direction?       te-label-direction
             |     |     +--rw label-end
             |     |     |  +--rw te-label
             |     |     |     +--rw (technology)?
             |     |     |     |  +--:(generic)
             |     |     |     |     +--rw generic?
             |     |     |     |             rt-types:generalized-label
             |     |     |     +--rw direction?       te-label-direction
             |     |     +--rw label-step
             |     |     |  +--rw (technology)?
             |     |     |     +--:(generic)
             |     |     |        +--rw generic?   int32
             |     |     +--rw range-bitmap?   yang:hex-string
             |     +--rw is-allowed?          boolean
             |     +--rw underlay {te-topology-hierarchy}?
             |     |  +--rw enabled?                     boolean
             |     |  +--rw primary-path
             |     |  |  +--rw network-ref?
             |     |  |  |       -> /nw:networks/network/network-id
             |     |  |  +--rw path-element* [path-element-id]
             |     |  |     +--rw path-element-id              uint32
             |     |  |     +--rw (type)?
             |     |  |        +--:(numbered-node-hop)
             |     |  |        |  +--rw numbered-node-hop
             |     |  |        |     +--rw node-id     te-node-id
             |     |  |        |     +--rw hop-type?   te-hop-type
             |     |  |        +--:(numbered-link-hop)
             |     |  |        |  +--rw numbered-link-hop
             |     |  |        |     +--rw link-tp-id    te-tp-id
             |     |  |        |     +--rw hop-type?     te-hop-type
             |     |  |        |     +--rw direction?
             |     |  |        |             te-link-direction
             |     |  |        +--:(unnumbered-link-hop)
             |     |  |        |  +--rw unnumbered-link-hop
             |     |  |        |     +--rw link-tp-id    te-tp-id
             |     |  |        |     +--rw node-id       te-node-id
             |     |  |        |     +--rw hop-type?     te-hop-type
             |     |  |        |     +--rw direction?
             |     |  |        |             te-link-direction
             |     |  |        +--:(as-number)
             |     |  |        |  +--rw as-number-hop
             |     |  |        |     +--rw as-number    inet:as-number
             |     |  |        |     +--rw hop-type?    te-hop-type
             |     |  |        +--:(label)
             |     |  |           +--rw label-hop
             |     |  |              +--rw te-label
             |     |  |                 +--rw (technology)?
             |     |  |                 |  +--:(generic)
             |     |  |                 |     +--rw generic?
             |     |  |                 |             rt-
   types:generalized-label
```

```
             |     |  |                 +--rw direction?
             |     |  |                         te-label-direction
             |     |  +--rw backup-path* [index]
             |     |  |  +--rw index           uint32
             |     |  |  +--rw network-ref?
             |     |  |  |       -> /nw:networks/network/network-id
             |     |  |  +--rw path-element* [path-element-id]
             |     |  |     +--rw path-element-id              uint32
             |     |  |     +--rw (type)?
             |     |  |        +--:(numbered-node-hop)
             |     |  |        |  +--rw numbered-node-hop
             |     |  |        |     +--rw node-id     te-node-id
             |     |  |        |     +--rw hop-type?   te-hop-type
             |     |  |        +--:(numbered-link-hop)
             |     |  |        |  +--rw numbered-link-hop
             |     |  |        |     +--rw link-tp-id    te-tp-id
             |     |  |        |     +--rw hop-type?     te-hop-type
             |     |  |        |     +--rw direction?
             |     |  |        |             te-link-direction
             |     |  |        +--:(unnumbered-link-hop)
             |     |  |        |  +--rw unnumbered-link-hop
             |     |  |        |     +--rw link-tp-id    te-tp-id
             |     |  |        |     +--rw node-id       te-node-id
             |     |  |        |     +--rw hop-type?     te-hop-type
             |     |  |        |     +--rw direction?
             |     |  |        |             te-link-direction
             |     |  |        +--:(as-number)
             |     |  |        |  +--rw as-number-hop
             |     |  |        |     +--rw as-number    inet:as-number
             |     |  |        |     +--rw hop-type?    te-hop-type
             |     |  |        +--:(label)
             |     |  |           +--rw label-hop
             |     |  |              +--rw te-label
             |     |  |                 +--rw (technology)?
             |     |  |                 |  +--:(generic)
             |     |  |                 |     +--rw generic?
             |     |  |                 |             rt-
   types:generalized-label
             |     |  |                 +--rw direction?
             |     |  |                         te-label-direction
             |     |  +--rw protection-type?            identityref
             |     |  +--rw tunnel-termination-points
             |     |  |  +--rw source?        binary
             |     |  |  +--rw destination?   binary
             |     |  +--rw tunnels
             |     |     +--rw sharing?   boolean
             |     |     +--rw tunnel* [tunnel-name]
             |     |        +--rw tunnel-name    string
             |     |        +--rw sharing?       boolean
             |     +--rw path-constraints
             |     |  +--rw te-bandwidth
             |     |  |  +--rw (technology)?
             |     |  |     +--:(generic)
             |     |  |        +--rw generic?   te-bandwidth
             |     |  +--rw link-protection?          identityref
             |     |  +--rw setup-priority?           uint8
             |     |  +--rw hold-priority?            uint8
             |     |  +--rw signaling-type?           identityref
             |     |  +--rw path-metric-bounds
             |     |  |  +--rw path-metric-bound* [metric-type]
             |     |  |     +--rw metric-type    identityref
             |     |  |     +--rw upper-bound?   uint64
             |     |  +--rw path-affinities-values
             |     |  |  +--rw path-affinities-value* [usage]
             |     |  |     +--rw usage    identityref
             |     |  |     +--rw value?   admin-groups
             |     |  +--rw path-affinity-names
             |     |  |  +--rw path-affinity-name* [usage]
             |     |  |     +--rw usage            identityref
             |     |  |     +--rw affinity-name* [name]
             |     |  |        +--rw name    string
             |     |  +--rw path-srlgs-lists
             |     |  |  +--rw path-srlgs-list* [usage]
             |     |  |     +--rw usage     identityref
             |     |  |     +--rw values*   srlg
             |     |  +--rw path-srlgs-names
             |     |  |  +--rw path-srlgs-name* [usage]
             |     |  |     +--rw usage    identityref
             |     |  |     +--rw names*   string
             |     |  +--rw disjointness?
             |     |          te-path-disjointness
             |     +--rw optimizations
             |     |  +--rw (algorithm)?
             |     |     +--:(metric) {path-optimization-metric}?
             |     |     |  +--rw optimization-metric* [metric-type]
             |     |     |  |  +--rw metric-type
             |     |     |  |  |       identityref
             |     |     |  |  +--rw weight?
             |     |     |  |  |       uint8
             |     |     |  |  +--rw explicit-route-exclude-objects
             |     |     |  |  |  +--rw route-object-exclude-object*
             |     |     |  |  |          [index]
             |     |     |  |  |     +--rw index
             |     |     |  |  |     |       uint32
             |     |     |  |  |     +--rw (type)?
             |     |     |  |  |        +--:(numbered-node-hop)
             |     |     |  |  |        |  +--rw numbered-node-hop
             |     |     |  |  |        |     +--rw node-id
             |     |     |  |  |        |     |       te-node-id
             |     |     |  |  |        |     +--rw hop-type?
             |     |     |  |  |        |             te-hop-type
             |     |     |  |  |        +--:(numbered-link-hop)
             |     |     |  |  |        |  +--rw numbered-link-hop
             |     |     |  |  |        |     +--rw link-tp-id
             |     |     |  |  |        |     |       te-tp-id
             |     |     |  |  |        |     +--rw hop-type?
             |     |     |  |  |        |     |       te-hop-type
             |     |     |  |  |        |     +--rw direction?
             |     |     |  |  |        |             te-link-direction
             |     |     |  |  |        +--:(unnumbered-link-hop)
             |     |     |  |  |        |  +--rw unnumbered-link-hop
             |     |     |  |  |        |     +--rw link-tp-id
             |     |     |  |  |        |     |       te-tp-id
             |     |     |  |  |        |     +--rw node-id
             |     |     |  |  |        |     |       te-node-id
             |     |     |  |  |        |     +--rw hop-type?
             |     |     |  |  |        |     |       te-hop-type
             |     |     |  |  |        |     +--rw direction?
             |     |     |  |  |        |             te-link-direction
             |     |     |  |  |        +--:(as-number)
             |     |     |  |  |        |  +--rw as-number-hop
             |     |     |  |  |        |     +--rw as-number
             |     |     |  |  |        |     |       inet:as-number
             |     |     |  |  |        |     +--rw hop-type?
             |     |     |  |  |        |             te-hop-type
             |     |     |  |  |        +--:(label)
             |     |     |  |  |        |  +--rw label-hop
             |     |     |  |  |        |     +--rw te-label
             |     |     |  |  |        |        +--rw (technology)?
             |     |     |  |  |        |        |  +--:(generic)
             |     |     |  |  |        |        |     +--rw generic?
             |     |     |  |  |        |        |             rt-
   types:generalized-label
             |     |     |  |  |        |        +--rw direction?
             |     |     |  |  |        |                te-label-
   direction
             |     |     |  |  |        +--:(srlg)
             |     |     |  |  |           +--rw srlg
             |     |     |  |  |              +--rw srlg?   uint32
             |     |     |  |  +--rw explicit-route-include-objects
             |     |     |  |     +--rw route-object-include-object*
             |     |     |  |             [index]
             |     |     |  |        +--rw index
             |     |     |  |        |       uint32
             |     |     |  |        +--rw (type)?
             |     |     |  |           +--:(numbered-node-hop)
             |     |     |  |           |  +--rw numbered-node-hop
             |     |     |  |           |     +--rw node-id
             |     |     |  |           |     |       te-node-id
             |     |     |  |           |     +--rw hop-type?
             |     |     |  |           |             te-hop-type
             |     |     |  |           +--:(numbered-link-hop)
             |     |     |  |           |  +--rw numbered-link-hop
             |     |     |  |           |     +--rw link-tp-id
             |     |     |  |           |     |       te-tp-id
             |     |     |  |           |     +--rw hop-type?
             |     |     |  |           |     |       te-hop-type
             |     |     |  |           |     +--rw direction?
             |     |     |  |           |             te-link-direction
             |     |     |  |           +--:(unnumbered-link-hop)
             |     |     |  |           |  +--rw unnumbered-link-hop
             |     |     |  |           |     +--rw link-tp-id
             |     |     |  |           |     |       te-tp-id
             |     |     |  |           |     +--rw node-id
             |     |     |  |           |     |       te-node-id
             |     |     |  |           |     +--rw hop-type?
             |     |     |  |           |     |       te-hop-type
             |     |     |  |           |     +--rw direction?
             |     |     |  |           |             te-link-direction
             |     |     |  |           +--:(as-number)
             |     |     |  |           |  +--rw as-number-hop
             |     |     |  |           |     +--rw as-number
             |     |     |  |           |     |       inet:as-number
             |     |     |  |           |     +--rw hop-type?
             |     |     |  |           |             te-hop-type
             |     |     |  |           +--:(label)
             |     |     |  |              +--rw label-hop
             |     |     |  |                 +--rw te-label
             |     |     |  |                    +--rw (technology)?
             |     |     |  |                    |  +--:(generic)
             |     |     |  |                    |     +--rw generic?
             |     |     |  |                    |             rt-
   types:generalized-label
             |     |     |  |                    +--rw direction?
             |     |     |  |                            te-label-
   direction
             |     |     |  +--rw tiebreakers
             |     |     |     +--rw tiebreaker* [tiebreaker-type]
             |     |     |        +--rw tiebreaker-type    identityref
             |     |     +--:(objective-function)
             |     |              {path-optimization-objective-
   function}?
             |     |        +--rw objective-function
             |     |           +--rw objective-function-type?
             |     |                   identityref
             |     +--ro path-properties
             |        +--ro path-metric* [metric-type]
             |        |  +--ro metric-type           identityref
             |        |  +--ro accumulative-value?   uint64
             |        +--ro path-affinities-values
             |        |  +--ro path-affinities-value* [usage]
             |        |     +--ro usage    identityref
             |        |     +--ro value?   admin-groups
             |        +--ro path-affinity-names
             |        |  +--ro path-affinity-name* [usage]
             |        |     +--ro usage            identityref
             |        |     +--ro affinity-name* [name]
             |        |        +--ro name    string
             |        +--ro path-srlgs-lists
             |        |  +--ro path-srlgs-list* [usage]
             |        |     +--ro usage     identityref
             |        |     +--ro values*   srlg
             |        +--ro path-srlgs-names
             |        |  +--ro path-srlgs-name* [usage]
             |        |     +--ro usage    identityref
             |        |     +--ro names*   string
             |        +--ro path-route-objects
             |           +--ro path-route-object* [index]
             |              +--ro index                        uint32
             |              +--ro (type)?
             |                 +--:(numbered-node-hop)
             |                 |  +--ro numbered-node-hop
             |                 |     +--ro node-id     te-node-id
             |                 |     +--ro hop-type?   te-hop-type
             |                 +--:(numbered-link-hop)
             |                 |  +--ro numbered-link-hop
             |                 |     +--ro link-tp-id    te-tp-id
             |                 |     +--ro hop-type?     te-hop-type
             |                 |     +--ro direction?
             |                 |             te-link-direction
             |                 +--:(unnumbered-link-hop)
             |                 |  +--ro unnumbered-link-hop
             |                 |     +--ro link-tp-id    te-tp-id
             |                 |     +--ro node-id       te-node-id
             |                 |     +--ro hop-type?     te-hop-type
             |                 |     +--ro direction?
             |                 |             te-link-direction
             |                 +--:(as-number)
             |                 |  +--ro as-number-hop
             |                 |     +--ro as-number    inet:as-number
             |                 |     +--ro hop-type?    te-hop-type
             |                 +--:(label)
             |                    +--ro label-hop
             |                       +--ro te-label
             |                          +--ro (technology)?
             |                          |  +--:(generic)
             |                           |     +--ro generic?
             |                           |             rt-
   types:generalized-label
             |                           +--ro direction?
             |                                 te-label-direction
             +--ro oper-status?
             |       te-types:te-oper-status
             +--ro geolocation
             |  +--ro altitude?    int64
             |  +--ro latitude?    geographic-coordinate-degree
             |  +--ro longitude?   geographic-coordinate-degree
             +--ro statistics
             |  +--ro discontinuity-time?         yang:date-and-time
             |  +--ro tunnel-termination-point
             |  |  +--ro disables?             yang:counter32
             |  |  +--ro enables?              yang:counter32
             |  |  +--ro maintenance-clears?   yang:counter32
             |  |  +--ro maintenance-sets?     yang:counter32
             |  |  +--ro modifies?             yang:counter32
             |  |  +--ro downs?                yang:counter32
             |  |  +--ro ups?                  yang:counter32
             |  |  +--ro in-service-clears?    yang:counter32
             |  |  +--ro in-service-sets?      yang:counter32
             |  +--ro local-link-connectivity
             |     +--ro creates?    yang:counter32
             |     +--ro deletes?    yang:counter32
             |     +--ro disables?   yang:counter32
             |     +--ro enables?    yang:counter32
             |     +--ro modifies?   yang:counter32
             +--rw supporting-tunnel-termination-point*
                     [node-ref tunnel-tp-ref]
                +--rw node-ref         inet:uri
                +--rw tunnel-tp-ref    binary
     augment /nw:networks/nw:network/nt:link:
       +--rw te!
          +--rw (bundle-stack-level)?
          |  +--:(bundle)
          |  |  +--rw bundled-links
          |  |     +--rw bundled-link* [sequence]
          |  |        +--rw sequence      uint32
          |  |        +--rw src-tp-ref?   leafref
          |  |        +--rw des-tp-ref?   leafref
          |  +--:(component)
          |     +--rw component-links
          |        +--rw component-link* [sequence]
          |           +--rw sequence              uint32
          |           +--rw src-interface-ref?   string
          |           +--rw des-interface-ref?   string
          +--rw te-link-template*
          |       -> ../../../../te/templates/link-template/name
          |       {template}?
          +--rw te-link-attributes
          |  +--rw access-type?
          |  |       te-types:te-link-access-type
          |  +--rw external-domain
          |  |  +--rw network-ref?
          |  |  |       -> /nw:networks/network/network-id
          |  |  +--rw remote-te-node-id?      te-types:te-node-id
          |  |  +--rw remote-te-link-tp-id?   te-types:te-tp-id
          |  +--rw is-abstract?                      empty
          |  +--rw name?                             string
          |  +--rw underlay {te-topology-hierarchy}?
          |  |  +--rw enabled?                     boolean
          |  |  +--rw primary-path
          |  |  |  +--rw network-ref?
          |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  +--rw path-element* [path-element-id]
          |  |  |     +--rw path-element-id              uint32
          |  |  |     +--rw (type)?
          |  |  |        +--:(numbered-node-hop)
          |  |  |        |  +--rw numbered-node-hop
          |  |  |        |     +--rw node-id     te-node-id
          |  |  |        |     +--rw hop-type?   te-hop-type
          |  |  |        +--:(numbered-link-hop)
          |  |  |        |  +--rw numbered-link-hop
          |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |        |     +--rw direction?    te-link-direction
          |  |  |        +--:(unnumbered-link-hop)
          |  |  |        |  +--rw unnumbered-link-hop
          |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |        |     +--rw node-id       te-node-id
          |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |        |     +--rw direction?    te-link-direction
          |  |  |        +--:(as-number)
          |  |  |        |  +--rw as-number-hop
          |  |  |        |     +--rw as-number    inet:as-number
          |  |  |        |     +--rw hop-type?    te-hop-type
          |  |  |        +--:(label)
          |  |  |           +--rw label-hop
          |  |  |              +--rw te-label
          |  |  |                 +--rw (technology)?
          |  |  |                 |  +--:(generic)
          |  |  |                 |     +--rw generic?
          |  |  |                 |             rt-types:generalized-
   label
          |  |  |                 +--rw direction?
          |  |  |                         te-label-direction
          |  |  +--rw backup-path* [index]
          |  |  |  +--rw index           uint32
          |  |  |  +--rw network-ref?
          |  |  |  |       -> /nw:networks/network/network-id
          |  |  |  +--rw path-element* [path-element-id]
          |  |  |     +--rw path-element-id              uint32
          |  |  |     +--rw (type)?
          |  |  |        +--:(numbered-node-hop)
          |  |  |        |  +--rw numbered-node-hop
          |  |  |        |     +--rw node-id     te-node-id
          |  |  |        |     +--rw hop-type?   te-hop-type
          |  |  |        +--:(numbered-link-hop)
          |  |  |        |  +--rw numbered-link-hop
          |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |        |     +--rw direction?    te-link-direction
          |  |  |        +--:(unnumbered-link-hop)
          |  |  |        |  +--rw unnumbered-link-hop
          |  |  |        |     +--rw link-tp-id    te-tp-id
          |  |  |        |     +--rw node-id       te-node-id
          |  |  |        |     +--rw hop-type?     te-hop-type
          |  |  |        |     +--rw direction?    te-link-direction
          |  |  |        +--:(as-number)
          |  |  |        |  +--rw as-number-hop
          |  |  |        |     +--rw as-number    inet:as-number
          |  |  |        |     +--rw hop-type?    te-hop-type
          |  |  |        +--:(label)
          |  |  |           +--rw label-hop
          |  |  |              +--rw te-label
          |  |  |                 +--rw (technology)?
          |  |  |                 |  +--:(generic)
          |  |  |                 |     +--rw generic?
          |  |  |                 |             rt-types:generalized-
   label
          |  |  |                 +--rw direction?
          |  |  |                         te-label-direction
          |  |  +--rw protection-type?             identityref
          |  |  +--rw tunnel-termination-points
          |  |  |  +--rw source?        binary
          |  |  |  +--rw destination?   binary
          |  |  +--rw tunnels
          |  |     +--rw sharing?   boolean
          |  |     +--rw tunnel* [tunnel-name]
          |  |        +--rw tunnel-name    string
          |  |        +--rw sharing?       boolean
          |  +--rw admin-status?
          |  |       te-types:te-admin-status
          |  +--rw link-index?                       uint64
          |  +--rw administrative-group?
          |  |       te-types:admin-groups
          |  +--rw interface-switching-capability*
          |  |       [switching-capability encoding]
          |  |  +--rw switching-capability    identityref
          |  |  +--rw encoding                identityref
          |  |  +--rw max-lsp-bandwidth* [priority]
          |  |     +--rw priority        uint8
          |  |     +--rw te-bandwidth
          |  |        +--rw (technology)?
          |  |           +--:(generic)
          |  |              +--rw generic?   te-bandwidth
          |  +--rw label-restrictions
          |  |  +--rw label-restriction* [index]
          |  |     +--rw restriction?    enumeration
          |  |     +--rw index           uint32
          |  |     +--rw label-start
          |  |     |  +--rw te-label
          |  |     |     +--rw (technology)?
          |  |     |     |  +--:(generic)
          |  |     |     |     +--rw generic?
          |  |     |     |             rt-types:generalized-label
          |  |     |     +--rw direction?       te-label-direction
          |  |     +--rw label-end
          |  |     |  +--rw te-label
          |  |     |     +--rw (technology)?
          |  |     |     |  +--:(generic)
          |  |     |     |     +--rw generic?
          |  |     |     |             rt-types:generalized-label
          |  |     |     +--rw direction?       te-label-direction
          |  |     +--rw label-step
          |  |     |  +--rw (technology)?
          |  |     |     +--:(generic)
          |  |     |        +--rw generic?   int32
          |  |     +--rw range-bitmap?   yang:hex-string
          |  +--rw link-protection-type?              identityref
          |  +--rw max-link-bandwidth
          |  |  +--rw te-bandwidth
          |  |     +--rw (technology)?
          |  |        +--:(generic)
          |  |           +--rw generic?   te-bandwidth
          |  +--rw max-resv-link-bandwidth
          |  |  +--rw te-bandwidth
          |  |     +--rw (technology)?
          |  |        +--:(generic)
          |  |           +--rw generic?   te-bandwidth
          |  +--rw unreserved-bandwidth* [priority]
          |  |  +--rw priority        uint8
          |  |  +--rw te-bandwidth
          |  |     +--rw (technology)?
          |  |        +--:(generic)
          |  |           +--rw generic?   te-bandwidth
          |  +--rw te-default-metric?                 uint32
          |  +--rw te-delay-metric?                   uint32
          |  +--rw te-igp-metric?                     uint32
          |  +--rw te-srlgs
          |  |  +--rw value*   te-types:srlg
          |  +--rw te-nsrlgs {nsrlg}?
          |     +--rw id*   uint32
          +--ro oper-status?                  te-types:te-oper-status
          +--ro is-transitional?              empty
          +--ro information-source?           te-info-source
          +--ro information-source-instance?   string
          +--ro information-source-state
          |  +--ro credibility-preference?    uint16
          |  +--ro logical-network-element?   string
          |  +--ro network-instance?          string
          |  +--ro topology
          |     +--ro link-ref?      leafref
          |     +--ro network-ref?   -> /nw:networks/network/network-id
          +--ro information-source-entry*
          |       [information-source information-source-instance]
          |  +--ro information-source               te-info-source
          |  +--ro information-source-instance       string
          |  +--ro information-source-state
          |  |  +--ro credibility-preference?    uint16
          |  |  +--ro logical-network-element?   string
          |  |  +--ro network-instance?          string
          |  |  +--ro topology
          |  |     +--ro link-ref?      leafref
          |  |     +--ro network-ref?
          |  |             -> /nw:networks/network/network-id
          |  +--ro link-index?                       uint64
          |  +--ro administrative-group?
          |  |       te-types:admin-groups
          |  +--ro interface-switching-capability*
          |  |       [switching-capability encoding]
          |  |  +--ro switching-capability    identityref
          |  |  +--ro encoding                identityref
          |  |  +--ro max-lsp-bandwidth* [priority]
          |  |     +--ro priority        uint8
          |  |     +--ro te-bandwidth
          |  |        +--ro (technology)?
          |  |           +--:(generic)
          |  |              +--ro generic?   te-bandwidth
          |  +--ro label-restrictions
          |  |  +--ro label-restriction* [index]
          |  |     +--ro restriction?    enumeration
          |  |     +--ro index           uint32
          |  |     +--ro label-start
          |  |     |  +--ro te-label
          |  |     |     +--ro (technology)?
          |  |     |     |  +--:(generic)
          |  |     |     |     +--ro generic?
          |  |     |     |             rt-types:generalized-label
          |  |     |     +--ro direction?       te-label-direction
          |  |     +--ro label-end
          |  |     |  +--ro te-label
          |  |     |     +--ro (technology)?
          |  |     |     |  +--:(generic)
          |  |     |     |     +--ro generic?
          |  |     |     |             rt-types:generalized-label
          |  |     |     +--ro direction?       te-label-direction
          |  |     +--ro label-step
          |  |     |  +--ro (technology)?
          |  |     |     +--:(generic)
          |  |     |        +--ro generic?   int32
          |  |     +--ro range-bitmap?   yang:hex-string
          |  +--ro link-protection-type?             identityref
          |  +--ro max-link-bandwidth
          |  |  +--ro te-bandwidth
          |  |     +--ro (technology)?
          |  |        +--:(generic)
          |  |           +--ro generic?   te-bandwidth
          |  +--ro max-resv-link-bandwidth
          |  |  +--ro te-bandwidth
          |  |     +--ro (technology)?
          |  |        +--:(generic)
          |  |           +--ro generic?   te-bandwidth
          |  +--ro unreserved-bandwidth* [priority]
          |  |  +--ro priority       uint8
          |  |  +--ro te-bandwidth
          |  |     +--ro (technology)?
          |  |        +--:(generic)
          |  |           +--ro generic?   te-bandwidth
          |  +--ro te-default-metric?                uint32
          |  +--ro te-delay-metric?                  uint32
          |  +--ro te-igp-metric?                    uint32
          |  +--ro te-srlgs
          |  |  +--ro value*   te-types:srlg
          |  +--ro te-nsrlgs {nsrlg}?
          |     +--ro id*   uint32
          +--ro recovery
          |  +--ro restoration-status?   te-types:te-recovery-status
          |  +--ro protection-status?    te-types:te-recovery-status
          +--ro underlay {te-topology-hierarchy}?
          |  +--ro dynamic?     boolean
          |  +--ro committed?   boolean
          +--ro statistics
             +--ro discontinuity-time?                yang:date-and-time
             +--ro disables?                          yang:counter32
             +--ro enables?                           yang:counter32
             +--ro maintenance-clears?                yang:counter32
             +--ro maintenance-sets?                  yang:counter32
             +--ro modifies?                          yang:counter32
             +--ro downs?                             yang:counter32
             +--ro ups?                               yang:counter32
             +--ro fault-clears?                      yang:counter32
             +--ro fault-detects?                     yang:counter32
             +--ro protection-switches?               yang:counter32
             +--ro protection-reverts?                yang:counter32
             +--ro restoration-failures?              yang:counter32
             +--ro restoration-starts?                yang:counter32
             +--ro restoration-successes?             yang:counter32
             +--ro restoration-reversion-failures?    yang:counter32
             +--ro restoration-reversion-starts?      yang:counter32
             +--ro restoration-reversion-successes?   yang:counter32
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--rw te-tp-id?   te-types:te-tp-id
       +--rw te!
          +--rw admin-status?
          |       te-types:te-admin-status
          +--rw name?                             string
          +--rw interface-switching-capability*
          |       [switching-capability encoding]
          |  +--rw switching-capability    identityref
          |  +--rw encoding                identityref
          |  +--rw max-lsp-bandwidth* [priority]
          |     +--rw priority        uint8
          |     +--rw te-bandwidth
          |        +--rw (technology)?
          |           +--:(generic)
          |              +--rw generic?   te-bandwidth
          +--rw inter-domain-plug-id?            binary
          +--rw inter-layer-lock-id*             uint32
          +--ro oper-status?
          |       te-types:te-oper-status
          +--ro geolocation
             +--ro altitude?    int64
             +--ro latitude?    geographic-coordinate-degree
             +--ro longitude?   geographic-coordinate-degree
```

Appendix B. Companion YANG Model for Non-NMDA Compliant Implementations

The YANG module ietf-te-topology defined in this document is designed to be used in conjunction with implementations that support the Network Management Datastore Architecture (NMDA) defined in [RFC8342]. In order to allow implementations to use the model even in cases when NMDA is not supported, the following companion module ietf-te-topology-state is defined as a state model, which mirrors the module ietf-te-topology defined earlier in this document. However, all data nodes in the companion module are non-configurable, to represent the applied configuration or the derived operational states.

The companion module, ietf-te-topology-state, is redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the module ietf-te-topology-state mirrors that of the module ietf-te-topology. The YANG tree of the module ietf-te-topology-state is not depicted separately.

B.1. TE Topology State YANG Module

This module references [RFC6001], [RFC8345], and [I-D.ietf-teas-yang-te-types].

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

  prefix "tet-s";

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

  import ietf-te-topology {
    prefix "tet";
  }

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

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

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

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

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

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

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

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

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

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

  description
    "TE topology state model.

     Copyright (c) 2019 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.
     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust’s Legal Provisions
     Relating to IETF Documents
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see the
     RFC itself for full legal notices.";

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

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

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

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

  /*
   * Data nodes
   */
  augment "/nw-s:networks/nw-s:network/nw-s:network-types" {
    description
      "Introduce new network type for TE topology.";
    container te-topology {
      presence "Indicates TE topology.";
      description
        "Its presence identifies the TE topology type.";
    }
  }

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

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

  augment "/nw-s:networks/nw-s:network/nw-s:node" {
    when "../nw-s:network-types/tet-s:te-topology" {
      description
        "Augmentation parameters apply only for networks with
         TE topology type.";
    }
    description
      "Configuration parameters for TE at node level.";
    leaf te-node-id {
      type te-types:te-node-id;
      description
        "The identifier of a node in the TE topology.
         A node is specific to a topology to which it belongs.";
    }
    container te {
      must "../te-node-id" {
        description
          "te-node-id is mandatory.";
      }
      must "count(../nw-s:supporting-node)<=1" {
        description
          "For a node in a TE topology, there cannot be more
           than 1 supporting node. If multiple nodes are abstracted,
           the underlay-topology is used.";
      }
      presence "TE support.";
      description
        "Indicates TE support.";
      uses tet:te-node-augment;
    } // te
  }

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

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

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

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

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

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

Appendix C. Example: YANG Model for Technology Specific Augmentations

This section provides an example YANG module to define a technology specific TE topology model for the example-topology described in Section 6.

