

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 15, 2020

I. Bryskin
Individual
V. Beeram
T. Saad
Juniper Networks
X. Liu
Volta Networks
Y. Lee
Sung Kyun Kwan University
A. Guo
Futurewei
January 12, 2020

Basic YANG Model for Steering Client Services To Server Tunnels
draft-bryskin-teas-service-tunnel-steering-model-04

Abstract

This document describes a YANG data model for managing pools of transport tunnels and steering client services on them.

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 15, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	4
1.2. Tree Diagrams	4
1.3. Prefixes in Data Node Names	4
2. Explicit vs. Implicit Service2tunnel Mapping. Steering Services to Transport Tunnel Pools	5
3. The purpose of the model	5
4. Model Design	6
5. Tree Structure	7
6. YANG Modules	8
7. IANA Considerations	16
8. Security Considerations	17
9. References	17
9.1. Normative References	17
9.2. Informative References	19
Authors' Addresses	19

1. Introduction

Client layer services/signals are normally mapped onto carrying them across the network transport tunnels via client/server layer adaptation relationships. Such relationships are usually modeled as multi-layer topologies, whereas tunnels set up in underlay (server) topologies support links in respective overlay (client) topologies. In this respect having a link in a client topology means that the client layer traffic could be forwarded between link termination points (LTPs) terminating the link on opposite sides by the supporting tunnel(s) provisioned in the server layer topology.

This said there are numerous use cases in which describing the client service to server tunnel bindings via the topology formalism is impractical. Below are some examples of such use cases:

- o Mapping client services onto tunnels within the same network layer, for example, mapping L3 VPNs or MPLS-SR services onto IP MPLS tunnels;
- o Mapping client services onto tunnels provisioned in the highest layer topology supported by the network. For example, mapping

L2VPNs or E(V)PL services onto IP MPLS tunnels provisioned in an IP network;

- o Mapping client services to tunnels provisioned in separate network layers at the network's access points. Consider, for example, an OTN/ODUk network that is used to carry client signals of, say, 20 different types (e.g. Ethernet, SDH, FKON, etc.) entering and exiting the network over client facing interfaces. Although it is possible to describe such a network as a 21-layer TE topology with the OTN/ODUk topology serving each of the 20 client layer topologies [I-D.ietf-teas-yang-te-topo], such a description would be verbose, cumbersome, difficult to expand to accommodate additional client signals and unnecessary, because the client layer topologies would have zero switching flexibility inside the network (i.e. contain only unrelated links connecting access points across respective layer networks), and all what is required to know from the point of view of a management application is what ODUk tunnels are established or required, which client signals the tunnels could carry and at which network border nodes and how the client signals could be bound (i.e. adopted) to the tunnels.

It is worth noting that such non-topological client-service-to-server-tunnel mapping almost always happens on network border nodes. However, there are also important use cases where such a mapping is required in the middle of the network. One such use case is controlling on IP/MPLS FRR PLRs which LSPs are mapped onto which backup tunnels.

It is important to bear in mind that service2tunnel mappings could be very complex: large number of instances of services of the same or different types (possibly governed by different models) could be mapped on the same set of tunnels, with the latter being set in different network layers and of either TE or non-TE nature, P2P or P2MP or MP2MP type. Furthermore, the mappings could be hierarchical: tunnels carrying services could be clients of other tunnels.

Despite of the differences of transport tunnels and of services they carry the srvice2tunnel mappings could be modeled in a simple uniform way. Access to a data store of such mappings could be beneficial to network management applications. It would be possible, for example, to discover which services depend on which tunnels, which services will be affected if a given tunnel goes out of service, how many more services could be placed onto a given TE tunnel without the latter violating its TE commitments (such as bandwidth and delay). It would be also possible to demand in a single request moving numerous (ranges of) service instances from one set of tunnels to another.

This document defines a YANG data model for facilitating said service2tunnel mappings.

The YANG model in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

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:

- o augment
- o data model
- o 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 [RFC8340].