```
module example-topology {
  yang-version 1.1;

  namespace "http://example.com/example-topology";
  prefix "ex-topo";

  import ietf-network {
    prefix "nw";
  }

  import ietf-network-topology {
    prefix "nt";
  }

  import ietf-te-topology {
    prefix "tet";
  }

  organization
    "Example Organization";
  contact
    "Editor: Example Author";

  description
    "This module defines a topology data model for the example
     technology.";

  revision 2018-06-15 {
    description
      "Initial revision.";
    reference
      "Example reference.";
  }

  /*
   * Data nodes
```

```
      */
     augment "/nw:networks/nw:network/nw:network-types/"
       + "tet:te-topology" {
       description
         "Augment network types to define example topology type.";
       container example-topology {
         presence
           "Introduce new network type for example topology.";
         description
           "Its presence identifies the example topology type.";
       }
     }

     augment "/nw:networks/nw:network/tet:te" {
       when "../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment network topology.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-1 {
           type uint8;
           description "Attribute 1 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes" {
       when "../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment node attributes.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-2 {
           type uint8;
           description "Attribute 2 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices" {
       when "../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment node connectivity matrices.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-3 {
           type uint8;
           description "Attribute 3 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix" {
       when "../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment node connectivity matrix.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-3 {
           type uint8;
           description "Attribute 3 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point" {
       when "../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment tunnel termination point.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-4 {
           type uint8;
           description "Attribute 4 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
           + "tet:te" {
       when "../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment link termination point.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-5 {
           type uint8;
           description "Attribute 5 for example technology.";
         }
       }
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
           + "tet:te-link-attributes" {
```

```
       when "../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       description "Augment link attributes.";
       container attributes {
         description "Attributes for example technology.";
         leaf attribute-6 {
           type uint8;
           description "Attribute 6 for example technology.";
         }
       }
     }

     /*
      * Augment TE bandwidth.
      */

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:max-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:max-resv-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:unreserved-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:client-layer-adaptation/"
       + "tet:switching-capability/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:path-constraints/tet:te-bandwidth/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:max-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:max-resv-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:max-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
```

```
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:max-resv-link-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:unreserved-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }

     augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
       + "tet:te/"
       + "tet:interface-switching-capability/tet:max-lsp-bandwidth/"
       + "tet:te-bandwidth/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf bandwidth-1 {
             type uint32;
             description "Bandwidth 1 for example technology.";
           }
         }
       }
       description "Augment TE bandwidth.";
     }
     /*
      * Augment TE label.
      */

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/tet:te/tet:templates/"
       + "tet:link-template/tet:te-link-attributes/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     /* Under te-node-attributes/connectivity-matrices */

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
```

```
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:path-properties/tet:path-route-objects/"
       + "tet:path-route-object/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
```

```
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
              example topology type.";
         }
         case "example" {
           container example {
             description "Attributes for example technology.";
             leaf label-1 {
               type uint32;
               description "Label 1 for example technology.";
             }
           }
         }
         description "Augment TE label.";
       }

       /* Under te-node-attributes/.../connectivity-matrix */

       augment "/nw:networks/nw:network/nw:node/tet:te/"
         + "tet:te-node-attributes/tet:connectivity-matrices/"
         + "tet:connectivity-matrix/tet:from/"
         + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
         + "tet:te-label/tet:technology" {
         when "../../../../../../../../../../../nw:network-types/"
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
              example topology type.";
         }
         case "example" {
           container example {
             description "Attributes for example technology.";
             leaf label-1 {
               type uint32;
               description "Label 1 for example technology.";
             }
           }
         }
         description "Augment TE label.";
       }
```

```
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/tet:from/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/tet:to/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/tet:to/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:te-node-attributes/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:path-properties/tet:path-route-objects/"
       + "tet:path-route-object/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../nw:network-types/"
```

```
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     /* Under information-source-entry/connectivity-matrices */

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
```

```
         description "Augment TE label.";
       }

       augment "/nw:networks/nw:network/nw:node/tet:te/"
         + "tet:information-source-entry/tet:connectivity-matrices/"
         + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
         + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
         when "../../../../../../../../../../../../nw:network-types/"
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
              example topology type.";
         }
         case "example" {
           container example {
             description "Attributes for example technology.";
             leaf label-1 {
               type uint32;
               description "Label 1 for example technology.";
             }
           }
         }
         description "Augment TE label.";
       }

       augment "/nw:networks/nw:network/nw:node/tet:te/"
         + "tet:information-source-entry/tet:connectivity-matrices/"
         + "tet:path-properties/tet:path-route-objects/"
         + "tet:path-route-object/tet:type/"
         + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
         when "../../../../../../../../../../../../nw:network-types/"
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
              example topology type.";
         }
         case "example" {
           container example {
             description "Attributes for example technology.";
             leaf label-1 {
               type uint32;
               description "Label 1 for example technology.";
             }
           }
         }
         description "Augment TE label.";
       }

       /* Under information-source-entry/.../connectivity-matrix */

       augment "/nw:networks/nw:network/nw:node/tet:te/"
         + "tet:information-source-entry/tet:connectivity-matrices/"
         + "tet:connectivity-matrix/tet:from/"
         + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
         + "tet:te-label/tet:technology" {
         when "../../../../../../../../../../nw:network-types/"
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
              example topology type.";
         }
         case "example" {
           container example {
             description "Attributes for example technology.";
             leaf label-1 {
               type uint32;
               description "Label 1 for example technology.";
             }
           }
         }
         description "Augment TE label.";
       }

       augment "/nw:networks/nw:network/nw:node/tet:te/"
         + "tet:information-source-entry/tet:connectivity-matrices/"
         + "tet:connectivity-matrix/tet:from/"
         + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
         + "tet:te-label/tet:technology" {
         when "../../../../../../../../../../nw:network-types/"
           + "tet:te-topology/ex-topo:example-topology" {
           description
             "Augmentation parameters apply only for networks with
```

```
             example topology type.";
        }
        case "example" {
          container example {
            description "Attributes for example technology.";
            leaf label-1 {
              type uint32;
              description "Label 1 for example technology.";
            }
          }
        }
        description "Augment TE label.";
      }

      augment "/nw:networks/nw:network/nw:node/tet:te/"
        + "tet:information-source-entry/tet:connectivity-matrices/"
        + "tet:connectivity-matrix/tet:to/"
        + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
        + "tet:te-label/tet:technology" {
        when "../../../../../../../../../../nw:network-types/"
          + "tet:te-topology/ex-topo:example-topology" {
          description
            "Augmentation parameters apply only for networks with
             example topology type.";
        }
        case "example" {
          container example {
            description "Attributes for example technology.";
            leaf label-1 {
              type uint32;
              description "Label 1 for example technology.";
            }
          }
        }
        description "Augment TE label.";
      }

      augment "/nw:networks/nw:network/nw:node/tet:te/"
        + "tet:information-source-entry/tet:connectivity-matrices/"
        + "tet:connectivity-matrix/tet:to/"
        + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
```

```
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }
     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:information-source-entry/tet:connectivity-matrices/"
       + "tet:connectivity-matrix/"
       + "tet:path-properties/tet:path-route-objects/"
       + "tet:path-route-object/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     /* Under tunnel-termination-point/local-link-connectivities */

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../nw:network-types/tet:te-topology/"
          + "ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
```

```
            "Augmentation parameters apply only for networks with
             example topology type.";
        }
        case "example" {
          container example {
            description "Attributes for example technology.";
            leaf label-1 {
              type uint32;
              description "Label 1 for example technology.";
            }
          }
        }
        description "Augment TE label.";
      }

      augment "/nw:networks/nw:network/nw:node/tet:te/"
        + "tet:tunnel-termination-point/tet:local-link-connectivities/"
        + "tet:path-properties/tet:path-route-objects/"
        + "tet:path-route-object/tet:type/"
        + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
        when "../../../../../../../../../../../../nw:network-types/"
          + "tet:te-topology/ex-topo:example-topology" {
          description
            "Augmentation parameters apply only for networks with
             example topology type.";
        }
        case "example" {
          container example {
            description "Attributes for example technology.";
            leaf label-1 {
              type uint32;
              description "Label 1 for example technology.";
            }
          }
        }
        description "Augment TE label.";
      }

      /* Under tunnel-termination-point/.../local-link-connectivity */

      augment "/nw:networks/nw:network/nw:node/tet:te/"
```

```
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nw:node/tet:te/"
       + "tet:tunnel-termination-point/tet:local-link-connectivities/"
       + "tet:local-link-connectivity/"
       + "tet:path-properties/tet:path-route-objects/"
       + "tet:path-route-object/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     /* Under te-link-attributes */

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
```

```
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:underlay/tet:primary-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:te-link-attributes/"
       + "tet:underlay/tet:backup-path/tet:path-element/tet:type/"
       + "tet:label/tet:label-hop/tet:te-label/tet:technology" {
       when "../../../../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }
     /* Under te-link information-source-entry */

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-start/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
         }
       }
       description "Augment TE label.";
     }

     augment "/nw:networks/nw:network/nt:link/tet:te/"
       + "tet:information-source-entry/"
       + "tet:label-restrictions/tet:label-restriction/tet:label-end/"
       + "tet:te-label/tet:technology" {
       when "../../../../../../../nw:network-types/"
         + "tet:te-topology/ex-topo:example-topology" {
         description
           "Augmentation parameters apply only for networks with
            example topology type.";
       }
       case "example" {
         container example {
           description "Attributes for example technology.";
           leaf label-1 {
             type uint32;
             description "Label 1 for example technology.";
           }
```

```
          }
        }
        description "Augment TE label.";
      }
    }
```

Contributors

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

Dieter Beller
Nokia
Email: Dieter.Beller@nokia.com

Carlo Perocchio
Ericsson
Email: carlo.perocchio@ericsson.com

Italo Busi
Huawei Technologies
Email: Italo.Busi@huawei.com

Authors' Addresses

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com

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

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Juniper Networks
Email: tsaad@juniper.net

Himanshu Shah
Ciena
Email: hshah@ciena.com

Oscar Gonzalez De Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group
Internet Draft
Intended Status: Standard Track
Expires: November 8, 2019

Y. Lee (Editor)
Dhruv Dhody
Satish Karunanithi
Huawei

Ricard Vilalta
CTTC

Daniel King
Lancaster University

Daniele Ceccarelli
Ericsson

May 8, 2019

YANG models for VN & TE Performance Monitoring Telemetry and Scaling Intent Autonomics

draft-lee-teas-actn-pm-telemetry-autonomics-17

Status of this Memo

Copyright Notice

Abstract

This document provides YANG data models that describe performance monitoring telemetry and scaling intent mechanism for TE-tunnels and Virtual Networks (VN).

The models presented in this draft allow customers to subscribe to and monitor their key performance data of their interest on the level of TE-tunnel or VN. The models also provide customers with the ability to program autonomic scaling intent mechanism on the level of TE-tunnel as well as VN.

Table of Contents

1. Introduction

The YANG model discussed in [VN] is used to operate customer-driven Virtual Networks (VNs) during the VN instantiation, VN computation, and its life-cycle service management and operations. YANG model discussed in [TE-Tunnel] is used to operate TE-tunnels during the tunnel instantiation, and its life-cycle management and operations.

The models presented in this draft allow the applications hosted by the customers to subscribe to and monitor their key performance data of their interest on the level of VN [VN] or TE-tunnel [TE-Tunnel]. The key characteristic of the models presented in this document is a top-down programmability that allows the applications hosted by the customers to subscribe to and monitor key performance data of their interest and autonomic scaling intent mechanism on the level of VN as well as TE-tunnel.

According to the classification of [RFC8309], the YANG data models presented in this document can be classified as customer service models, which is mapped to CMI (Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface) of ACTN [RFC8453].

[RFC8233] describes key network performance data to be considered for end-to-end path computation in TE networks. Key performance indicator (KPI) is a term that describes critical performance data that may affect VN/TE-tunnel service. The services provided can be optimized to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers.

This document provides YANG data models generically applicable to any VN/TE-Tunnel service clients to provide an ability to program their customized performance monitoring subscription and publication data models and automatic scaling in/out intent data models. These models can be utilized by a client network controller to initiate these capability to a transport network controller communicating with the client controller via a NETCONF [RFC8341] or a RESTCONF [RFC8040] interface.

The term performance monitoring being used in this document is different from the term that has been used in transport networks for many years. Performance monitoring in this document refers to subscription and publication of streaming telemetry data. Subscription is initiated by the client (e.g., CNC) while publication is provided by the network (e.g., MDSC/PNC) based on the client's subscription. As the scope of performance monitoring in this document is telemetry data on the level of client's VN or TE-tunnel, the entity interfacing the client (e.g., MDSC) has to provide VN or TE-tunnel level information. This would require controller capability to derive VN or TE-tunnel level performance data based on lower-level data collected via PM counters in the Network Elements (NE). How the controller entity derives such customized level data (i.e., VN or TE-tunnel level) is out of the scope of this document.

The data model includes configuration and state data according to the new Network Management Datastore Architecture [RFC8342].

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

Key Performance Data: This refers to a set of data the customer is interested in monitoring for their instantiated VNs or TE-tunnels. Key performance data and key performance indicators are inter-exchangeable in this draft.

Scaling: This refers to the network ability to re-shape its own resources. Scale out refers to improve network performance by increasing the allocated resources, while scale in refers to decrease the allocated resources, typically because the existing resources are unnecessary.

Scaling Intent: To declare scaling conditions, scaling intent is used. Specifically, scaling intent refers to the intent expressed by the client that allows the client to program/configure conditions of their key performance data either for scaling out or scaling in. Various conditions can be set for scaling intent on either VN or TE-tunnel level.

Network Autonomics: This refers to the network automation capability that allows client to initiate scaling intent mechanisms and provides the client with the status of the adjusted network resources based on the client's scaling intent in an automated fashion.

1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
rt	ietf-routing-types	[RFC8294]
te	ietf-te	[TE-Tunnel]
te-types	ietf-te-types	[TE-Types]
te-tel	ietf-te-kpi-telemetry	[This I-D]
vn	ietf-vn	[VN]
vn-tel	ietf-vn-kpi-telemetry	[This I-D]

Table 1: Prefixes and corresponding YANG modules

2. Use-Cases

[PERF] describes use-cases relevant to this draft. It introduces the dynamic creation, modification and optimization of services based on the performance monitoring. Figure 1 shows a high-level workflows for dynamic service control based on traffic monitoring.

```
      +----------------------------------------------+
      | Client   +-----------------------------+     |
      |          | Dynamic Service Control APP |     |
      |          +-----------------------------+     |
      +----------------------------------------------+
      1.Traffic|  /|\4.Traffic            | /|\
      Monitor& |   | Monitor              |  | 8.Traffic
      Optimize |   | Result     5.Service |  | modify &
      Policy   |   |              modify& |  | optimize
              \|/  |        optimize Req.\|/ | result
      +----------------------------------------------+
      | Orchestrator                                 |
      |    +-------------------------------+         |
      |    |Dynamic Service Control Agent  |         |
      |    +-------------------------------+         |
      |    +---------------+ +-------------------+   |
      |    | Flow Optimize | | vConnection Agent |   |
      |    +---------------+ +-------------------+   |
      +----------------------------------------------+
      2. Path |   /|\3.Traffic            | /|\
      Monitor |    | Monitor              |  |7.Path
      Request |    | Result      6.Path   |  | modify &
              |    |             modify&  |  | optimize
             \|/   |        optimize Req.\|/ | result
      +----------------------------------------------+
      | Network SDN Controller                       |
      |  +----------------------+ +-----------------+|
      |  | Network Provisioning | |Abstract Topology||
      |  +----------------------+ +-----------------+|
      |  +------------------+ +--------------------+ |
      |  |Network Monitoring| |Physical Topology DB| |
      |  +------------------+ +--------------------+ |
      +----------------------------------------------+
```

Figure 1 Workflows for dynamic service control based on traffic monitoring

Some of the key points from [PERF] are as follows:

- Network traffic monitoring is important to facilitate automatic discovery of the imbalance of network traffic, and initiate the network optimization, thus helping the network operator or the virtual network service provider to use the network more efficiently and save the Capital Expense (CAPEX) and the Operating Expense (OPEX).

- Customer services have various Service Level Agreement (SLA) requirements, such as service availability, latency, latency jitter, packet loss rate, Bit Error Rate (BER), etc. The transport network can satisfy service availability and BER requirements by providing different protection and restoration mechanisms. However, for other performance parameters, there are no such mechanisms. In order to provide high quality services according to customer SLA, one possible solution is to measure the SLA related performance parameters, and dynamically provision and optimize services based on the performance monitoring results.
- Performance monitoring in a large scale network could generate a huge amount of performance information. Therefore, the appropriate way to deliver the information in the client and network interfaces should be carefully considered.

3. Design of the Data Models

The YANG models developed in this document describe two models:

(i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & 7.1 for details).

(ii) VN KPI Telemetry Model which provides the VN level of the aggregated performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer (See Section 3.2 & 7.2 for details).

3.1. TE KPI Telemetry Model

This module describes performance telemetry for TE-tunnel model. The telemetry data is augmented to tunnel state. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data (See Section 5 for details)

The TE KPI Telemetry Model augments the TE-Tunnel Model to enhance TE performance monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.

```
                +------------+          +--------------+
                | TE-Tunnel  |          |    TE KPI    |
                |   Model    |<---------|  Telemetry   |
                +------------+ augments |     Model    |
                                        +--------------+
```

3.2. VN KPI Telemetry Model

This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters (See Section 4 for illustrations).

Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters. Grouping operation (such as maximum, mean) could be set at the time of configuration. For example, if maximum grouping operation is used for delay at the VN level, the VN telemetry data is reported as the maximum {delay_vn_member_1, delay_vn_member_2,.. delay_vn_member_N}. Thus, this telemetry abstraction mechanism allows the grouping of a certain common set of telemetry values under a grouping operation. This can be done at the VN-member level to suggest how the E2E telemetry be inferred from the per domain tunnel created and monitored by PNCs. One proposed example is the following:

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

     1.Client sets the      |   /|\   2. Orchestrator pushes:
     grouping op, and       |    |
     subscribes to the      |    |    VN level telemetry for
     VN level telemetry for |    |    - VN Utilized-bw-percentage
     Delay and              |    |       (Minimum across VN Members)
     Utilized-bw-pecentage  |    |    - VN Delay (Maximum across VN
                           \|/   |     Members)
      +-------------------------------------------------------------+
      | Orchestrator                                                |
      |                                                             |
      +-------------------------------------------------------------+
```

The VN Telemetry Model augments the basic VN model to enhance VN monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of VNs based on the performance monitoring data collected via the VN Telemetry YANG model.

```
                +----------+          +--------------+
                |    VN    | augments |      VN      |
                |   Model  |<---------|   Telemetry  |
                +----------+          |     Model    |
                                      +--------------+
```

4. Autonomic Scaling Intent Mechanism

Scaling intent configuration mechanism allows the client to configure automatic scale-in and scale-out mechanisms on both the TE-tunnel and the VN level. Various conditions can be set for auto-scaling based on the PM telemetry data.

There are a number of parameters involved in the mechanism:

- scale-out-intent or scale-in-intent: whether to scale-out or scale-in.
- performance-type: performance metric type (e.g., one-way-delay, one-way-delay-min, one-way-delay-max, two-way-delay, two-way-delay-min, two-way-delay-max, utilized bandwidth, etc.)

- . threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.
- . scaling-operation-type: in case where scaling condition can be set with one or more performance types, then scaling-operation-type (AND, OR, MIN, MAX, etc.) is applied to these selected performance types and its threshold values.
- . Threshold-time: the duration for which the criteria must hold true.
- . Cooldown-time: the duration after a scaling action has been triggered, for which there will be no further operation.

The following tree is a part of ietf-te-kpi-telemetry tree whose model is presented in full detail in Sections 6 & 7.

```
module: ietf-te-kpi-telemetry
  augment /te:te/te:tunnels/te:tunnel:
    +-rw te-scaling-intent
    |  +-rw scale-in-intent
    |  |  +-rw threshold-time?            uint32
    |  |  +-rw cooldown-time?             uint32
    |  |  +-rw scale-in-operation-type?   scaling-criteria-operation
    |  |  +-rw scaling-condition* [performance-type]
    |  |     +-rw performance-type           identityref
    |  |     +-rw threshold-value?           string
    |  |     +-rw te-telemetry-tunnel-ref?
    |  |        -> /te:te/tunnels/tunnel/name
    |  +-rw scale-out-intent
    |     +-rw threshold-time?             uint32
    |     +-rw cooldown-time?              uint32
    |     +-rw scale-out-operation-type?   scaling-criteria-operation
    |     +-rw scaling-condition* [performance-type]
    |        +-rw performance-type           identityref
    |        +-rw threshold-value?           string
    |        +-rw te-telemetry-tunnel-ref?
    |           -> /te:te/tunnels/tunnel/name
```

Let say the client wants to set the scaling out operation based on two performance-types (e.g., two-way-delay and utilized-bandwidth for a te-tunnel), it can be done as follows:

- . Set Threshold-time: x (sec) (duration for which the criteria must hold true)

- Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)
- Set AND for the scale-out-operation-type

In the scaling condition's list, the following two components can be set:

List 1: Scaling Condition for Two-way-delay

- performance type: Two-way-delay
- threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- performance type: Utilized bandwidth
- threshold-value: w megabytes

5. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [YANG-PUSH] and [Event-Notification] can be used. This mechanism currently allows the user to:

- Subscribe to notifications on a per client basis.

- Specify subtree filters or xpath filters so that only interested contents will be sent.

- Specify either periodic or on-demand notifications.

5.1. YANG Push Subscription Examples

[YANG-PUSH] allows subscriber applications to request a continuous, customized stream of updates from a YANG datastore.

Below example shows the way for a client to subscribe to the telemetry information for a particular tunnel (Tunnel1). The telemetry parameter that the client is interested in is one-way-delay.

```
   <netconf:rpc netconf:message-id="101"
       xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
       <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
          <filter netconf:type="subtree">
             <te xmlns="urn:ietf:params:xml:ns:yang:ietf-te">
                <tunnels>
                   <tunnel>
                     <name>Tunnel1</name>
                     <identifier/>
                     <state>
                       <te-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                             ietf-te-kpi-telemetry">
                           <one-way-delay/>
                        </te-telemetry>
                     </state>
                    </tunnel>
                 </tunnels>
             </te>
          </filter>
          <period>500</period>
          <encoding>encode-xml</encoding>
       </establish-subscription>
    </netconf:rpc>
```

This example shows the way for a client to subscribe to the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way-delay and one-way-utilized-bandwidth.

```
   <netconf:rpc netconf:message-id="101"
       xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
       <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
          <filter netconf:type="subtree">
             <vn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-vn">
                <vn>
                   <vn-list>
                     <vn-id/>
                     <vn-name/>
                     <vn-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                            ietf-vn-kpi-telemetry">
                         <one-way-delay/>
                         <one-way-utilized-bandwidth/>
                     </vn-telemetry >
                   </vn-list>
                 </vn>
             </vn-state>
          </filter>
          <period>500</period>
       </establish-subscription>
    </netconf:rpc>
```

6. YANG Data Tree

```
module: ietf-te-kpi-telemetry
  augment /te:te/te:tunnels/te:tunnel:
    +--rw te-scaling-intent
    |  +--rw scale-in-intent
    |  |  +--rw threshold-time?            uint32
    |  |  +--rw cooldown-time?             uint32
    |  |  +--rw scale-in-operation-type?   scaling-criteria-operation
    |  |  +--rw scaling-condition* [performance-type]
    |  |     +--rw performance-type           identityref
    |  |     +--rw threshold-value?           string
    |  |     +--rw te-telemetry-tunnel-ref?
    |  |             -> /te:te/tunnels/tunnel/name
    |  +--rw scale-out-intent
    |     +--rw threshold-time?             uint32
    |     +--rw cooldown-time?              uint32
    |     +--rw scale-out-operation-type?   scaling-criteria-operation
    |     +--rw scaling-condition* [performance-type]
    |        +--rw performance-type           identityref
    |        +--rw threshold-value?           string
    |        +--rw te-telemetry-tunnel-ref?
    |                -> /te:te/tunnels/tunnel/name
    +--ro te-telemetry
       +--ro id?                            string
       +--ro performance-metrics-one-way
       |  +--ro one-way-delay?                           uint32
       |  +--ro one-way-delay-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-residual-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-residual-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-available-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-available-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-utilized-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-utilized-bandwidth-normality?
       |          te-types:performance-metrics-normality
       +--ro performance-metrics-two-way
       |  +--ro two-way-delay?            uint32
       |  +--ro two-way-delay-normality?
       |          te-types:performance-metrics-normality
       +--ro te-ref?
               -> /te:te/tunnels/tunnel/name


module: ietf-vn-kpi-telemetry
  augment /vn:vn/vn:vn-list:
    +--rw vn-scaling-intent
    |  +--rw scale-in-intent
    |  |  +--rw threshold-time?            uint32
    |  |  +--rw cooldown-time?             uint32
    |  |  +--rw scale-in-operation-type?   scaling-criteria-operation
    |  |  +--rw scaling-condition* [performance-type]
    |  |     +--rw performance-type           identityref
    |  |     +--rw threshold-value?           string
    |  |     +--rw te-telemetry-tunnel-ref?
    |  |             -> /te:te/tunnels/tunnel/name
    |  +--rw scale-out-intent
    |     +--rw threshold-time?             uint32
    |     +--rw cooldown-time?              uint32
    |     +--rw scale-out-operation-type?   scaling-criteria-operation
    |     +--rw scaling-condition* [performance-type]
    |        +--rw performance-type           identityref
    |        +--rw threshold-value?           string
    |        +--rw te-telemetry-tunnel-ref?
    |                -> /te:te/tunnels/tunnel/name
    +--ro vn-telemetry
       +--ro performance-metrics-one-way
       |  +--ro one-way-delay?                           uint32
       |  +--ro one-way-delay-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-residual-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-residual-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-available-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-available-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-utilized-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-utilized-bandwidth-normality?
       |          te-types:performance-metrics-normality
       +--ro performance-metrics-two-way
       |  +--ro two-way-delay?            uint32
       |  +--ro two-way-delay-normality?
       |          te-types:performance-metrics-normality
       +--ro grouping-operation?            grouping-operation
  augment /vn:vn/vn:vn-list/vn:vn-member-list:
    +--ro vn-member-telemetry
       +--ro performance-metrics-one-way
       |  +--ro one-way-delay?                           uint32
       |  +--ro one-way-delay-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-residual-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-residual-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
       |  +--ro one-way-available-bandwidth?
       |  |       rt-types:bandwidth-ieee-float32
       |  +--ro one-way-available-bandwidth-normality?
       |  |       te-types:performance-metrics-normality
```

```
           |  +--ro one-way-utilized-bandwidth?
           |  |       rt-types:bandwidth-ieee-float32
           |  +--ro one-way-utilized-bandwidth-normality?
           |          te-types:performance-metrics-normality
           +--ro performance-metrics-two-way
           |  +--ro two-way-delay?             uint32
           |  +--ro two-way-delay-normality?
           |          te-types:performance-metrics-normality
           +--ro te-grouped-params*
           |       -> /te:te/tunnels/tunnel/te-kpi:te-telemetry/id
           +--ro grouping-operation?            grouping-operation
```

7. Yang Data Model

7.1. ietf-te-kpi-telemetry model

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-te-kpi-telemetry@2019-04-18.yang"

module ietf-te-kpi-telemetry {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry";
  prefix te-tel;

  import ietf-te {
    prefix te;
   reference
     "RFC YYYY: A YANG Data Model for Traffic Engineering
      Tunnels and Interfaces";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-ietf-teas-yang-te
    becomes an RFC.*/

  import ietf-te-types {
    prefix te-types;
   reference
     "RFC YYYY: Traffic Engineering Common YANG Types";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-ietf-teas-yang-te-types
    becomes an RFC.*/
  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "Editor: Young Lee <leeyoung@huawei.com>
     Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
     Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
     Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";
  description
    "This module describes YANG data model for performance
    monitoring telemetry for te tunnels.

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

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

    This version of this YANG module is part of RFC XXXX; see
     the RFC itself for full legal notices.";

  /* Note: The RFC Editor will replace XXXX with the number
     assigned to the RFC once draft-lee-teas-pm-telemetry-
     autonomics becomes an RFC.*/

  revision 2019-04-18 {
    description
      "Initial revision. This YANG file defines
       a YANG model for TE telemetry.";
    reference "Derived from earlier versions of base YANG files";
  }

  identity telemetry-param-type {
    description
      "Base identity for telemetry param types";
  }

  identity one-way-delay {
    base telemetry-param-type;
    description
      "To specify average Delay in one (forward)
       direction";
```

```
   reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  identity two-way-delay {
    base telemetry-param-type;
    description
      "To specify average Delay in both (forward and reverse)
       directions";
   reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  identity one-way-delay-variation {
    base telemetry-param-type;
    description
      "To specify average Delay Variation in one (forward) direction";
   reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  identity two-way-delay-variation {
    base telemetry-param-type;
    description
      "To specify average Delay Variation in both (forward and reverse)
       directions";
   reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  identity utilized-bandwidth {
    base telemetry-param-type;
    description
      "To specify utilized bandwidth over the specified source
       and destination.";
   reference
      "RFC7471: OSPF Traffic Engineering (TE) Metric Extensions.
       RFC8570: IS-IS Traffic Engineering (TE) Metric Extensions.
       RFC7823: Performance-Based Path Selection for Explicitly
       Routed Label Switched Paths (LSPs) Using TE Metric
       Extensions";
  }

  identity utilized-percentage {
    base telemetry-param-type;
    description
      "To specify utilization percentage of the entity
       (e.g., tunnel, link, etc.)";
  }

  typedef scaling-criteria-operation {
    type enumeration {
      enum AND {
        description
          "AND operation";
      }
      enum OR {
        description
          "OR operation";
      }
    }
    description
      "Operations to analize list of scaling criterias";
  }

  grouping scaling-duration {
    description
      "Base scaling criteria durations";
    leaf threshold-time {
      type uint32;
      units "seconds";
      description
        "The duration for which the criteria must hold true";
    }
    leaf cooldown-time {
      type uint32;
      units "seconds";
      description
```

```
        "The duration after a scaling-in/scaling-out action has been
         triggered, for which there will be no further operation";
    }
  }

  grouping scaling-criteria {
    description
      "Grouping for scaling criteria";
    leaf performance-type {
      type identityref {
        base telemetry-param-type;
      }
      description
        "Reference to the tunnel level telemetry type";
    }
    leaf threshold-value {
      type string;
      description
        "Scaling threshold for the telemetry parameter type";
    }
    leaf te-telemetry-tunnel-ref {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/te:name";
      }
      description
        "Reference to tunnel";
    }
  }

  grouping scaling-in-intent {
    description
      "Basic scaling in intent";
    uses scaling-duration;
    leaf scale-in-operation-type {
      type scaling-criteria-operation;
      default "AND";
      description
        "Operation to be applied to check between
         scaling criterias to check if the scale in
         threshold condition has been met.
         Defaults to AND";
    }
    list scaling-condition {
      key "performance-type";
      description
        "Scaling conditions";
      uses scaling-criteria;
    }
  }

  grouping scaling-out-intent {
    description
      "Basic scaling out intent";
    uses scaling-duration;
    leaf scale-out-operation-type {
      type scaling-criteria-operation;
      default "OR";
      description
        "Operation to be applied to check between
         scaling criterias to check if the scale out
         threshold condition has been met.
         Defauls to OR";
    }
    list scaling-condition {
      key "performance-type";
      description
        "Scaling conditions";
      uses scaling-criteria;
    }
  }

  augment "/te:te/te:tunnels/te:tunnel" {
    description
      "Augmentation parameters for config scaling-criteria
       TE tunnel topologies. Scale in/out criteria might be used
       for network autonomics in order the controller
       to react to a certain set of monitored params.";
    container te-scaling-intent {
      description
        "scaling intent";
      container scale-in-intent {
        description
          "scale-in";
        uses scaling-in-intent;
      }
      container scale-out-intent {
        description
          "scale-out";
        uses scaling-out-intent;
      }
    }
    container te-telemetry {
      config false;
      description
        "telemetry params";
      leaf id {
        type string;
        description
          "Id of telemetry param";
      }
      uses te-types:performance-metrics-attributes;
      leaf te-ref {
        type leafref {
          path "/te:te/te:tunnels/te:tunnel/te:name";
        }
        description
          "Reference to measured te tunnel";
      }
    }
  }
}
<CODE ENDS>
```

7.2. ietf-vn-kpi-telemetry model

The YANG code is as follows:

```
<CODE BEGINS> file "ietf-vn-kpi-telemetry@2019-04-18.yang"

module ietf-vn-kpi-telemetry {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry";
  prefix vn-tel;

  import ietf-vn {
    prefix vn;
   reference
     "RFC YYYY: A YANG Data Model for VN Operation";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-ietf-teas-actn-vn-yang
    becomes an RFC.*/

  import ietf-te {
    prefix te;
   reference
     "RFC YYYY: A YANG Data Model for Traffic Engineering
```

```
      Tunnels and Interfaces";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-ietf-teas-yang-te
    becomes an RFC.*/

  import ietf-te-types {
    prefix te-types;
   reference
     "RFC YYYY: Traffic Engineering Common YANG Types";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-ietf-teas-yang-te-types
    becomes an RFC.*/

  import ietf-te-kpi-telemetry {
    prefix te-kpi;
   reference
     "RFC YYYY: YANG models for VN & TE Performance Monitoring
      Telemetry and Scaling Intent Autonomics";
  }

  /* Note: The RFC Editor will replace YYYY with the number
     assigned to the RFC once draft-lee-teas-actn-pm-telemetry
    -autonomics becomes an RFC.*/

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "Editor: Young Lee <leeyoung@huawei.com>
     Editor: Dhruv Dhody <dhruv.ietf@gmail.com>
     Editor: Ricard Vilalta <ricard.vilalta@cttc.es>
     Editor: Satish Karunanithi <satish.karunanithi@gmail.com>";

  description
    "This module describes YANG data models for performance
    monitoring telemetry for vn.