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

Table 1: Prefixes and Corresponding YANG Modules

2. Explicit vs. Implicit Service2tunnel Mapping. Steering Services to Transport Tunnel Pools

There are use cases in which client services require hard separation of the transport carrying them from the transport carrying other services. However, environment in which the services may share the same transport tunnels is far more common. For this reason the model defined in this document suggests replacing (or at least augmenting) the explicit service2tunnel mapping configuration (in which the tunnels are referred to by their IDs/names) with an implicit mapping. Specifically, the model introduces the notion of tunnel pool. A tunnel pool could be referred to by its network unique color and requires a service2tunnel mapping configuration to specify the tunnel pool color(s) instead of tunnel IDs/names. The model governs tunnel pool data store independently from the services steered on the tunnels. It is assumed (although not required) that the tunnels - constituents/components of a tunnel pool - are of the same type, provisioned using a common template. Importantly they could be dynamically added to/removed from the pool without necessitating service2tunnel mapping re-configuration. Such a service to tunnel pool steering approach has the following advantages:

- o Scalability and efficiency: pool component bandwidth utilization could be monitored, tunnels could be added to/removed from the pool if/when detected that current component bandwidth utilization has crossed certain thresholds. This allows for a very efficient network resource utilization and obviates the network management application from a very difficult task of service to tunnel mapping planning;
- o Automation and elasticity: pool component attributes could be modified - bandwidth auto-adjusted, protection added, delay constrained, etc.. The tunnels could be completely or partially replaced with tunnels of different types (e.g. TE vs. non-TE, P2P vs. P2MP, etc.) or even provisioned in different network layers (OTN/ODUk tunnels replacing IP TE tunnels). Importantly, all such modifications do not require service2tunnel mapping re-configurations as long as the modified or new tunnels remain within the same tunnel pool(s);
- o Transparency: new service sites supported by additional PEs could be added without service2tunnel mapping re-configuration.

3. The purpose of the model

The model is targeted to facilitate for network management applications, such as service orchestrators, the control of pools of transport tunnels and steering onto them client services

independently of network technology/layer specifics of both the services and the tunnels. The model could be applied to/implemented on physical devices, such as IP routers, as well as on abstract topology nodes. Furthermore, the model could be supported by a network (domain) controller, such as ACTN PNC, to act as a proxy server on behalf of any network element/node (physical or abstract) under its control.

4. Model Design

The data store described/governed by the model is comprised of a single top level list - TunnelPools. A TunnelPool, list element, is a container describing a set of transport tunnels (presumably with similar characteristics) identified by a network unique ID (color). A given TunnelPool could be generic to the entire network or specific to a particular network slice or network abstract topology. Furthermore, a TunnelPool may have no tunnels (i.e. may have empty Tunnels list). Service steered onto such a TunnelPool will be carried by best effort forwarding technique and flexibility available in the slice/topology the TunnelPool is assigned to or generally in the network

The TunnelPool container has the following fields:

- o Color [uint32 list key];
- o Slice/Abstract topology ID (if zero, the TunnelPool is generic to the network).
- o Tunnels list;
- o Services list.

The Tunnels list describes the pool constituents - active transport tunnels. The list members - Tunnel containers - include the following information:

- o tunnel type [e.g. P2P-TE, P2MP-TE, SR-TE, SR P2P, LDP P2P, LDP MP2MP, GRE, PBB, etc]
- o tunnel type specific tunnel ID [provided that a data store of the tunnel type, e.g. TE tunnels, is supported, the tunnelID allows for the management application to look up the tunnel in question to obtain detailed information about the tunnel];
- o tunnel encapsulation [e.g. MPLS label stack, Ethernet STAGs, GRE header, PBB header, etc].

The Services list describes services currently steered on the tunnel pool. The list members - Service containers - have the following attributes:

- o service type [e.g. fixed/transparent, L3VPN, L2VPN, EVPN, ELINE, EPL, EVPL, L1VPN, ACTN VN, etc.];
- o service type specific service ID [provided that a data store of the service type, e.g. L2VPN, is supported, the service ID allows for the management application to look up the service in question to obtain detailed information about the service];
- o client ports (source/destination node LTPs over which the service enters/exits the node/network, relevant only for fixed/transparent services);
- o service encapsulation [e.g. MPLS label stack, Ethernet CTAGs, etc.] - for service multiplexing/de-multiplexing on/from a statistically shared tunnel].