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

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

    This version of this YANG module is part of RFC XXXX; see
     the RFC itself for full legal notices.";

  /* Note: The RFC Editor will replace XXXX with the number
     assigned to the RFC once draft-lee-teas-pm-telemetry-
     autonomics becomes an RFC.*/

  revision 2019-04-18 {
    description
      "Initial revision. This YANG file defines
       the VN telemetry.";
    reference "Derived from earlier versions of base YANG files";
  }

  typedef grouping-operation {
    type enumeration {
      enum MINIMUM {
        description
          "Select the minimum param";
      }
      enum MAXIMUM {
        description
          "Select the maximum param";
      }
      enum MEAN {
        description
          "Select the MEAN of the params";
      }
      enum STD_DEV {
        description
          "Select the standard deviation of the
           monitored params";
      }
      enum AND {
        description
          "Select the AND of the params";
      }
      enum OR {
        description
          "Select the OR of the params";
      }
    }
    description
```

```
      "Operations to analize list of monitored params";
  }

  grouping vn-telemetry-param {
    description
      "augment of te-kpi:telemetry-param for VN specific params";
    leaf-list te-grouped-params {
      type leafref {
        path "/te:te/te:tunnels/te:tunnel/te-kpi:te-telemetry/te-kpi:id";
      }
      description
        "Allows the definition of a vn-telemetry param
         as a grouping of underlying TE params";
    }
    leaf grouping-operation {
      type grouping-operation;
      description
        "describes the operation to apply to
         te-grouped-params";
    }
  }

  augment "/vn:vn/vn:vn-list" {
    description
      "Augmentation parameters for state TE VN topologies.";
    container vn-scaling-intent {
      description
        "scaling intent";
      container scale-in-intent {
        description
          "VN scale-in";
        uses te-kpi:scaling-in-intent;
      }
      container scale-out-intent {
        description
          "VN scale-out";
        uses te-kpi:scaling-out-intent;
      }
    }
    container vn-telemetry {
      config false;
      description
        "VN telemetry params";
      uses te-types:performance-metrics-attributes;
      leaf grouping-operation {
        type grouping-operation;
        description
```

```
          "describes the operation to apply to the VN-members";
      }
    }
  }
  augment "/vn:vn/vn:vn-list/vn:vn-member-list" {
    description
      "Augmentation parameters for state TE vn member topologies.";
    container vn-member-telemetry {
      config false;
      description
        "VN member telemetry params";
      uses te-types:performance-metrics-attributes;
      uses vn-telemetry-param;
    }
  }
}
<CODE ENDS>
```

8. Security Considerations

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

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. The NETCONF Protocol over Secure Shell (SSH) [RFC6242] describes a method for invoking and running NETCONF within a Secure Shell (SSH) session as an SSH subsystem. The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true"). These data nodes may be considered sensitive or vulnerable in some network environments.

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

```
 /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent
/te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent

/vn:vn/vn:vn-list/vn-scaling-intent/scale-in-intent
/vn:vn/vn:vn-list/vn-scaling-intent/scale-out-intent
```

9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

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

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

This document registers the following YANG modules in the YANG Module.

Names registry [RFC7950]:

```
--------------------------------------------------------------------
name:          ietf-te-kpi-telemetry
namespace:     urn:ietf:params:xml:ns:yang:ietf-te-kpi-telemetry
prefix:        te-tel
reference:     RFC XXXX (TDB)
```

```
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:         ietf-vn-kpi-telemetry
   namespace:    urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry
   prefix:       vn-tel
   reference:    RFC XXXX (TDB)
   --------------------------------------------------------------------
```

10. Acknowledgements

We thank Rakesh Gandhi, Tarek Saad and Igor Bryskin for useful discussions and their suggestions for this work.

11. References

11.1. Normative References

[RFC6242] M. Wasserman, "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

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

[RFC7950] M. Bjorklund, Ed., "The YANG 1.1 Data Modeling Language", August 2016.

[RFC8040] A. Bierman, M. Bjorklund, K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.

[RFC8233] D. Dhody, et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to compute service aware Label Switched Path (LSP)", RFC 8233, September 2017.

[RFC8341] A. Bierman, M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.

[RFC8342] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018.

[RFC8446] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3", RFC8446, August 2018.

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[TE-Types] T. Saad, et.al, "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types, work in progress.

[VN] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

11.2. Informative References

[RFC3688] M. Mealling, "The IETF XML Registry", RFC 3688, January 2004.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, March 2015.

[RFC7823] Atlas, A., Drake, J., Giacalone, S., and S. Previdi,"Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions", RFC 7823, May 2016.

[RFC8294] X. Liu, et al, "Routing Area Common YANG Data Types", RFC 8294, December 2017.

[RFC8340] M. Bjorklund and L. Berger (Editors), "YANG Tree Diagrams", RFC 8340, March 2018.

[YANG-PUSH] A. Clemm, et al, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push, work in progress.

[Event-Notification] E. Voit, et al, "Dynamic subscription to YANG Events and Datastores over NETCONF", draft-ietf-netconf-netconf-event-notifications, work in progress.

[PERF] Y. XU, et al., "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control, work in progress.

[RFC8309] Q. Wu, W. Cheng, and A. Farrel. "Service Models Explained", RFC 8309, January 2018.

[RFC8453] D. Ceccarelli and Y. Lee (Editors), "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, March 2019.

12. Contributors

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive Suite 173
Plano, TX 75024, USA

Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technology
Leela Palace
Bangalore, Karnataka 560008
India

Email: dhruv.dhody@huawei.com

Satish Karunanithi
Huawei Technology
Leela Palace
Bangalore, Karnataka 560008
India

Email: satish.karunanithi@gmail.com

Ricard Vilalta
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)
Av. Carl Friedrich Gauss 7
08860 - Castelldefels
Barcelona (Spain)
Email: ricard.vilalta@cttc.es

Daniel King
Lancaster University

Email: d.king@lancaster.ac.uk

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

TEAS Working Group
Internet Draft
Intended Status: Standard Track
Expires: November 29, 2018

Y. Lee (Editor)
Dhruv Dhody
Huawei
D. Ceccarelli
Ericsson
Igor Bryskin
Huawei
Bin Yeong Yoon
ETRI

May 29, 2018

A Yang Data Model for ACTN VN Operation

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

Abstract

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

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

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

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

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

1.1. Terminology

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

2. ACTN CMI context

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

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

Figure 1. ACTN CMI

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

2.1. Type 1 VN

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

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

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

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

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

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

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

Figure 2. Abstract Node (One node topology)

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

2.2. Type 2 VN

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

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

```
        +-----------------------------------------------+
        |            S1                S2               |
        |             O---------------O                 |
        |     ________/ \______         \               |
        |    /                 \         \              |
        |S3 /                   \ S4      \ S5         |
  L1----|-O----------------------O---------O-----------|------L4
        |   \                     \         \          |
        |    \                     \         \         |
        |     \ S6                  \ S7      \ S8     |
        |      O     ----------------O---------O-------|------L7
        |     / \   /                 \   ____/        |
        |S9  /   \ /S10                \ /             |
 L2-----|---O-----O---------------------O--------------|------L8
        |  /                          S11              |
 L3-----|--                                            |
        |                                              |
        +----------------------------------------------+
```

Figure 3. Type 2 topology

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

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

3. High-Level Control Flows with Examples

3.1. Type 1 VN Illustration

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

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

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

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

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

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

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

3.2. Type 2 VN Illustration

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

VN-Member 1	L1-L4
VN-Member 2	L1-L7 (via S4 and S7)
VN-Member 3	L2-L4
VN-Member 4	L3-L8 (via S10)

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

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

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

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

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

```
               +--------+                                +--------+
               |  CNC   |                                |  MDSC  |
               +--------+                                +--------+
                    |                                         |
                    |                                         |
   CNC POST ACTN VN |                                         |
   Identifying AP,  |                                         |
   VNAP and VN-     |  POST /ACTN VN                          | MDSC populates
   Members          |---------------------------------------->| a single Abst.
                    |                       HTTP 200          | node topology
                    |<----------------------------------------| by itself
                    |                                         |
   CNC POST the ACTN|  POST /ACTN VN                          |
   VN identifying   |---------------------------------------->|
   AP, VNAP and VN- |                                         |
   Members and maps |                                         |
   to the TE-topo   |                         HTTP 200        |
                    |<----------------------------------------|
                    |                                         |
                    |                                         |
   CNC GET the ACTN |  GET /ACTN VN                           |
   VN YANG status   |---------------------------------------->|
                    |                                         |
                    |  HTTP 200 (ACTN VN with status)         |
                    |<----------------------------------------|
                    |                                         |
```

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

4.1. Customer view of VN

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

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

4.2. Innovative Services

4.2.1. VN Compute

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

4.2.2. Multi-sources and Multi-destinations

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

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

4.2.3. Others

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

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

4.3. Summary

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

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

5. ACTN VN YANG Model (Tree Structure)

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

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

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

6. ACTN-VN YANG Code

The YANG code is as follows:

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

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

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

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

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

    organization
        "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";
    contact
        "Editor: Young Lee <leeyoung@huawei.com>
               : Dhruv Dhody <dhruv.ietf@gmail.com>";
    description
        "This module contains a YANG module for the ACTN VN. It
        describes a VN operation module that takes place in the
        context of the CNC-MDSC Interface (CMI) of the ACTN
        architecture where the CNC is the actor of a VN
        Instantiation/modification /deletion.";
    revision 2018-02-27 {
        description
            "initial version.";
        reference
            "TBD";
    }
    /*
     * Features
     */
    feature multi-src-dest {
        description
          "Support for selection of one src or destination
          among multiple.";
    }

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

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

     typedef vn-disjointness {
          type bits {
               bit node {
                    position 0;
                    description "node disjoint";
              }
               bit link {
                    position 1;
                    description "link disjoint";
               }
               bit srlg {
                    position 2;
                    description "srlg disjoint";
               }
          }
          description
              "type of the resource disjointness for
               VN level applied across all VN members
               in a VN";
     }

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

        leaf max-bandwidth {
            type te-types:te-bandwidth;
            description
                "max bandwidth of the AP";
        }
        leaf avl-bandwidth {
            type te-types:te-bandwidth;
            description
                "available bandwidth of the AP";
        }
        /*add details and any other properties of AP,
        not associated by a VN
        CE port, PE port etc.
        */
        list vn-ap {
             key vn-ap-id;
             uses vn-ap;
             description
                 "list of VNAP in this AP";
        }
    }//access-point
    grouping vn-member {
        description
            "vn-member is described by this container";
        leaf vn-member-id {
            type uint32;
            description
                "vn-member identifier";
        }
        container src
        {
            description
                "the source of VN Member";
            leaf src {
                type leafref {
                    path "/actn/ap/access-point-list/access-point-id";
                }
                description
                    "reference to source AP";
            }
            leaf src-vn-ap-id{
                type leafref {
                    path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
                }
                description
                    "reference to source VNAP";
            }
            leaf multi-src {
                if-feature multi-src-dest;
                type boolean;
                description
                    "Is source part of multi-source, where
                    only one of the source is enabled";
            }
        }
        container dest
        {
            description
                "the destination of VN Member";
            leaf dest {
                type leafref {
                    path "/actn/ap/access-point-list/access-point-id";
                }
                description
                    "reference to destination AP";
            }
            leaf dest-vn-ap-id{
                type leafref {
                    path "/actn/ap/access-point-list/vn-ap/vn-ap-id";
                }
                description
                    "reference to dest VNAP";
            }
            leaf multi-dest {
                if-feature multi-src-dest;
                type boolean;
                description
                    "Is destination part of multi-destination, where
                    only one of the destination is enabled";
            }
        }
        leaf connetivity-matrix-id{
            type leafref {
                path "/nw:networks/nw:network/nw:node/tet:te/"
                + "tet:te-node-attributes/"
                + "tet:connectivity-matrices/"
                + "tet:connectivity-matrix/tet:id";
            }
            description
                "reference to connetivity-matrix";
        }
    }//vn-member
    /*
    grouping policy {
        description
            "policy related to vn-member-id";
        leaf local-reroute {
            type boolean;
            description
                "Policy to state if reroute
                can be done locally";
        }
        leaf push-allowed {
            type boolean;
            description
                "Policy to state if changes
                can be pushed to the customer";
        }
        leaf incremental-update {
            type boolean;
            description
                "Policy to allow only the
                changes to be reported";
        }
    }//policy
    */
      grouping vn-policy {
           description
                 "policy for VN-level diverisity";
           leaf vn-level-diversity {
                    type vn-disjointness;
                    description
                         "the type of disjointness on the VN level
                         (i.e., across all VN members)";
            }
      }
    /*
    grouping metrics-op {
        description
            "metric related information";
        list metric{
            key "metric-type";
            config false;
            description
                "The list of metrics for VN";
            leaf metric-type {
                type identityref {
                    base te-types:path-metric-type;
                }
                description
                    "The VN metric type.";
            }
            leaf value{
                type uint32;
                description
                    "The limit value";
            }
        }
    }
    */
    /*
    grouping metrics {
        description
            "metric related information";
        list metric{
            key "metric-type";
            description
                "The list of metrics for VN";
            uses te:path-metrics-bounds_config;
            container optimize{
                description
                    "optimizing constraints";
                leaf enabled{
                    type boolean;
                    description
                        "Metric to optimize";
                 }
                 leaf value{
                     type uint32;
                     description
                        "The computed value";
                }
            }
        }
    }
    */
    /*
    grouping service-metric {
        description
            "service-metric";
        uses te:path-objective-function_config;
        uses metrics;
        uses te-types:common-constraints_config;
        uses te:protection-restoration-params_config;
        uses policy;
    }//service-metric
    */
    /*
     * Configuration data nodes
     */
    container actn {
        description
            "actn is described by this container";
        container ap {
            description
                "AP configurations";
         list access-point-list {
                key "access-point-id";
                description
                    "access-point identifier";
                uses access-point{
                    description
                        "access-point information";
                }
         }
        }
        container vn {
            description
                "VN configurations";
            list vn-list {
                key "vn-id";
                description
                    "a virtual network is identified by a vn-id";
                leaf vn-id {
                    type uint32;
                    description
                        "a unique vn identifier";
                }
                leaf vn-name {
                    type string;
                    description "vn name";
                }
                leaf vn-topology-id{
                    type te-types:te-topology-id;
                    description
                        "An optional identifier to the TE Topology
                         Model where the abstract nodes and links
                         of the Topology can be found for Type 2
                         VNS";
                }
                leaf abstract-node {
                     type leafref {
                       path "/nw:networks/nw:network/nw:node/"
                            + "tet:te-node-id";
                     }
                     description
                       "a reference to the abstract node in TE
                        Topology";
                }
                list vn-member-list{
                    key "vn-member-id";
                    description
                        "List of VN-members in a VN";
                    uses vn-member;
                    /*uses metrics-op;*/
                    leaf oper-status {
                        type identityref {
                            base vn-state-type;
                        }
                        config false;
                        description
                            "VN-member operational state.";
                    }

                }
                  leaf if-selected{
                    if-feature multi-src-dest;
                      type boolean;
                      default false;
                    config false;
                      description
                          "Is the vn-member is selected among the
                           multi-src/dest options";
                  }
                /*
                container multi-src-dest{
                    if-feature multi-src-dest;
                    config false;
                    description
                        "The selected VN Member when multi-src
                        and/or mult-destination is enabled.";
                    leaf selected-vn-member{
                        type leafref {
                            path "/actn/vn/vn-list/vn-member-list"
                                 + "/vn-member-id";
                        }
                        description
                            "The selected VN Member along the set
                            of source and destination configured
                            with multi-source and/or multi-destination";
                    }
                }
                */
                /*uses service-metric;*/
                leaf admin-status {
                    type identityref {
                        base vn-admin-state-type;
                    }
                    default vn-admin-state-up;
                    description "VN administrative state.";
                }
                leaf oper-status {
                    type identityref {
                        base vn-state-type;
                    }
                    config false;
                    description "VN operational state.";
                }
                    uses vn-policy;
            }//vn-list
        }//vn
    }//actn
    /*
    * Notifications - TBD
    */
    /*
    * RPC
    */
    rpc  vn-compute{
        description
            "The VN computation without actual
            instantiation";
        input {
            leaf abstract-node {
                type leafref {
                  path "/nw:networks/nw:network/nw:node/"
                       + "tet:te-node-id";
                  }
                  description
                       "a reference to the abstract node in TE
                        Topology";
            }
            list vn-member-list{
                key "vn-member-id";
                description
                    "List of VN-members in a VN";
                uses vn-member;
            }
               uses vn-policy;
            /*uses service-metric;*/
        }
        output {
            list vn-member-list{
                key "vn-member-id";
                description
                    "List of VN-members in a VN";
                uses vn-member;
                  leaf if-selected{
                    if-feature multi-src-dest;
                      type boolean;
                      default false;
                      description
                          "Is the vn-member is selected among
                           the multi-src/dest options";
                  }
                /*uses metrics-op;*/
                leaf compute-status {
                    type identityref {
                        base vn-compute-state-type;
                    }
                    description
                        "VN-member compute state.";
                }
            }
            /*
            container multi-src-dest{
                if-feature multi-src-dest;
                description
                    "The selected VN Member when multi-src
                    and/or mult-destination is enabled.";
                leaf selected-vn-member-id{
                    type uint32;
                    description
                        "The selected VN Member-id from the
                        input";
                }
            }*/
        }
    }
}
```

<CODE ENDS>

7. JSON Example

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

The example in this section includes following VN

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

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

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

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

7.1. ACTN VN JSON

```
{
   "actn":{
      "ap":{
         "access-point-list": [
              {
               "access-point-id": 101,
               "access-point-name": "101",
               "vn-ap": [
                  {
                     "vn-ap-id": 10101,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "1-0-1"
                  },
                  {
                     "vn-ap-id": 10102,
                     "vn": 2,
                     "abstract-node": "D2",
                     "ltp": "1-0-1"
                  },
                  {
                     "vn-ap-id": 10103,
                     "vn": 3,
                     "abstract-node": "D3",
                     "ltp": "1-0-1"
                  },
               ]
            },
            {
               "access-point-id": 202,
               "access-point-name": "202",
               "vn-ap": [
                  {
                     "vn-ap-id": 20201,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "2-0-2"
                  }
               ]
            },
            {
               "access-point-id": 303,
               "access-point-name": "303",
               "vn-ap": [
                  {
                     "vn-ap-id": 30301,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "3-0-3"
                  },
                  {
                     "vn-ap-id": 30303,
                     "vn": 3,
                     "abstract-node": "D3",
                     "ltp": "3-0-3"
                  }
               ]
            },
            {
               "access-point-id": 440,
               "access-point-name": "440",
               "vn-ap": [
                  {
                     "vn-ap-id": 44001,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "4-4-0"
                  }
               ]
            },
            {
               "access-point-id": 550,
               "access-point-name": "550",
               "vn-ap": [
                  {
                     "vn-ap-id": 55002,
                     "vn": 2,
                     "abstract-node": "D2",
                     "ltp": "5-5-0"
                  }
               ]
            },
            {
               "access-point-id": 770,
               "access-point-name": "770",
               "vn-ap": [
                  {
                     "vn-ap-id": 77001,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "7-7-0"
                  },
                  {
                     "vn-ap-id": 77003,
                     "vn": 3,
                     "abstract-node": "D3",
                     "ltp": "7-7-0"
                  }
               ]
            },
            {
               "access-point-id": 880,
               "access-point-name": "880",
               "vn-ap": [
                  {
                     "vn-ap-id": 88001,
                     "vn": 1,
                     "abstract-node": "D1",
                     "ltp": "8-8-0"
                  },
                  {
                     "vn-ap-id": 88003,
                     "vn": 3,
                     "abstract-node": "D3",
                     "ltp": "8-8-0"
                  }
               ]
            }
         ]
      },
      "vn":{
         "vn-list": [
            {
               "vn-id": 1,
               "vn-name": "vn1",
               "vn-topology-id": "te-topology:abstract1",
               "abstract-node": "D1",
               "vn-member-list": [
                  {
                     "vn-member-id": 104,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10101,
                     },
                     "dest": {
                        "dest": 440,
                        "dest-vn-ap-id": 44001,
                     },
                     "connectivity-matrix-id": 104
                  },
                  {
                     "vn-member-id": 107,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10101,
                     },
                     "dest": {
                        "dest": 770,
                        "dest-vn-ap-id": 77001,
                     },
                     "connectivity-matrix-id": 107
                  },
                  {
                     "vn-member-id": 204,
                     "src": {
                        "src": 202,
                        "dest-vn-ap-id": 20401,
                     },
                     "dest": {
                        "dest": 440,
                        "dest-vn-ap-id": 44001,
                     },
                     "connectivity-matrix-id": 204
                  },
                  {
                     "vn-member-id": 308,
                     "src": {
                        "src": 303,
                        "src-vn-ap-id": 30301,
                     },
                     "dest": {
                        "dest": 880,
                        "src-vn-ap-id": 88001,
                     },
                     "connectivity-matrix-id": 308
                  },
                  {
                     "vn-member-id": 108,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10101,
                     },
                     "dest": {
                        "dest": 880,
                        "dest-vn-ap-id": 88001,
                     },
                     "connectivity-matrix-id": 108
                  }
               ]
            },
            {
               "vn-id": 2,
               "vn-name": "vn2",
               "vn-topology-id": "te-topology:abstract2",
               "abstract-node": "D2",
               "vn-member-list": [
                  {
                     "vn-member-id": 105,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10102,
                     },
                     "dest": {
                        "dest": 550,
                        "dest-vn-ap-id": 55002,
                     },
                     "connectivity-matrix-id": 105
                  }
               ]
            },
            {
               "vn-id": 3,
               "vn-name": "vn3",
               "vn-topology-id": "te-topology:abstract3",
               "abstract-node": "D3",
               "vn-member-list": [
                  {
                     "vn-member-id": 104,
                     "src": {
                        "src": 101,
                     },
                     "dest": {
                        "dest": 440,
                        "multi-dest": true
                     }
                  },
                  {
                     "vn-member-id": 107,
                     "src": {
                        "src": 101,
                        "src-vn-ap-id": 10103,
                     },
                     "dest": {
                        "dest": 770,
                        "dest-vn-ap-id": 77003,
                        "multi-dest": true
                     },
                     "connectivity-matrix-id": 107,
                     "if-selected":true,
                  },
                  {
                     "vn-member-id": 204,
                     "src": {
                        "src": 202,
                        "multi-src": true,
                     },
                     "dest": {
                        "dest": 440,
                     },
                  },
                  {
                     "vn-member-id": 304,
                     "src": {
                        "src": 303,
                        "src-vn-ap-id": 30303,
                        "multi-src": true,
                     },
                     "dest": {
                        "dest": 440,
                        "src-vn-ap-id": 44003,
                     },
                     "connectivity-matrix-id": 304,
                     "if-selected":true,
                  },
               ]
            },

         ]
      }

   }
}
```

7.2. TE-topology JSON

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

                        },
                        "connectivity-matrix": [
                          {
                            "id": 104,
                            "from": "1-0-1",
                            "to": "4-4-0"
                          },
                          {
                            "id": 107,
                            "from": "1-0-1",
                            "to": "7-7-0"
                          },
                          {
                            "id": 204,
                            "from": "2-0-2",
                            "to": "4-4-0"
                          },
                          {
                            "id": 308,
                            "from": "3-0-3",
                            "to": "8-8-0"
                          },
                          {
                            "id": 108,
                            "from": "1-0-1",
                            "to": "8-8-0"
                          },
                        ]
                      }
                    }
                  },
                  "termination-point": [
                    {
                      "tp-id": "1-0-1",
                      "te-tp-id": 10001,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                            }
                        ]
                      }
                    },
                    {
                      "tp-id": "1-1-0",
                      "te-tp-id": 10100,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-0-2",
                      "te-tp-id": 20002,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-2-0",
                      "te-tp-id": 20200,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-0-3",
                      "te-tp-id": 30003,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-3-0",
                      "te-tp-id": 30300,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-0-4",
                      "te-tp-id": 40004,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-4-0",
                      "te-tp-id": 40400,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-0-5",
                      "te-tp-id": 50005,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-5-0",
                      "te-tp-id": 50500,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-0-6",
                      "te-tp-id": 60006,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-6-0",
                      "te-tp-id": 60600,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-0-7",
                      "te-tp-id": 70007,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-7-0",
                      "te-tp-id": 70700,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-0-8",
                      "te-tp-id": 80008,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-8-0",
                      "te-tp-id": 80800,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    }
                  ]
                }
              ]
            },
            {
              "network-types": {
                "te-topology": {}
              },
              "network-id": "abstract2",
              "provider-id": 201,
              "client-id": 600,
              "te-topology-id": "te-topology:abstract2",
              "node": [
               {
                  "node-id": "D2",
                  "te-node-id": "2.0.1.2",
                  "te": {
                    "te-node-attributes": {
                      "domain-id" : 1,
                      "is-abstract": [null],
                      "connectivity-matrices": {
                        "is-allowed": true,
                        "underlay": {
                           "enabled": true
                        },
                        "path-constraints": {
                          "bandwidth-generic": {
                            "te-bandwidth": {
                              "generic": [
                                {
                                  "generic": "0x1p10"
                                }
                              ]
                            }
                          }
                        },
                        "optimizations": {
                           "objective-function": {
                               "objective-function-type": "of-maximize-residual-
   bandwidth"
                           }
                        },
                        "connectivity-matrix": [
                          {
                            "id": 105,
                            "from": "1-0-1",
                            "to": "5-5-0",
                            "underlay": {
                               "enabled": true,
                               "primary-path": {
                                   "network-ref": "absolute",
                                   "path-element": [
                                     {
                                       "path-element-id": 1,
                                       "index": 1,
                                       "numbered-hop": {
                                         "address": "4.4.4.4",
                                         "hop-type": "STRICT"
                                        }
                                      },
                                      {
                                       "path-element-id": 2,
                                       "index": 2,
                                       "numbered-hop": {
                                         "address": "7.7.7.7",
                                         "hop-type": "STRICT"
                                       }
                                      }
                                   ]
                               }
                            }
                          }
                        ]
                      }
                    }
                  },
                  "termination-point": [
                    {
                      "tp-id": "1-0-1",
                      "te-tp-id": 10001,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                            }
                        ]
                      }
                    },
                    {
                      "tp-id": "1-1-0",
                      "te-tp-id": 10100,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-0-2",
                      "te-tp-id": 20002,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-2-0",
                      "te-tp-id": 20200,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-0-3",
                      "te-tp-id": 30003,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-3-0",
                      "te-tp-id": 30300,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-0-4",
                      "te-tp-id": 40004,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-4-0",
                      "te-tp-id": 40400,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-0-5",
                      "te-tp-id": 50005,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-5-0",
                      "te-tp-id": 50500,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-0-6",
                      "te-tp-id": 60006,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-6-0",
                      "te-tp-id": 60600,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-0-7",
                      "te-tp-id": 70007,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-7-0",
                      "te-tp-id": 70700,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-0-8",
                      "te-tp-id": 80008,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-8-0",
                      "te-tp-id": 80800,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    }
                  ]
                }
              ]
            },
            {
              "network-types": {
                "te-topology": {}
              },
              "network-id": "abstract3",
              "provider-id": 201,
              "client-id": 600,
              "te-topology-id": "te-topology:abstract3",
              "node": [
                {
                  "node-id": "D3",
                  "te-node-id": "3.0.1.1",
                  "te": {
                    "te-node-attributes": {
                      "domain-id" : 3,
                      "is-abstract": [null],
                      "connectivity-matrices": {
                        "is-allowed": true,
                        "path-constraints": {
                          "bandwidth-generic": {
                            "te-bandwidth": {
                              "generic": [
                                {
                                  "generic": "0x1p10",
                                }
                              ]
                            }
                          }
                        },
                        "connectivity-matrix": [
                          {
                            "id": 107,
                            "from": "1-0-1",
                            "to": "7-7-0"
                          },
                          {
                            "id": 308,
                            "from": "3-0-3",
                            "to": "8-8-0"
                          },
                        ]
                      }
                    }
                  },
                  "termination-point": [
                    {
                      "tp-id": "1-0-1",
                      "te-tp-id": 10001,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                            }
                        ]
                      }
                    },
                    {
                      "tp-id": "1-1-0",
                      "te-tp-id": 10100,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                              "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-0-2",
                      "te-tp-id": 20002,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "2-2-0",
                      "te-tp-id": 20200,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-0-3",
                      "te-tp-id": 30003,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "3-3-0",
                      "te-tp-id": 30300,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-0-4",
                      "te-tp-id": 40004,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "4-4-0",
                      "te-tp-id": 40400,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-0-5",
                      "te-tp-id": 50005,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "5-5-0",
                      "te-tp-id": 50500,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-0-6",
                      "te-tp-id": 60006,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "6-6-0",
                      "te-tp-id": 60600,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-0-7",
                      "te-tp-id": 70007,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "7-7-0",
                      "te-tp-id": 70700,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-0-8",
                      "te-tp-id": 80008,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    },
                    {
                      "tp-id": "8-8-0",
                      "te-tp-id": 80800,
                      "te": {
                        "interface-switching-capability": [
                          {
                            "switching-capability": "switching-otn",
                            "encoding": "lsp-encoding-oduk"
                          }
                        ]
                      }
                    }
                  ]
                }
              ]
            },
          ]
        }
      }
```

8. Security Considerations

TDB

9. IANA Considerations

TDB

10. Acknowledgments

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

11. References

11.1. Normative References

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

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

11.2. Informative References

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

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

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

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

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

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

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

12. Contributors

Contributor's Addresses

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

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

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

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

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

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

Authors' Addresses

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

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

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

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

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

TEAS WG
Internet Draft
Intended status: standard track
Expires: June 30, 2019

Young Lee
Dhruv Dhody
Huawei

Daniele Ceccarelli
Ericsson

Jeff Tantsura
Apstra

Giuseppe Fioccola
Huawei

Qin Wu
Huawei

December 30, 2018

Traffic Engineering and Service Mapping Yang Model

draft-lee-teas-te-service-mapping-yang-13

Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPM Service Model) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Abstraction and Control of Traffic Engineered Networks Virtual Network model). This model is referred to as TE Service Mapping Model and is applicable generically to the operator’s need for seamless control and management of their VPN services with TE tunnel support.

The model is principally used to allow monitoring and diagnostics of the management systems to show how the service requests are mapped onto underlying network resource and TE models.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modeling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [RFC8199] and [RFC8309].

Framework for Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453] introduces an architecture to support virtual network services and connectivity services. [ACTN-VN-YANG] defines a YANG model and describes how customers or end-to-end orchestrators can request and/or instantiate a generic virtual network service. [ACTN-Applicability] describes the way IETF YANG models of different classifications can be applied to the ACTN interfaces. In particular, it describes how customer service models can be mapped into the CNC-MDSC Interface (CMI) of the ACTN architecture.

The models presented in this document are also applicable in generic context [RFC8309] as part of Customer Service Model used between Service Orchestrator and Customer.

[RFC8299] provides a L3VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[L2SM] provides a L2VPN service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

[L1CSM] provides a L1 connectivity service delivery YANG model for PE-based VPNs. The scope of that draft is limited to a set of domains under control of the same network operator to deliver services requiring TE tunnels.

While the IP/MPLS Provisioning Network Controller (PNC) is responsible for provisioning the VPN service on the Provider Edge (PE) nodes, the Multi-Domain Service Coordinator (MDSC) can coordinate how to map the VPN services onto Traffic Engineering (TE) tunnels. This is consistent with the two of the core functions of the MDSC specified in [RFC8453]:

- Customer mapping/translation function: This function is to map customer requests/commands into network provisioning requests that can be sent to the PNC according to the business policies that have been provisioned statically or dynamically. Specifically, it provides mapping and translation of a customer’s service request into a set of parameters that are specific to a network type and technology such that the network configuration process is made possible.

- Virtual service coordination function: This function translates customer service-related information into virtual network service operations in order to seamlessly operate virtual networks while meeting a customer’s service requirements. In the context of ACTN, service/virtual service coordination includes a number of service orchestration functions such as multi-destination load balancing, guarantees of service quality, bandwidth and throughput. It also includes notifications for service fault and performance degradation and so forth.

Section 2 describes a set of TE & service related parameters that this document addresses as new and advanced parameters that are not included in generic service models. Section 3 discusses YANG modeling approach.

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
tsm-types	ietf-te-service-mapping-types	[RFCXXXX}
l1	ietf-l1csm	[L1CSM]
l2vpn-svc	ietf-l2vpn-svc	[L2SM]
l3vpn-svc	ietf-l3vpn-svc	[RFC8299]
l1-tsm	ietf-l1csm-te-service-mapping	[RFCXXXX]
l2-tsm	ietf-l2sm-te-service-mapping	[RFCXXXX]
l3-tsm	ietf-l3sm-te-service-mapping	[RFCXXXX]
vn	ietf-vn	[ACTN-VN]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[TE-Types]
te-topo	ietf-te-topology	[TE-Topo]
te	ietf-te	[TE-Tunnel]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor will replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. TE & Service Related Parameters

While L1/2/3 service models [L1CSM, L2SM, L3SM] are intended to provide service-specific parameters for VPN service instances, there are a number of TE & Service related parameters that are not included in the generic service models.

Additional service parameters and policies that are not included in the aforementioned service models are addressed in the YANG models defined in this document.

2.1. VN/Tunnel Selection Requirements

In some cases, the service requirements may need addition TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example, for network slicing. The establishment of TE tunnels is subject to the network operator’s policies.

To summarize, there are three modes of VN/Tunnel selection operations to be supported as follows. Additional modes may be defined in the future.

- New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN.

  Under this mode, the following sub-categories can be supported:

  1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.

  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.

  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.

- VN/Tunnel Sharing - A customer could request a VPN service where new tunnels (or a VN) do not need to be created for each VPN and can be shared across multiple VPNs. Further, the mapping YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the tunnels in use. No modification of the properties of a tunnel (or VN) is allowed in this mode: an existing tunnel can only be selected.

- VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

2.2. Availability Requirement

Availability is another service requirement or intent that may influence the selection or provisioning of TE tunnels or a VN to support the requested service. Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure.

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this draft.

3. YANG Modeling Approach

This section provides how the TE & Service mapping parameters are supported using augmentation of the existing service models (i.e., [L1CSM], [L2SM], and [L3SM]). Figure 1 shows the scope of the Augmented LxSM Model.

```
   +--------------+        +-----------------------------+           +----------+
   |    LxSM      |o-------|                             | . . . . . | ACTN VN  |
   +--------------+ augment|                             |           +----------+
                           |                             |           +----------+
   +--------------+        |    Augmented LxSM Model     | . . . . . | TE-topo  |
   | TE & Service |------->|                             |           +----------+
   | Mapping Types| import |                             |           +----------+
   +--------------+        |                             | . . . . . | TE-tunnel|
                           +-----------------------------+ reference +----------+
```

Figure 1. Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE & Service related parameters (defined in Section 2) grouping information from TE & Service Mapping Types. The TE & Service Mapping Types (ietf-te-service-mapping-types) module is the repository of all common groupings imported by each augmented LxSM model. Any future service models would import this grouping file.

The role of the augmented LxSm service model is to expose the mapping relationship between service models and TE models so that VN/VPN service instantiations provided by the underlying TE networks can be viewed outside of the MDSC, for example by an operator who is diagnosing the behavior of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various constraints via TE-topology model [TE-topo], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE & Service Mapping Types.

It is possible that new service models will be defined at some future time and that it will be desirable to map them to underlying TE constructs in the same way as the three existing models are augmented.

4. L3VPN Architecture in the ACTN Context

Figure 2 shows the architectural context of this document referencing the ACTN components and interfaces.

```
                              +----------------------------+
                              |  Customer Service Manager  |
                              |  +-----------------------+ |
                              |  |           CNC           + |
                              |  +-+-------------------+-+ |
                              +----|-------------------|---+
                                   |                   |
                                   |CMI(Augmented L3SM)|CMI(VN)
                                   |                   |
                  +----------------|-------------------|----+
                  | +--------------|-----------------+ |    |
                  | | MDSC         |                 | |    |
                  | |              |                 | |    |
                  | |  +-----------+--------------+  | |    |
      TE-Svc-Map<------+ Service Mapping Function |  | |    |
                  | |  +-----------+--------------+  | |    |
                  | |              |                 | |    |
                  | +-------+------|-----------------+ |    |
                  |         |      |                   |    |
                  |         |      |CMI(VN)            |    |
                  |         |      |                   |    |
                  |         |   +--|-------------------|--+ |
                  |         |   |  |        MDSC       |  | |
                  |         |   | ++-------------------++ | |
                  |         |   | +   Service Mapping   +---->TE-Svc-Map
                  |         |   | ++----------+---------+ | |
                  |         |   +--|----------|-----------+ |
                  +---------|------|----------|-------------+
                            |      |          |
                            | +----+--------+ |
                            | |             | |
        MPI(VPN / TE models)| |             | |MPI(TE / L1 models)
                            | |             | |
                      +-----|-|---+   +-----|-|----+
           IP/MPLS    |  +--+-+-+ |   |  +--+-+-+  | Optical Domain
           Domain     |  | PNC1 | |   |  | PNC2 |  | Controller
           Controller |  +--+---+ |   |  +--+---+  |
                      +-----|-----+   +-----|------+
                            |               |
                            V               | SBI
                +---------------------+     |
               /    IP/MPLS Network    \    |
              +-------------------------+   |
                                            V
                                 +---------------------+
                                /    Optical Network    \
                               +-------------------------+
```

Figure 2: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 2.

. CNC: The Customer Network Controller is responsible for generating service requests. In the context of an L3VPN, the CNC uses the Augmented L3SM to express the service request and communicate it to the network operator.

- MDSC: This entity is responsible for coordinating a L3VPN service request (expressed via the Augmented L3SM) with the IP/MPLS PNC and the Transport PNC. For TE services, one of the key responsibilities of the MDSC is to coordinate with both the IP PNC and the Transport PNC for the mapping of the Augmented L3VPN Service Model to the ACTN VN model. In the VN/TE-tunnel binding case, the MDSC will need to coordinate with the Transport PNC to dynamically create the TE-tunnels in the transport network as needed. These tunnels are added as links in the IP/MPLS Layer topology. The MDSC coordinates with IP/MPLS PNC to create the TE-tunnels in the IP/MPLS layer, as part of the ACTN VN creation.
- PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 2 shows two distinct PNCs.
  - IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.
  - Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

There are four main interfaces shown in Figure 2.

- CMI: The CNC-MDSC Interface is used to communicate service requests from the customer to the operator. The requests may be expressed as Augmented VPN service requests (L2SM, L3SM), as connectivity requests (L1CSM), or as virtual network requests (ACTN VN).
- MPI: The MDSC-PNC Interface is used by the MDSC to orchestrate networks under the control of PNCs. The requests on this interface may use TE tunnel models, TE topology models, VPN network configuration models or layer one connectivity models.
- SBI: The Southbound Interface is used by the PNC to control network devices and is out of scope for this document.
- The TE Service Mapping Model as described in this document can be used to see the mapping between service models and VN models and TE Tunnel/Topology models. That mapping may occur in the CNC if a service request is mapped to a VN request. Or it may occur in the MDSC where a service request is mapped to a TE tunnel, TE topology, or VPN network configuration model. The TE Service Mapping Model may be read from the CNC or MDSC to understand how the mapping has been made and to see the purpose for which network resources are used.

As shown in Figure 2, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

The high-level control flows for one example are as follows:

1. A customer asks for an L3VPN between CE1 and CE2 using the Augmented L3SM model.

2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [ACTN-VN-YANG] is used to configure a new VN based on this VPN and map the VPN service to the ACTN VN. In case an existing tunnel is to be used, each device will select which tunnel to use and populate this mapping information.

3. The MDSC interacts with both the IP/MPLS PNC and the Transport PNC to create a PE-PE tunnel in the IP network mapped to a TE tunnel in the transport network by providing the inter-layer access points and tunnel requirements. The specific service information is passed to the IP/MPLS PNC for the actual VPN configuration and activation.

   a. The Transport PNC creates the corresponding TE tunnel matching with the access point and egress point.
   b. The IP/MPLS PNC maps the VPN ID with the corresponding TE tunnel ID to bind these two IDs.

4. The IP/MPLS PNC creates/updates a VRF instance for this VPN customer. This is not in the scope of this document.

4.1. Service Mapping

Augmented L3SM and L2SM can be used to request VPN service creation including the creation of sites and corresponding site network access connection between CE and PE. A VPN-ID is used to identify each VPN service ordered by the customer. The ACTN VN can be used further to establish PE-to-PE connectivity between VPN sites belonging to the same VPN service. A VN-ID is used to identify each virtual network established between VPN sites.

Once the ACTN VN has been established over the TE network (maybe a new VN, maybe modification of an existing VN, or maybe the use of an unmodified existing VN), the mapping between the VPN service and the ACTN VN service can be created.

4.2. Site Mapping

The elements in Augmented L3SM and L2SM define site location parameters and constraints such as distance and access diversity that can influence the placement of network attachment points (i.e, virtual network access points (VNAP)). To achieve this, a central directory can be set up to establish the mapping between location parameters and constraints and network attachment point location. Suppose multiple attachment points are matched, the management system can use constraints or other local policy to select the best candidate network attachment points.

After a network attachment point is selected, the mapping between VPN site and VNAP can be established as shown in Table 1.

Site	Site Network Access	Location (Address, Postal Code, State, City,Country Code)	Access Diversity (Constraint-Type, Group-id,Target Group-id)	PE
SITE1	ACCESS1	(,,US,NewYork,)	(10,PE-Diverse,10)	PE1
SITE2	ACCESS2	(,,CN,Beijing,)	(10,PE-Diverse,10)	PE2
SITE3	ACCESS3	(,,UK,London,)	(12,same-PE,12)	PE4
SITE4	ACCESS4	(,,FR,Paris,)	(20,Bearer-Diverse,20)	PE7

Table 1 : Mapping Between VPN Site and VNAP

5. Applicability of TE-Service Mapping in Generic context

As discussed in the Introduction Section, the models presented in this document are also applicable generically outside of the ACTN architecture. [RFC8309] defines Customer Service Model between Customer and Service Orchestrator and Service Delivery Model between Service Orchestrator and Network Orchestrator(s). TE-Service mapping models defined in this document can be regarded primarily as Customer Service Model and secondarily as Service Deliver Model.

6. YANG Data Trees