5. Tree Structure

```

module: ietf-tunnel-steering
  +--rw tunnel-pools
    +--rw tunnel-pool* [color]
      +--rw color                uint32
      +--rw description?         string
      +--rw te-topology-identifier
        +--rw provider-id?      te-types:te-global-id
        +--rw client-id?        te-types:te-global-id
        +--rw topology-id?      te-types:te-topology-id
      +--rw service* [service-type id]
        +--rw service-type      identityref
        +--rw id                string
        +--rw encapsulation
          +--rw type?           identityref
          +--rw value?          binary
        +--rw access-point* [node-address link-termination-point]
          +--rw node-address      inet:ip-address
          +--rw link-termination-point string
          +--rw direction?       enumeration
      +--rw tunnel* [tunnel-type source destination tunnel-id]
        +--rw tunnel-type        identityref
        +--rw source              inet:ip-address
        +--rw destination         inet:ip-address
        +--rw tunnel-id          binary
        +--rw encapsulation
          +--rw type?            identityref
          +--rw value?           binary

```

6. YANG Modules

```

<CODE BEGINS> file "ietf-tunnel-steering@2020-01-05.yang"
module ietf-tunnel-steering {
  yang-version 1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tunnel-steering";

  prefix "tnl-steer";

  import ietf-inet-types {
    prefix inet;
  }

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

```


organization

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

contact

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

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

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

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

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

Editor: Young Lee
 <<mailto:leeyoung@huawei.com>> ";

description

"This data model is for steering client service to server
tunnels.

Copyright (c) 2020 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 2020-01-05 {  
  description "Initial revision";  
  reference "TBD";  
}
```

```
/*  
 * Typedefs  
 */
```

```
/*
 * Identities
 */
identity service-type {
  description "Base identity for client service type.";
}
identity service-type-l3vpn {
  base service-type;
  description
    "L3VPN service.";
}
identity service-type-l2vpn {
  base service-type;
  description
    "L2VPN service.";
}
identity service-type-evpn {
  base service-type;
  description
    "EVPN service.";
}
identity service-type-eline {
  base service-type;
  description
    "ELINE service.";
}
identity service-type-epl {
  base service-type;
  description
    "EPL service.";
}
identity service-type-evpl {
  base service-type;
  description
    "EVPL service.";
}
identity service-type-l1vpn {
  base service-type;
  description
    "L1VPN service.";
}
identity service-type-actn-vn {
  base service-type;
  description
    "ACTN VN service.";
}
identity service-type-transparent {
  base service-type;
```

```
    description
      "Transparent LAN service.";
  }

  identity tunnel-type {
    description "Base identity for tunnel type.";
  }
  identity tunnel-type-te-p2p {
    base tunnel-type;
    description
      "TE point-to-point tunnel type.";
  }
  identity tunnel-type-te-p2mp {
    base tunnel-type;
    description
      "TE point-to-multipoint tunnel type.";
    reference "RFC4875";
  }
  identity tunnel-type-te-sr {
    base tunnel-type;
    description
      "Segment Rouging TE tunnel type.";
  }
  identity tunnel-type-sr {
    base tunnel-type;
    description
      "Segment Rouging tunnel type.";
  }
  identity tunnel-type-ldp-p2p {
    base tunnel-type;
    description
      "LDP point-to-point tunnel type.";
  }
  identity tunnel-type-ldp-mp2mp {
    base tunnel-type;
    description
      "Multicast LDP multipoint-to-multipoint tunnel type.";
  }
  identity tunnel-type-gre {
    base tunnel-type;
    description
      "GRE tunnel type.";
  }
  identity tunnel-type-pbb {
    base tunnel-type;
    description
      "PBB tunnel type.";
  }
}
```