```
module: ietf-l1csm-te-service-mapping
  augment /l1:l1-connectivity/l1:services/l1:service:
    +-rw te-service-mapping!
  augment /l1:l1-connectivity/l1:services/l1:service:
    +-rw te-mapping
       +-rw map-type?              identityref
       +-rw availability-type?     identityref
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?      -> /vn:actn/vn/vn-list/vn-id
          +-:(te-topo)
          |  +-rw vn-topology-id?   te-types:te-topology-id
          |  +-rw abstract-node?    -> /nw:networks/network/node/node-id
          +-:(te-tunnel)
             +-rw te-tunnel-list*   te:tunnel-ref
  augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id
  augment /l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id



module: ietf-l2sm-te-service-mapping
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
    +-rw te-service-mapping!
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-service:
    +-rw te-mapping
       +-rw map-type?              identityref
       +-rw availability-type?     identityref
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?      -> /vn:actn/vn/vn-list/vn-id
          +-:(te-topo)
          |  +-rw vn-topology-id?   te-types:te-topology-id
          |  +-rw abstract-node?    -> /nw:networks/network/node/node-id
          +-:(te-tunnel)
             +-rw te-tunnel-list*   te:tunnel-ref
  augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site/l2vpn-svc:site-net
work-
accesses/l2vpn-svc:site-network-access:
    +-rw (te)?
       +-:(actn-vn)
       |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-id
       +-:(te)
          +-rw ltp?           te-types:te-tp-id



   module: ietf-l3sm-te-service-mapping
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
       +-rw te-service-mapping!
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-service:
       +-rw te-mapping
          +-rw map-type?                identityref
          +-rw availability-type?      identityref
          +-rw (te)?
             +-:(actn-vn)
             |  +-rw actn-vn-ref?      -> /vn:actn/vn/vn-list/vn-id
             +-:(te-topo)
             |  +-rw vn-topology-id?   te-types:te-topology-id
             |  +-rw abstract-node?    -> /nw:networks/network/node/node-id
             +-:(te-tunnel)
                +-rw te-tunnel-list*   te:tunnel-ref
     augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site/l3vpn-svc:site-
   network-accesses/l3vpn-svc:site-network-access:
       +-rw (te)?
          +-:(actn-vn)
          |  +-rw actn-vn-ref?   -> /vn:actn/ap/access-point-list/access-point-i
d
          +-:(te)
             +-rw ltp?           te-types:te-tp-id
```

7. YANG Data Models

The YANG codes are as follows:

```
   <CODE BEGINS> file "ietf-te-service-mapping-types@2018-12-30.yang"

   module ietf-te-service-mapping-types {

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

          prefix "tsm";

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

          import ietf-network {
              prefix "nw";
          }

          import ietf-te {
              prefix "te";
          }

          import ietf-vn {
              prefix "vn";
          }

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

          contact
              "Editor: Young Lee <leeyoung@huawei.com>
                       Dhruv Dhody <dhruv.ietf@gmail.com>
                      Qin Wu <bill.wu@huawei.com>";
          description
              "This module contains a YANG module for TE & Service mapping
            parameters and policies as a common grouping applicable to
            variuous service models (e.g., L1CSM, L2SM, L3SM, etc.)";

          revision 2018-12-30 {
              description
                  "initial version.";
              reference
                  "TBD";
          }


          /*
           * Identity for map-type
           */
         identity map-type {
            description
            "Base identity from which specific map types are
             derived.";
         }

         identity new {
            base map-type;
            description
              "The new VN/tunnels are binded to the service.";
         }
         identity detnet-hard-isolation {
            base new;
            description
              "Hard isolation with deterministic characteristics.";
         }

         identity hard-isolation {
            base new;
            description
              "Hard isolation.";
         }

         identity soft-isolation {
            base new;
            description
              "Soft-isolation.";
         }

         identity select {
            base map-type;
            description
              "The VPN service selects an existing tunnel with no
               modification.";
         }

         identity modify {
            base map-type;
            description
              "The VPN service selects an existing tunnel and allows
               to modify the properties of the tunnel (e.g., b/w)";
         }

          /*
           * Identity for availability-type
           */
         identity availability-type {
            description
              "Base identity from which specific map types are
               derived.";
         }

         identity level-1 {
            base availability-type;
            description
              "level 1: 99.9999%";
         }

         identity level-2 {
            base availability-type;
            description
              "level 2: 99.999%";
         }

         identity level-3 {
            base availability-type;
            description
              "level 3: 99.99%";
         }

         identity level-4 {
            base availability-type;
            description
              "level 4: 99.9%";
         }

         identity level-5 {
            base availability-type;
            description
              "level 5: 99%";
         }

          /*
           * Groupings
           */

         grouping te-ref {
            description
              "The reference to TE.";
            choice te {
               description
                  "The TE";
                  case actn-vn {
                      leaf actn-vn-ref {
                          type leafref {
                              path "/vn:actn/vn:vn/vn:vn-list/vn:vn-id";
                          }
                          description
                              "The reference to ACTN VN";
                      }
                  }
                  case te-topo {
                     leaf vn-topology-id{
                         type te-types:te-topology-id;
                         description
                             "An identifier to the TE Topology Model
                              where the abstract nodes and links of
                              the Topology can be found for Type 2
                              VNS";
                     }
                     leaf abstract-node {
                       type leafref {
                         path "/nw:networks/nw:network/nw:node/"
                         + "nw:node-id";
                       }
                       description
                         "a reference to the abstract node in TE
                         Topology";
                     }
                  }
                  case te-tunnel {
                      leaf-list te-tunnel-list {
                          type te:tunnel-ref;
                          description
                              "Reference to TE Tunnels";
                      }

                  }

              }

          }

          grouping te-endpoint-ref {
              description
                 "The reference to TE endpoints.";
              choice te {
                 description
                    "The TE";
                 case actn-vn {
                    leaf actn-vn-ref {
                          type leafref {
                              path "/vn:actn/vn:ap/vn:access-point-list"
                              + "/vn:access-point-id";
                          }
                          description
                              "The reference to ACTN VN";
                      }
                 }
                 case te {
                    leaf ltp {
                         type te-types:te-tp-id;
                         description
                             "Reference LTP in the TE-topology";
                    }
                 }
              }
          }

          grouping te-mapping {
              description
                  "Mapping between Services and TE";
              container te-mapping {
                  description
                      "Mapping between Services and TE";
                  leaf map-type {
                     type identityref {
                base map-type;
                   }
                     description
                       "Isolation Requirements, Tunnel Bind or
                        Tunnel Selection";
                  }
                  leaf availability-type {
                     type identityref {
                       base availability-type;
                     }
                     description
                       "Availability Requirement for the Service";
                  }
                  uses te-ref;
              }
          }

   }
   <CODE ENDS>


   <CODE BEGINS> file "ietf-l1csm-te-service-mapping@2018-10-05.yang"



      module ietf-l1csm-te-service-mapping {

          namespace "urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping";

          prefix "tm";

          import ietf-te-service-mapping-types {
             prefix "tsm-types";
         }

         import ietf-l1csm {
              prefix "l1";
          }
```

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

contact
    "Editor: Young Lee <leeyoung@huawei.com>
             Dhruv Dhody <dhruv.ietf@gmail.com>
   Qin Wu <bill.wu@huawei.com>";
description
    "This module contains a YANG module for the mapping of
    Layer 1 Connectivity Service Module (L1CSM) to the TE and VN ";

revision 2018-10-05 {
    description
        "initial version.";
    reference
        "TBD";
}

/*
 * Configuration data nodes
 */
augment "/l1:l1-connectivity/l1:services/l1:service" {
  description
    "l1csm augmented to include TE parameters and mapping";
  container te-service-mapping {
     presence "indicates l1 service to te mapping";
     description
       "Container to augment l1csm to TE parameters and mapping";
   }
}

augment "/l1:l1-connectivity/l1:services/l1:service" {
   description
     "This augment is only valid for TE mapping --
     te mapping is added";
   uses tsm-types:te-mapping;
}

augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-1" {
  description
     "This augment is only valid for TE mapping --
      endpoint-1 te-reference is added";
  uses tsm-types:te-endpoint-ref;
}

augment "/l1:l1-connectivity/l1:services/l1:service/l1:endpoint-2" {
  description
```

```
             "This augment is only valid for TE mapping --
             endpoint-2 te-reference is added";
           uses tsm-types:te-endpoint-ref;
         }
     }


 <CODE ENDS>


 <CODE BEGINS> file "ietf-l2sm-te-service-mapping@2018-10-05.yang"



    module ietf-l2sm-te-service-mapping {

        namespace "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";

        prefix "tm";

        import ietf-te-service-mapping-types {
          prefix "tsm-types";
        }

        import ietf-l2vpn-svc {
           prefix "l2vpn-svc";
        }

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

        contact
            "Editor: Young Lee <leeyoung@huawei.com>
                     Dhruv Dhody <dhruv.ietf@gmail.com>
                   Qin Wu <bill.wu@huawei.com>";
        description
            "This module contains a YANG module for the mapping of
            Layer 2 Service Model (L1CSM) to the TE and VN ";

        revision 2018-10-05 {
            description
                "initial version.";
            reference
                "TBD";
        }
```

```
          /*
           * Configuration data nodes
           */
        augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
            description
              "l2sm augmented to include TE parameters and mapping";
            container te-service-mapping {
               presence "indicates l2 service to te mapping";
               description
                 "Container to augment l2sm to TE parameters and mapping";
             }
        }

        augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/l2vpn-svc:vpn-servi
ce" {
            description
              "This augment is only valid for TE mapping --
               te mapping is added";
            uses tsm-types:te-mapping;
        }

        augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
            +"/l2vpn-svc:site-network-accesses/l2vpn-svc:site-network-access" {
              description
                 "This augment is only valid for TE mapping --
                  network-access te-reference is added";
              uses tsm-types:te-endpoint-ref;
          }
      }



   <CODE ENDS>


   <CODE BEGINS> file "ietf-l3sm-te-service-mapping@2018-10-05.yang"




   module ietf-l3sm-te-service-mapping {

       namespace "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";

       prefix "tm";

       import ietf-te-service-mapping-types {
         prefix "tsm-types";
       }

       import ietf-l3vpn-svc {
          prefix "l3vpn-svc";
       }

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

       contact
           "Editor: Young Lee <leeyoung@huawei.com>
                    Dhruv Dhody <dhruv.ietf@gmail.com>
                   Qin Wu <bill.wu@huawei.com>";
       description
           "This module contains a YANG module for the mapping of
           Layer 3 Service Model (L3SM) to the TE and VN ";

       revision 2018-10-05 {
           description
               "initial version.";
           reference
               "TBD";
       }

       /*
        * Configuration data nodes
        */
       augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
         description
            "l3sm augmented to include TE parameters and mapping";
         container te-service-mapping {
            presence "indicates l3 service to te mapping";
            description
              "Container to augment l3sm to TE parameters and mapping";
         }
       }

       augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services/l3vpn-svc:vpn-servic
e" {
          description
            "This augment is only valid for TE mapping --
             te mapping is added";
          uses tsm-types:te-mapping;
       }

       augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
         +"/l3vpn-svc:site-network-accesses/l3vpn-svc:site-network-access" {
```

```
         description
            "This augment is only valid for TE mapping --
             network-access te-reference is added";
         uses tsm-types:te-endpoint-ref;
       }
   }


   <CODE ENDS>
```

8. Security

The configuration, state, and action data defined in this document are designed to be accessed via a management protocol with a secure transport layer, such as NETCONF [RFC6241]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true") These data nodes may be considered sensitive or vulnerable in some network environments.

9. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

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

   --------------------------------------------------------------------
   URI: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping
   Registrant Contact: The IESG.
   XML: N/A, the requested URI is an XML namespace.
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   URI: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
   Registrant Contact: The IESG.
   XML: N/A, the requested URI is an XML namespace.
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   URI: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping
   Registrant Contact: The IESG.
   XML: N/A, the requested URI is an XML namespace.
   --------------------------------------------------------------------
```

This document registers the following YANG modules in the YANG Module.

Names registry [RFC7950]:

```
   --------------------------------------------------------------------
   name:          ietf-te-service-mapping-types
   namespace:     urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-
   types
   reference:     RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:          ietf-l1csm-te-service-mapping
   namespace:     urn:ietf:params:xml:ns:yang:ietf-l1cms-te-service-
   mapping
   reference:     RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:          ietf-l2sm-te-service-mapping
   namespace:     urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-
   mapping
   reference:     RFC XXXX (TDB)
   --------------------------------------------------------------------

   --------------------------------------------------------------------
   name:          ietf-l3sm-te-service-mapping
   namespace:     urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-
   mapping
   reference:     RFC XXXX (TDB)
   --------------------------------------------------------------------
```

10. Acknowledgements

We thank Diego Caviglia and Igor Bryskin for useful discussions and motivation for this work.

11. References

11.1. Informative References

[RFC4110] R. Callon and M. Suzuki, "A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)", RFC 4110, July 2005.

[RFC6020] M. Bjorklund, Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC8309] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", RFC 8309, January 2018.

[RFC8199] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", RFC 8199, July 2017.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.

[RFC8453] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

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

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[TE-Types] T. Saad (Editor), "Traffic Engineering Common YANG Types", draft-ietf-teas-yang-te-types, work in progress.

[ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[ACTN-Applicability] Y. Lee, et al, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks, draft-ietf-teas-actn-yang, work in progress.

[RFC8299] Q. Wu, S. Litkowski, L.Tomotaki, and K. Ogaki, "YANG Data Model for L3VPN service delivery", RFC 8299, January 2018.

[L2SM] B. Wen, et al, "A YANG Data Model for L2VPN Service Delivery", draft-ietf-l2sm-l2vpn-service-model, work in progress.

[L1CSM] G. Fioccola, et al, "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang, work in progress.

12. Contributors

Adrian Farrel
Old Dog Consulting

Email: adrian@olddog.co.uk

Italo Busi
Huawei Technologies

Email: Italo.Busi@huawei.com

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838

Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies

Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Apstra

EMail: jefftant@gmail.com

Giuseppe Fioccola
Huawei
Email: giuseppe.fioccola@huawei.com

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

TEAS Working Group A.Wang
Internet Draft China Telecom
Boris Khasanov
Huawei Technologies
Sudhir Cheruathur
Juniper Networks
Chun Zhu
ZTE Company

Intended status: Standard Track February 14, 2018
Expires: August 13, 2018

PCEP Extension for Native IP Network
draft-wang-pce-pcep-extension-native-ip-01.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the PCEP extension for CCDR application in Native IP network. The scenario and architecture of CCDR in native IP is described in [draft-ietf-teas-native-ip-scenarios] and [draft-ietf-teas-pce-native-ip]. This draft describes the key information that is transferred between PCE and PCC to accomplish the end2end traffic assurance in Native IP network under central control mode.

Table of Contents

1. Introduction

Traditionally, MPLS-TE traffic assurance requires the corresponding network devices support MPLS or the complex RSVP/LDP/Segment Routing etc. technologies to assure the end-to-end traffic performance. But in native IP network, there will be no such signaling protocol to synchronize the action among different network devices. It is necessary to use the central control mode that described in [draft-ietf-teas-pce-control-function] to correlate the forwarding behavior among different network devices. Draft [draft-ietf-teas-pce-native-ip] describes the architecture and solution philosophy for the end2end traffic assurance in Native IP network via Dual/Multi BGP solution. This draft describes the corresponding PCEP extension to transfer the key information about peer address list, peer prefix association and the explicit peer route on on-path router.

2. Conventions used in this document

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

3. New Objects Extension

Three new objects are defined in this draft; they are Peer Address List Object (PAL Object), Peer Prefix Association Object (PPA Object) and Explicit Peer Route object (EPR Object).

Peer Address List object is used to tell the network device which peer it should be peered with dynamically, Peer Prefix Association is used to tell which prefixes should be advertised via the corresponding peer and Explicit Peer Route object is used to point out which route should be to taken to arrive to the peer.

4. Object Formats.

Each extension object takes the similar format, that is to say, it began with the common object header defined in [RFC5440] as the following:

```
    0                   1                   2                   3

    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   | Object-Class  |   OT  |Res|P|I|   Object Length (bytes)       |

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   |                                                               |

   //                        (Object body)                        //

   |                                                               |

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

Different object-class, object type and the corresponding object body is defined separated in the following section.

4.1. Peer Address List object.

The Peer Address List object is used in a PCE Initiate message [draft-ietf-pce-pce-initiated-lsp] to specify the ip address of peer that the received network device should establish the BGP relationship with.

This Object should only be sent to the head and end router of the end2end path in case there is no RR involved. If the RR is used between the head end routers, then such information should be sent to head router/RR and end router/RR respectively.

Peer Address List object Object-Class is **

Peer Address List object Object-Type is **

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Peer Num    |   Peer-Id    |     AT        |      Resv.
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Local IP Address(4/16 Bytes)                  |
//                 Peer IP Address(4/16 Bytes)                 //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Peer Num (8 bits): Peer Address Number on the advertised router.

Peer-Id(8 bits): To distinguish the different peer pair, will be referenced in Peer Prefix Association, if the PCE use multi-BGP solution for different QoS assurance requirement.

AT(8 bits): Address Type. To indicate the address type of Peer.
Equal to 4, if the following IP address of peer is belong to IPv4;
Equal to 6 if the following IP address of peer is belong to IPv6.

Resv(8 bits): Reserved for future use.

Local IP Address(4/16 Bytes): IPv4 address of the local router, used to peer with other end router. When AT equal to 4, length is 32bit; when AT equal to 16, length is 128bit;

Peer IP Address(4/16 Bytes): IPv4 address of the peer router, used to peer with the local router. When AT equal to 4, length is 32bit; IPv6 address of the peer when AT equal to 16, length is 128bit;

4.2. Peer Prefix Association

THE Peer Prefix Association object is carried within in a PCE Initiate message [draft-ietf-pce-pce-initiated-lsp] to specify the IP prefixes that should be advertised by the corresponding Peer.

This Object should only be sent to the head and end router of the end2end path in case there is no RR involved. If the RR is used between the head end routers, then such information should be sent to head router/RR and end router/RR respectively.

Peer Prefix Association object Object-Class is **

Peer Prefix Association object Object-Type is **

```
    0                   1                   2                   3

    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   |   Peer-Id     |       AT      |      Resv.    | Prefixes Num.

   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   |             Peer Associated IP Prefix TLV                     |

   //            Peer Associated IP Prefix TLV                    //

   |             Peer Associated IP Prefix TLV                     |

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

Peer-Id(8 bits): To indicate which peer should be used to advertise the following IP Prefix TLV. This value is assigned in the Peer Address List object and is referred in this object.

AT(8 bits): Address Type. To indicate the address type of Peer. Equal to 4, if the following IP address of peer is belong to IPv4; Equal to 6 if the following IP address of peer is belong to IPv6.

Resv(8 bits): Reserved for future use.

Prefixes Num(8 bits): Number of prefixes that advertised by the corresponding Peer. It should be equal to num of the following IP prefix TLV.

Peer Associated IP Prefix TLV: Variable Length, use the TLV format to indicate the advertised IP Prefix.

4.3. EXPLICIT PEER ROUTE Object

THE EXPLICIT PEER ROUTE Object is carried in a PCE Initiate message [draft-ietf-pce-pce-initiated-lsp] to specify the explicit peer route to the corresponding peer address on each device that is on the end2end assurance path.

This Object should be sent to all the devices that locates on the end2end assurance path that calculated by PCE.

EXPLICIT PEER ROUTE Object Object-Class is **

EXPLICIT PEER ROUTE Object Object-Type is **

```
 0                   1                   2                   3

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|   Peer-Id     |       AT      |      Resv.                    |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

|           Next Hop Address to the Peer (IPv4/IPv6)            |

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

Peer-Id(8 bits): To indicate the peer that the following next hop address point to. This value is assigned in the Peer Address List object and is referred in this object.

AT(8 bits): Address Type. To indicate the address type of explicit peer route. Equal to 4, if the following next hop address to the peer is belong to IPv4; Equal to 6 if the following next hop address to the peer is belong to IPv6.

Resv(16 bits): Reserved for future use.

Next Hop Address to the Peer TLV: Variable Length, use the TLV format to indicate the next hop address to the corresponding peer that indicated by the Peer-Id.

5. Management Consideration.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Conclusions

TBD

9. References

9.1. Normative References

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

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

9.2. Informative References

[I-D.draft-ietf-pce-pce-initiated-lsp-07]
E.Crabbe, I.Minei, S.Sivabalan, R.Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", https://tools.ietf.org/html/draft-ietf-pce-pce-initiated-lsp-07 (work in progress), July, 2016

[I-D. draft-ietf-teas-native-ip-scenarios]
Wang, X.Huang et al. "CCDR Scenario, Simulation and Suggestion" https://datatracker.ietf.org/doc/draft-ietf-teas-native-ip-scenarios/ (work in progress), February, 2018

[I-D. draft-ietf-teas-pce-native-ip]
Aijun Wang, Quintin Zhao, Boris Khasanov, Huaimo Chen,Raghavendra Mallya, Shaofu Peng "PCE in Native IP Network", https://datatracker.ietf.org/doc/draft-ietf-teas-pce-native-ip/

(work in progress), February, 2018

[I-D.draft-ietf-teas-pce-control-function]
Farrel, Q.Zhao "An Architecture for use of PCE and PCEP in a Network with Central Control"
https://tools.ietf.org/html/draft-ietf-teas-pce-central-control-01

(work in progress),December, 2016

10. Acknowledgments

TBD

Authors' Addresses

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

Email: wangaj.bri@chinatelecom.cn

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

EMail: khasanov.boris@huawei.com

Sudhir Cheruathur
Juniper Networks
1133 Innovation Way
Sunnyvale, California 94089 USA

Email: scheruathur@juniper.net

Chun Zhu
ZTE Corporation
50 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China
Email:zhu.chun1@zte.com.cn

TEAS Working Group   A.Wang
Internet Draft   China Telecom
Xiaohong Huang
BUPT
Caixia Kou
BUPT
Lu Huang
China Mobile
Penghui Mi
Tencent Company

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

CCDR Scenario, Simulation and Suggestion
draft-wang-teas-ccdr-05.txt

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 July 24, 2018.

Copyright Notice

Abstract

This document describes the scenarios, simulation and suggestions for the "Centrally Control Dynamic Routing (CCDR)" architecture, which integrates the merit of traditional distributed protocols (IGP/BGP), and the power of centrally control technologies (PCE/SDN) to provide one feasible traffic engineering solution in various complex scenarios for the service provider.

Traditional MPLS-TE solution is mainly used in static network planning scenario and is difficult to meet the QoS assurance requirements in real-time traffic network. With the emerge of SDN concept and related technologies, it is possible to simplify the complexity of distributed control protocol, utilize the global view of network condition, give more efficient solution for traffic engineering in various complex scenarios.

Table of Contents

1. Introduction

Internet network is composed mainly tens of thousands of routers that run distributed protocol to exchange the reachability information between them. The path for the destination network is mainly calculated and controlled by the traditional IGP protocols. These distributed protocols are robust enough to support the current evolution of Internet but has some difficulties when the application requires the end-to-end QoS performance, or the service provider wants to maximize the links utilization within their network.

MPLS-TE technology is one perfect solution for the finely planned network but it will put heavy burden on the router when we use it to solve the dynamic QoS assurance requirements within real time traffic network.

SR(Segment Routing) is another prominent solution that integrates some merits of traditional distributed protocol and the advantages of centrally control mode, but it requires the underlying network, especially the provider edge router to do label push and pop action in-depth, and need some complex solutions for co-exist with the Non-SR network. Finally, it can only maneuver the end-to-end path for MPLS and IPv6 traffic via different mechanisms.

The advantage of MPLS is mainly for traffic isolation, such as the L2/L3 VPN service deployments, but most of the current application requirements are only for high performances end-to-end QoS assurance. Without the help of centrally control architecture, the service provider almost can’t make such SLA guarantees upon the real time traffic situation.

This draft gives some scenarios that the centrally control dynamic routing (CCDR) architecture can easily solve, without adding more extra burdening on the router. It also gives the PCE algorithm results under the similar topology, traffic pattern and network size to illustrate the applicability of CCDR architecture. Finally, it gives some suggestions for the implementation and deployment of CCDR.

2. CCDR Scenarios.

The following sections describe some scenarios that the CCDR architecture is suitable for deployment.

2.1. Qos Assurance for Hybrid Cloud-based Application.

With the emerge of cloud computing technologies, enterprises are putting more and more services on the public oriented service infrastructure, but keep still some core services within their network. The bandwidth requirements between the private cloud and the public cloud are occasionally and the background traffic between these two sites varied from time to time. Enterprise cloud applications just want to invoke the network capabilities to make the end-to-end QoS assurance on demand. Otherwise, the traffic should be controlled by the distributed protocol.

CCDR, which integrates the merits of distributed protocol and the power of centrally control, is suitable for this scenario. The possible solution architecture is illustrated below:

```
                           +------------------------+
                           | Cloud Based Application|
                           +------------------------+
                                       |
                                 +-----------+
                                 |    PCE    |
                                 +-----------+
                                       |
                                       |
                              //---------------\\
                         /////                  \\\\\
    Private Cloud Site ||       Distributed          |Public Cloud Site
                        |       Control Network      |
                         \\\\\                  /////
                              \\---------------//
```

Fig.1 Hybrid Cloud Communication Scenario

By default, the traffic path between the private cloud site and public cloud site will be determined by the distributed control network. When some applications require the end-to-end QoS assurance, it can send these requirements to PCE, let PCE compute one e2e path which is based on the underlying network topology and the real traffic information, to accommodate the application’s bandwidth requirements. The proposed solution can refer the draft [draft-wang-teas-pce-native-ip]. Section 4 describes the detail simulation process and the results.

2.2. Increase link utilization based on tidal phenomena.

Currently, the network topology within MAN is generally in star mode as illustrated in Fig.2, with the different devices connect different customer types. The traffic pattern of these customers demonstrates some tidal phenomena that the links between the CR/BRAS and CR/SR will experience congestion in different periods because the subscribers under BRAS often use the network at night and the dedicated line users under SR often use the network during the daytime. The uplink between BRAS/SR and CR must satisfy the maximum traffic pattern between them and this causes the links underutilization.

```
                           +--------+
                           |   CR   |
                           +----|---+
                                |
                    --------|--------|-------|
                    |       |        |       |
                 +--|-+   +-|-    +--|-+   +-|+
                 |BRAS|   |SR|    |BRAS|   |SR|
                 +----+   +--+    +----+   +--+
```

Fig.2 STAR-style network topology within MAN

If we can consider link the BRAS/SR with local loop, and control the MAN with the CCDR architecture, we can exploit the tidal phenomena between BRAS/CR and SR/CR links, increase the efficiency of them.

```
                                    +-------+
                                -----  PCE  |
                                |   +-------+
                           +----|---+
                           |   CR   |
                           +----|---+
                                |
                    --------|--------|-------|
                    |       |        |       |
                 +--|-+   +-|-    +--|-+   +-|+
                 |BRAS-----SR|    |BRAS-----SR|
                 +----+   +--+    +----+   +--+
```

Fig.3 Increase the link utilization via CCDR

2.3. Traffic engineering for IDC/MAN asymmetric link

The operator’s networks are often comprised by tens of different domains, interconnected with each other, form very complex topology that illustrated in Fig.4. Due to the traffic pattern to/from MAN and IDC, the links between them are often in asymmetric style. It is almost impossible to balance the utilization of these links via the distributed protocol, but this unbalance phenomenon can be overcome via the CCDR architecture.

```
                 +---+                +---+
                 |MAN|-----------------IDC|
                 +-|-|       |        +-|-+
                   |     ---------|     |
                   ------|BackBone|------
                   |     ----|----|     |
                   |         |          |
                 +-|--       |        ----+
                 |IDC|----------------|MAN|
                 +---|                |---+
```

Fig.4 TE within Complex Multi-Domain topology

2.4. Network temporal congestion elimination.

In more general situation, there are often temporal congestion periods within part of the service provider’s network. Such congestion phenomena will appear repeatedly and if the service provider has some methods to mitigate it, it will certainly increase the satisfaction degree of their customer. CCDR is also suitable for such scenario that the traditional distributed protocol will process most of the traffic forwarding and the controller will schedule some traffic out of the congestion links to lower the utilization of them. Section 4 describes the simulation process and results about such scenario.

3. CCDR Simulation.

The following sections describe the topology, traffic matrix, end-to-end path optimization and congestion elimination in CCDR simulation.

3.1. Topology Simulation.

The network topology mainly contains nodes and links information. Nodes used in simulation have two types: core nodes and edge nodes. The core nodes are fully linked to each other. The edge nodes are connected only with some of the core nodes. Fig.5 is a topology example of 4 core nodes and 5 edge nodes. In CCDR simulation, 100 core nodes and 400 edge nodes are generated.

```
                                  +----+
                                 /|Edge|\
                                | +----+ |
                                |        |
                                |        |
                  +----+    +----+     +----+
                  |Edge|----|Core|-----|Core|---------+
                  +----+    +----+     +----+         |
                          /  |    \   /   |           |
                    +----+   |     \ /    |           |
                    |Edge|   |      X     |           |
                    +----+   |     / \    |           |
                          \  |    /   \   |           |
                  +----+    +----+     +----+         |
                  |Edge|----|Core|-----|Core|         |
                  +----+    +----+     +----+         |
                              |          |            |
                              |          +------\   +----+
                              |                  ---|Edge|
                              +-----------------/   +----+
```

Fig.5 Topology of simulation

The number of links connecting one edge node to the set of core nodes is randomly between 2 to 30, and the total number of links is more than 20000. Each link has its congestion threshold.

3.2. Traffic Matrix Simulation.

The traffic matrix is generated based on the link capacity of topology. It can result in many kinds of situations, such as congestion, mild congestion and non-congestion.

In CCDR simulation, the traffic matrix is 500*500. About 20% links are overloaded when the Open Shortest Path First (OSPF) protocol is used in the network.

3.3. CCDR End-to-End Path Optimization

The CCDR end-to-end path optimization is to find the best end-to-end path which is the lowest in metric value and each link of the path is far below link’s threshold. Based on the current state of the network, PCE within CCDR architecture combines the shortest path algorithm with penalty theory of classical optimization and graph theory.

Given background traffic matrix which is unscheduled, when a set of new flows comes into the network, the end-to-end path optimization finds the optimal paths for them. The selected paths bring the least congestion degree to the network.

The link utilization increment degree(UID) when the new flows are added into the network is shown in Fig.6. The first graph in Fig.6 is the UID with OSPF and the second graph is the UID with CCDR end-to-end path optimization. The average UID of graph one is more than 30%. After path optimization, the average UID is less than 5%. The results show that the CCDR end-to-end path optimization has an eye-catching decreasing in UID relative to the path chosen based on OSPF.

```
         +-----------------------------------------------------------+
        |                *                               *    *    *|
       60|                *                             * * *  *    *|
         |*      *       **     * *         *   *   *  ** * *  * * **|
         |*   * ** *   * **   *** **  *   * **  * * *  ** * *  *** **|
         |* * * ** *  ** **   *** *** **  **** ** ***  **** ** *** **|
       40|* * * ***** ** ***  *** *** **  **** ** *** ***** ****** **|
   UID(%)|* * ******* ** ***  *** ******* **** ** *** ***** *********|
         |*** ******* ** **** *********** *********** ***************|
         |******************* *********** *********** ***************|
       20|******************* ***************************************|
         |******************* ***************************************|
         |***********************************************************|
         |***********************************************************|
        0+-----------------------------------------------------------+
        0    100   200   300   400   500   600   700   800   900  1000
        +-----------------------------------------------------------+
        |                                                           |
       60|                                                           |
         |                                                           |
         |                                                           |
         |                                                           |
       40|                                                           |
   UID(%)|                                                           |
         |                                                           |
         |                                                           |
       20|                                                           |
         |                                                          *|
         |                                     *                    *|
         |        *         *  *    *       *  **                 * *|
        0+-----------------------------------------------------------+
        0    100   200   300   400   500   600   700   800   900  1000
                              Flow Number
```

Fig.6 Simulation result with congestion elimination

3.4. Network temporal congestion elimination

Different degree of network congestion is simulated. The congestion degree (CD) is defined as the link utilization beyond its threshold.

The CCDR congestion elimination performance is shown in Fig.7. The first graph is the congestion degree before the process of congestion elimination. The average CD of all congested links is more than 10%. The second graph shown in Fig.7 is the congestion degree after congestion elimination process. It shows only 12 links among totally 20000 links exceed the threshold, and all the congestion degree is less than 3%. Thus, after schedule of the traffic in congestion paths, the degree of network congestion is greatly eliminated and the network utilization is in balance.

```
                       Before congestion elimination
          +-----------------------------------------------------------+
          |                *                            ** *   ** ** *|
        20|                *                     *      **** * ** ** *|
          |*      *       **     * **       **  **** * ***** *********|
          |*   *  * *   * **** ****** *  ** *** **********************|
        15|* * * ** *  ** **** ********* *****************************|
          |* * ******  ******* ********* *****************************|
    CD(%) |* ********* ******* ***************************************|
        10|* ********* ***********************************************|
          |*********** ***********************************************|
          |***********************************************************|
         5|***********************************************************|
          |***********************************************************|
          |***********************************************************|
         0+-----------------------------------------------------------+
             0            0.5            1            1.5            2

                       After congestion elimination
        +-----------------------------------------------------------+
        |                                                           |
       20|                                                           |
         |                                                           |
         |                                                           |
       15|                                                           |
         |                                                           |
   CD(%) |                                                           |
       10|                                                           |
         |                                                           |
         |                                                           |
       5 |                                                           |
         |                                                           |
         |        *        **  * *  *  **   *  **                 *  |
       0 +-----------------------------------------------------------+
          0            0.5            1            1.5            2
                           Link Number(*10000)
```

Fig.7 Simulation result with congestion elimination

4. CCDR Deployment Consideration.

With the above CCDR scenarios and simulation results, we can know it is necessary and feasible to find one general solution to cope with various complex situations for the most complex optimal path computation in centrally manner based on the underlay network topology and the real time traffic.

[draft-wang-teas-native-ip] gives the principle solution for above scenarios, such thoughts can be extended to cover requirements that are more concretes in future.

5. Security Considerations

TBD

6. IANA Considerations

TBD

7. Conclusions

TBD

8. References

8.1. Normative References

[RFC5440]Vasseur, JP., Ed., and JL. Le Roux, Ed., "Path

Computation Element (PCE) Communication Protocol

(PCEP)", RFC 5440, March 2009,

<http://www.rfc-editor.org/info/rfc5440>.

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

8.2. Informative References

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

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

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

March,2017

[I-D. draft-wang-teas-pce-native-ip]

A.Wang, Quintin Zhao, Boris Khasanov, Penghui Mi,Raghavendra Mallya, Shaofu Peng   "PCE in Native IP Network"

https://tools.ietf.org/html/draft-wang-teas-pce-native-ip-03 March 13, 2017

[I-D. draft-wang-pcep-extension for native IP]

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

9. Contributors:

Tingting Yuan
Beijing University of Posts and Telecommunications
yuantingting@bupt.edu.cn

Qiong Sun
sunqiong.bri@chinatelecom.cn

Xiaoyan Wei
China Telecom Shanghai Company
weixiaoyan@189.cn

Dingyuan Hu
Beijing University of Posts and Telecommunications
hdy@bupt.edu.cn

10. Acknowledgments

TBD

Authors’ Addresses

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

Email: wangaj.bri@chinatelecom.cn

Xiaohong Huang
Beijing University of Posts and Telecommunications
No.10 Xitucheng Road, Haidian District
Beijing,China

EMail: huangxh@bupt.edu.cn

Caixia Kou
Beijing University of Posts and Telecommunications
No.10 Xitucheng Road, Haidian District
Beijing,China
koucx@lsec.cc.ac.cn

Lu Huang
China Mobile
32 Xuanwumen West Ave, Xicheng District
Beijing 100053
China
Email: hlisname@yahoo.com

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

Email kevinmi@tencent.com

TEAS Working Group
Internet Draft

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

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

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

Status of this Memo

Copyright Notice

Abstract

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

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

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

Table of Contents

1. Introduction

Draft [I-D.draft-wang-teas-ccdr] describes the scenario and simulation results for the CCDR traffic engineering. In summary, the requirements for CCDR traffic engineering in Native IP network are the following:
1) No complex MPLS signaling procedure.
2) End to End traffic assurance, determined QoS behavior.
3) Identical deployment method for intra- and inter- domain.
4) No influence to existing router forward behavior.
5) Can utilize the power of centrally control(PCE) and flexibility/robustness of distributed control protocol.
6) Coping with the differentiation requirements for large amount traffic and prefixes.
7) Flexible deployment and automation control.

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

2. Dual-BGP framework for simple topology.

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

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

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

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

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

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

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

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

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

Fig.1 Design Philosophy for Dual-BGP Framework

3. Dual-BGP in large Scale Topology

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

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

```
                  +------------R3--------------+
                  |                            |
       SW1-------R1-------R5---------R6-------R7--------SW2
                  |        |          |        |
                  +-------R2---------R4--------+
```

Fig.2 Dual-BGP Framework for large scale network

4. Multi-BGP for Extended Traffic Differentiation

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

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

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

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

Table 1. Traffic Requirement Criteria

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

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

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

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

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

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

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

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

Fig.3 PCE based framework for Multi-BGP deployment

6. PCEP extension for key parameters delivery.

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

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

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

7. CCDR Deployment Consideration

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

8.1 Scalability

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

8.2 High Availability

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

8.3 Incremental deployment

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

8. Security Considerations

TBD

9. IANA Considerations

TBD

10. Conclusions

TBD

11. References

11.1. Normative References

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

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

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

11.2. Informative References

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

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

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

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

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

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

March,2017

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

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

12. Acknowledgments

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

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

Authors’ Addresses

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

Email: wangaj.bri@chinatelecom.cn

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

EMail: quintin.zhao@huawei.com

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

EMail: khasanov.boris@huawei.com

Huaimo Chen
Huawei Technologies
Boston, MA,
USA

EMail: huaimo.chen@huawei.com

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

Email kevinmi@tencent.com

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

Email: rmallya@juniper.net

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

Email: peng.shaofu@zte.com.cn

TEAS WG

Internet Draft
Intended status: Informational

Expires: December 31, 2017

Young Lee
Haomian Zheng
Huawei

Daniel Ceccarrelli
Ericsson

Bin Yeong Yoon
ETRI

Oscar Gonzalez de Dios
Telefonica

Jong Yoon Shin
SKT

Sergio Belotti
Nokia

June 30, 2017

Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks

draft-zhang-teas-actn-yang-05

Status of this Memo

Copyright Notice

Abstract

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

This document explains how the different types of YANG models defined in the Operations and Management Area and in the Routing Area are applicable to the ACTN framework. This document also shows how the ACTN architecture can be satisfied using classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

Table of Contents

1. Introduction

Abstraction and Control of TE Networks (ACTN) describes a method for operating a Traffic Engineered (TE) network (such as an MPLS-TE network or a layer 1 transport network) to provide connectivity and virtual network services for customers of the TE network. The services provided can be tuned to meet the requirements (such as traffic patterns, quality, and reliability) of the applications hosted by the customers. More details about ACTN can be found in Section 2.

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC6020] is the language of choice for documenting data models, and YANG models have been produced to allow configuration or modelling of a variety of network devices, protocol instances, and network services. YANG data models have been classified in [Netmod-Yang-Model-Classification] and [Service-YANG].

This document shows how the ACTN architecture can be satisfied using classes of data model that have already been defined, and discusses the applicability of specific data models that are under development. It also highlights where new data models may need to be developed.

2. Abstraction and Control of TE Networks (ACTN) Architecture

[ACTN-Requirements] describes the high-level ACTN requirements. [ACTN-Frame] describes the architecture model for ACTN including the entities (Customer Network Controller (CNC), Multi-domain Service Coordinator (MDSC), and Physical Network Controller (PNC)) and their interfaces.

Figure 1 depicts a high-level control and interface architecture for ACTN and is a reproduction of Figure 3 from [ACTN-Frame]. A number of key ACTN interfaces exist for deployment and operation of ACTN-based networks. These are highlighted in Figure 1 (ACTN Interfaces) below:

```
                +--------------+        +---------------+        +--------------
+
                |    CNC-A     |        |     CNC-B     |        |     CNC-C
|
                |(DC provider) |        |     (ISP)     |        |     (MVNO)
|
                +--------------+        +---------------+        +--------------
+
                     \                          |                           /
      Business        \                         |                          /
      Boundary  =======\========================|=========================/=====
==
      Between           \                       | CMI                    /
      Customer &         -----------            |          --------------
      Network Provider              \           |         /
                                   +-----------------------+
                                   |          MDSC         |
                                   +-----------------------+
                                    /           |         \
                        ------------            |MPI       ----------------
                       /                        |                          \
                  +-------+                 +-------+                   +-------
+
                  |  PNC  |                 |  PNC  |                   |  PNC
|
                  +-------+                 +-------+                   +-------
+
                     | GMPLS               /      |                      /   \
                     | trigger            /       |SBI                  /     \
                  --------           -----        |                    /       \
                 (        )         (     )       |                   /
\
                -         -        ( Phys. )      |                  /       ---
--
                (  GMPLS   )        ( Net )       |                 /       (
  )
               (  Physical  )        ----         |                /       ( Phy
s. )
                (  Network )                   -----        -----          ( Ne
t )
                 -        -                   (     )      (     )          ---
--
                  (       )                  (  Phys. )   (  Phys. )
                  --------                    ( Net )      ( Net )
                                               -----        -----
```

Figure 1 : ACTN Interfaces

The interfaces and functions are described below (without modifying the definitions) in [ACTN-Frame]:

- The CNC-MDSC Interface (CMI) is an interface between a Customer Network Controller and a Multi Domain Service Controller. The interface will communicate the service request or application demand. A request will include specific service properties, for example, services type, bandwidth and constraint information. These constraints SHOULD be measurable by MDSC and therefore visible to CNC via CMI. The CNC can also request the creation of the virtual network based on underlying physical resources to provide network services for the applications. The CNC can provide the end-point information/characteristics, traffic matrix specifying specific customer constraints. The MDSC may also report potential network topology availability if queried for current capability from the Customer Network Controller.

- The MDSC-PNC Interface (MPI) is an interface between a Multi Domain Service Coordinator and a Physical Network Controller. It allows the MDSC to communicate requests to create/delete connectivity or to modify bandwidth reservations in the physical network. In multi-domain environments, each PNC is responsible for a separate domain. The MDSC needs to establish multiple MPIs, one for each PNC and perform coordination between them to provide cross-domain connectivity.

- The South-Bound Interface (SBI) is the provisioning interface for creating forwarding state in the physical network, requested via the Physical Network Controller. The SBI is not in the scope of ACTN, however, it is included in this document so that it can be compared to models in [Service-Yang].

3. Service Models

[Service-YANG] introduces a reference architecture to explain the nature and usage of service YANG models in the context of service orchestration. Figure 2 below depicts this relationship and is a reproduction of Figure 2 from [Service-YANG]. Four models depicted in Figure 2 are defined as follows:

- Customer Service Model: A customer service model is used to describe a service as offer or delivered to a customer by a network operator.
- Service Delivery Model: A service delivery model is used by a network operator to define and configure how a service is provided by the network.
- Network Configuration Model: A network configuration model is used by a network orchestrator to provide network-level configuration model to a controller.

. Device Configuration Model: A device configuration model is used by a controller to configure physical network elements.

```
                                                 Customer
                            ------------------   Service  ----------
                           |                  |  Model   |          |
                           |     Service      |<-------->| Customer |
                           |   Orchestrator   |          |          |
                           |                  |           ----------
                            ------------------
                              .            .              -----------
                             .              .      ......|Application|
                            .                .     :     |  BSS/OSS  |
                           .                  .    :      -----------
                          .  Service Delivery  .   :
                          .       Model        .   :
                 ------------------    ------------------
                |                  |  |                  |
                |     Network      |  |     Network      |
                |   Orchestrator   |  |   Orchestrator   |
                |                  |  |                  |
                .------------------    ------------------.
               .         :                       :        .
              .          : Network Configuration :         .
              .          :        Model          :         .
      ------------     ------------     ------------    ------------
     |            |   |            |   |            |  |            |
     | Controller |   | Controller |   | Controller |  | Controller |
     |            |   |            |   |            |  |            |
      ------------     ------------     ------------    ------------
         :              .       .                 :            :
         :             .         .      Device    :            :
         :            .           . Configuration :            :
         :            .           .     Model     :            :
     ---------     ---------   ---------     ---------      ---------
    | Network |   | Network | | Network |   | Network |    | Network |
    | Element |   | Element | | Element |   | Element |    | Element |
     ---------     ---------   ---------     ---------      ---------
```

Figure 2: An SDN Architecture with a Service Orchestrator

4. Service Model Mapping to ACTN

YANG models coupled with the RESTCONF/NETCONF protocol [Netconf][Restconf] provides solutions for the ACTN framework. This section explains which types of YANG models apply to each of the ACTN interfaces.

Refer to Figure 5 of [ACTN-Frame] for details of the mapping between ACTN functions and service models. In summary, the following mappings are held between and Service Yang Models and the ACTN interfaces.

- o Customer Service Model <-> CMI
- o Network Configuration Model <-> MPI
- o Device Configuration Model <-> SBI

4.1. Customer Service Models in the ACTN Architecture (CMI)

Customer Service Models, which are used between a customer and a service orchestrator as in [Service-YANG], should be used between the CNC and MDSC (e.g., CMI) serving as providing a simple intent-like model/interface.

Among the key functions of Customer Service Models on the CMI is the service request. A request will include specific service properties, including: service type and its characteristics, bandwidth, constraint information, and end-point characteristics.

The following table provides a list of functions needed to build the CMI. They are mapped with Customer Service Models.

Function	Yang Model
Transport Service Request	[Transport-Service-Model]
VN Service Request & Instantiation	[ACTN-VN-YANG]
VN Path Computation Request	[ACTN-VN-YANG]*
VN Performance Monitoring Telemetry	[ACTN-PM-Telemetry]**
Topology Abstraction	[TE-topology]

*VN Path computation request in the CMI context means network path computation request based on customer service connectivity request constraints prior to the instantiation of a VN creation.

**ietf-actn-te-kpi-telemetry model describes performance telemetry for ACTN VN model. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters. Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters.

4.2. Service Delivery Models in ACTN Architecture

The Service Delivery Models where the service orchestration and the network orchestration could be implemented as separate components as seen in [Service-YANG]. This is also known as Network Service Models. On the other hand, from an ACTN architecture point of view, the service delivery model between the service orchestrator and the network orchestrator is an internal interface between sub-components of the MDSC in a single MDSC model.

In the MDSC hierarchical model where there are multiple MDSCs, the interface between the top MDSC and the bottom MDSC can be mapped to service delivery models.

4.3. Network Configuration Models in ACTN Architecture (MPI)

The Network Configuration Models is used between the network orchestrator and the controller in [Service-YANG]. In ACTN, this model is used primarily between a MDSC and a PNC. The Network Configuration Model can be also used for the foundation of more advanced models, like hierarchical MDSCs (see Section 4.5)

The Network Configuration Model captures the parameters which are network wide information.

The following table provides a list of functions needed to build the MPI. They are mapped with Network Configuration Yang Models. Note that various Yang models are work in progress.

Function	Yang Model
Configuration Scheduling	[Schedule]
Path computation	[PATH_COMPUTATION-API]*
Path Provisioning	[TE-Tunnel]**
Topology Abstraction	[TE-topology]
Tunnel PM Telemetry	[ACTN-PM-Telemetry]***
Service Provisioning	TBD****
OTN Topology Abstraction	[OTN-YANG]
WSON Topology Abstraction	[WSON-YANG]
Flexi-grid Topology Abstraction	[Flexi-YANG]
ODU Tunnel Model	[ODU-Tunnel]

WSON TE Tunnel Model	[WSON-Tunnel]
Flexi-grid Tunnel Model	[Flexigrid-Tunnel]

* Related draft is presenting use cases for path computation API, and Yang related model is foreseen to be added.

** Note that path provisioning function is provided by ietf-te module in [TE-Tunnel].

** ietf-actn-te-kpi-telemetry model describes performance telemetry for TE-tunnel model. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the TE-tunnel level. Various conditions can be set for auto-scaling based on the telemetry data.

**** This function needs to be investigated further. This can be a part of [TE-Tunnel] which is to be determined. Service provisioning is an optional function that builds on top the path provisioning one.

Path provisioning and Topology abstraction functions are mandatory in any case, while Path Computation may be mandatory or optional depending on the type of topology abstraction used. Details of this topic are discussed in [ACTN-Abstraction].

Telemetry may also be an optional function.

4.4. Device Models in ACTN Architecture (SBI)

For the device YANG models are used for per-device configuration purpose, they can be used between the PNC and the physical network/devices. Note that SBI is not in the scope of ACTN. This section is provided to give some examples of YANG-based Device Models. An example of Device Models is ietf-te-device yang module defined in [TE-tunnel].

5. Examples of Using Different Types of YANG Models

5.1. Simple Connectivity Examples

The data model in [Transport-Service-Model] provides an intent-like connectivity service model which can be used in connection-oriented networks.

It would be used as follows in the ACTN architecture:

- A CNC uses this service model to specify the two client nodes that are to be connected, and also indicates the amount of traffic (i.e., the bandwidth required) and payload type. What may be additionally specified is the SLA that describes the required quality and resilience of the service.

- The MDSC uses the information in the request to pick the right network (domain) and also to select the provider edge nodes corresponding to the customer edge nodes.

  If there are multiple domains, then the MDSC needs to coordinate across domains to set up network tunnels to deliver a service. Thus coordination includes, but is not limited to, picking the right domain sequence to deliver a service. Before it can perform such functions, it needs to get the topology information from each PNC, using topology YANG models such as [te-topology]. The topology reported from PNC to MDSC can either be abstract or non-abstract.

  Additionally, an MDSC can initiate the creation of a tunnel (or tunnel segment) in order to fulfill the service request from CNC based on path computation upon the overall topology information it synthesized from different PNCs. The based model that can cater this purpose is the te-tunnel model specified in [te-tunnel].

- Then, the PNC needs to decide the explicit route of such a tunnel or tunnel segment (in case of multiple domains), and create such a tunnel using protocols such as PCEP and RSVP-TE or using per-hop configuration.

5.2. VN service example

The service model defined in [ACTN-VN-YANG] describes a virtual network (VN) as a service which is a set of multiple connectivity services:

. A CNC will request VN to the MDSC by specifying a list of VN members. Each VN member specifies either a single connectivity service, or a source with multiple potential destination points in the case that the precise destination sites are to be determined by MDSC.

  o In the first case, the procedure is the same as the connectivity service, except that in this case, there is a list of connections requested.

  o In the second case, where the CNC requests the MDSC to select the right destination out of a list of candidates, the MDSC needs to choose the best candidate and reply with the chosen destination for a given VN member. After this is selected, the connectivity request setup procedure is the same as in the connectivity-as-a-service example.

After the VN is set up, a successful reply message is sent from MDSC to CNC, indicating the VN is ready. This message can also be achieved by using the model defined in [ACTN-VN-YANG].

5.3. Data Center-Interconnection Example

This section describes more concretely how existing YANG models described in Section 4 map to an ACTN data center interconnection use case. Figure 3 shows a use-case which shows service policy-driven Data Center selection and is a reproduction of Figure A.1 from [ACTN-Info].

```
                             +----------------+
                             |       CNC      |
                             |   (Global DC   |
                             |   Operation    |
                             |    Control)    |
                             +--------+-------+
                                      | |   VN Requirement/Policy:
                    CMI:              | |  - Endpoint/DC location info
                 Service model        | |  - Endpoint/DC dynamic
                                      | |    selection policy
                                      | |    (for VM migration, DR, LB)
                                      | v
                            +---------+---------+
                            |  Multi-domain     | Service policy-driven
                            |Service Coordinator| dynamic DC selection
              MPI:          +-----+---+---+-----+
    Network Configuration         |   |   |
    Model                         |   |   |
                 +----------------+   |   +---------------+
                 |                    |                   |
           +-----+-----+       +------+-----+      +------+-----+
           |  PNC for  |       |  PNC for   |      |  PNC for   |
           | Transport |       | Transport  |      | Transport  |
           | Network A |       | Network B  |      | network C  |
           +-----------+       +------------+      +------------+
   Device        |                    |                   |
   Model         |                    |                   |
                 |                    |                   |
+---+      ------               ------              ------       +---+
|DC1|--////      \\\\       ////      \\\\      ////      \\\\---+DC5|
+---+ |              |     |              |    |              |  +---+
      |     TN A     +-----+     TN B     +----+      TN C    |
      /              |     |              |    |              |
     / \\\\      ////     / \\\\      ////      \\\\      ////
   +---+   ------        /      ------    \         ------ \
   |DC2|                /                  \                \+---+
   +---+               /                    \                |DC6|
                     +---+                   \ +---+         +---+
                     |DC3|                    \|DC4|
                     +---+                     +---+

                                                DR: Disaster Recovery
                                                LB: Load Balancing
```

Figure 3: Service Policy-driven Data Center Selection

Figure 3 shows how VN policies from the CNC (Global Data Center Operation) are incorporated by the MDSC to support multi-destination applications. Multi-destination applications refer to applications in which the selection of the destination of a network path for a given source needs to be decided dynamically to support such applications.

Data Center selection problems arise for VM mobility, disaster recovery and load balancing cases. VN’s policy plays an important role for virtual network operation. Policy can be static or dynamic. Dynamic policy for data center selection may be placed as a result of utilization of data center resources supporting VMs. The MDSC would then incorporate this information to meet the objective of this application.

5.3.1. CMI (CNC-MDSC Interface)

[ACTN-VN-YANG] is used to express the definition of a VN, its VN creation request, the service objectives (metrics, QoS parameters, etc.), dynamic service policy when VM needs to be moved from one Data Center to another Data Center, etc. This service model is used between the CNC and the MDSC (CMI). The CNC in this use-case is an external entity that wants to create a VN and operates on the VN.

5.3.2. MPI (MDSC-PNC Interface)

The Network Configuration Model is used between the MDSC and the PNCs. Based on the Customer Service Model’s request, the MDSC will need to translate the service model into the network configuration model to instantiate a set of multi-domain connections between the prescribed sources and the destinations. The MDSC will also need to dynamically interact with the CNC for dynamic policy changes initiated by the CNC. Upon the determination of the multi-domain connections, the MDSC will need to use the network configuration model such as [TE-Tunnel] to interact with each PNC involved on the path. [TE-Topology] is used to for the purpose of underlying domain network abstraction from the PNC to the MDSC.

5.3.3. PDI (PNC-Device interface)

The Device Model can be used between the PNC and its underlying devices that are controlled by the PNC. The PNC will need to trigger signaling using any mechanisms it employees (e.g. [RSVP-TE-YANG]) to provision its domain path segment. There can be a plethora of choices how to control/manage its domain network. The PNC is responsible to abstract its domain network resources and update it to the MDSC. Note that this interface is not in the scope of ACTN. This section is provided just for an illustration purpose.

6. Security

This document is an informational draft. When the models mentioned in this draft are implemented, detailed security consideration will be given in such work.

How security fits into the whole architecture has the following components:

- the use of Restconf security between components

- the use of authentication and policy to govern which services can be requested by different parties.

- how security may be requested as an element of a service and mapped down to protocol security mechanisms as well as separation (slicing) of physical resources)

7. Acknowledgements

We thank Adrian Farrel for providing useful comments and suggestions for this draft.

8. References

8.1. Informative References

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

[Netmod-Yang-Model-Classification] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", draft-ietf-netmod-yang-model-classification, work in progress.

[Netconf] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,

and A. Bierman, Ed., "Network Configuration Protocol

(NETCONF)", RFC 6241.

[Restconf] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf, work in progress.

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

[ACTN-Frame] D. Cecarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework, work in progress.

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

[TE-Tunnel] T. Saad (Editor), "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.

[ACTN-VN-YANG] Y. Lee (Editor), "A Yang Data Model for ACTN VN Operation", draft-lee-teas-actn-vn-yang, work in progress.

[ACTN-Info] Y. Lee & S. Belotti, "Information Model for Abstraction and Control of TE Networks (ACTN)", draft-leebelotti-teas-actn-info, work in progress.

[Transport-Service-Model] X. Zhang (Editor), "A Service YANG Model for Connection-oriented Transport Networks", draft-zhang-teas-transport-service-model, work in progress.

[PATH-COMPUTATION-API] I.Busi/S.Belotti et al. "Path Computation API", draft-busibel-ccamp-path-computation-api-00.txt, work in progress

[RSVP-TE-YANG] T. Saad (Editor), "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp, work in progress.

[Schedule] X. Liu, et. al., "A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule, work in progress.

[ACTN-Abstraction] Y. Lee, D. Dhody, and D. Ceccarelli, "ACTN Abstraction Methods", draft-lee-tease-actn-abstraction, work in progress.

[OTN-YANG] X. Zhang, A. Sharma, and X. Liu, "A YANG Data Model for Optical Transport Network Topology", draft-zhang-ccamp-l1-topo-yang, work in progress.

[WSON-YANG] Y. Lee, et. al., "A Yang Data Model for WSON Optical Networks", draft-ietf-ccamp-wson-yang, work in progress.

[Flexi-YANG] J.E. Lopez de Vergara, et. al., "YANG data model for Flexi-Grid Optical Networks", draft-vergara-ccamp-flexigrid-yang, work in progress.[ODU-Tunnel]   Sharma, R. Rao, and X. Zhang, "OTN Service YANG Model", draft-sharma-ccamp-otn-service-model, work in progress.

[ACTN-PM-Telemetry] Y. Lee, D. Dhody, S. Karunanithi, R. Vilalta, D. King, and D. Ceccarelli, "YANG models for ACTN TE Performance Monitoring Telemetry and Network Autonomics", draft-lee-teas-actn-pm-telemetry-autonomics, work in progress.

[WSON-Tunnel] Y. Lee, D. Dhody, V. Lopez, D. King, B. Yoon, and R. Vilalta, "A Yang Data Model for WSON Tunnel", draft-lee-ccamp-wson-tunnel-model, work in progress.

[Flexigrid-Tunnel] J. Vergara, D. Perdices, V. Lopez, O. Gonzalez de Dios, D. King, Y. Lee, and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-vergara-ccamp-flexigrid-media-channel-yang, work in progress.

9. Contributors

Contributor’s Addresses

Dhruv Dhody
Huawei Technologies

Email: dhruv.ietf@gmail.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com

Authors' Addresses

Young Lee
Huawei Technologies
5340 Legacy Drive
Plano, TX 75023, USA
Phone: (469)277-5838

Email: leeyoung@huawei.com

Haomian Zheng
Huawei Technologies

Email: zhenghaomian@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Bin Yeong Yoon
ETRI

Email: byyun@etri.re.kr

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

Jong Yoon Shin
SKT

Email: jongyoon.shin@sk.com

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com