```

identity service-encapsulation-type {
    description "Base identity for tunnel encapsulation.";
}
identity service-encapsulation-type-mpls-label {
    base service-encapsulation-type;
    description
        "Encapsulated by MPLS label stack, as an inner lable to
        identify the customer service.";
}
identity service-encapsulation-type-ethernet-c-tag {
    base service-encapsulation-type;
    description
        "Encapsulated by Ethernet C-TAG, to identify the customer
        service.";
}

identity tunnel-encapsulation-type {
    description "Base identity for tunnel encapsulation.";
}
identity tunnel-encapsulation-type-mpls-label {
    base tunnel-encapsulation-type;
    description
        "Encapsulated by MPLS label stack, as an outer label to
        be pushed into the tunnel.";
}
identity tunnel-encapsulation-type-ethernet-s-tag {
    base tunnel-encapsulation-type;
    description
        "Encapsulated by Ethernet S-TAG.";
}
identity tunnel-encapsulation-type-pbb {
    base tunnel-encapsulation-type;
    description
        "Encapsulated by PBB header.";
}
identity tunnel-encapsulation-type-gre {
    base tunnel-encapsulation-type;
    description
        "Encapsulated by GRE header.";
}

/*
 * Groupings
 */

/*
 * Configuration data and operational state data nodes
 */

```

```

container tunnel-pools {
  description
    "A list of mappings that steer client services to transport
    tunnel pools. The tunnel pools are managed independently from
    the services steered on them.";

  list tunnel-pool {
    key "color";
    description
      "A set of transport tunnels (presumably with similar
      characteristics) identified by a network unique ID, named
      'color'.";
    leaf color {
      type uint32;
      description
        "Unique ID of a tunnel pool.";
    }
    leaf description {
      type string;
      description
        "Client provided description of the tunnel pool.";
    }
    uses te-types:te-topology-identifier;

    list service {
      key "service-type id";
      description
        "A list of client services that are steered on this tunnel
        pool.";
      leaf service-type {
        type identityref {
          base service-type;
        }
        description
          "Service type required by the client.";
      }
      leaf id {
        type string;
        description
          "Unique ID of a client service for the specified
          service type.";
      }
      container encapsulation {
        description
          "The encapsulation information used to identify the
          customer service for multiplexing over shared tunnels.";
        leaf type {
          type identityref {

```

```

        base service-encapsulation-type;
    }
    description
        "The encapsulation type used to identify the customer
        service for multiplexing over shared tunnels.";
    }
    leaf value {
        type binary;
        description
            "The encapsulation value pushed to the tunnel to
            identify this service.
            If not specified, the system decides what
            value to be used for multiplexing.";
    }
}
list access-point {
    key "node-address link-termination-point";
    description
        "A list of client ports (Link Termination Points) for the
        service to enter or exist.";
    leaf node-address {
        type inet:ip-address;
        description
            "Node over which the service enters or exists.";
    }
    leaf link-termination-point {
        type string;
        description
            "Client port (Link Termination Point) over which the
            service enters or exists.";
    }
    leaf direction {
        type enumeration {
            enum "in" {
                description "The service enters to the network.";
            }
            enum "out" {
                description "The service exists from the network.";
            }
            enum "in-out" {
                description
                    "The service enters to and exists from the
                    network.";
            }
        }
    }
    description
        "Whether the service enters to or exists from the
        network.";
}

```

```

    }
  }
}
list tunnel {
  key "tunnel-type source destination tunnel-id";
  description
    "A list of tunnels in the tunnel pool.";

  leaf tunnel-type {
    type identityref {
      base tunnel-type;
    }
    description
      "Tunnel type based on constructing technologies and
      multipoint types, including P2P-TE, P2MP-TE, SR-TE,
      SR P2P, LDP P2P, LDP MP2MP, GRE, PBB, etc";
  }
  leaf source {
    type inet:ip-address;
    description
      "For a p2p or p2mp tunnel, this is the source address;
      for a mp2mp tunnel, this is the root address.";
    reference "RFC3209, RFC4875, RFC6388, RFC7582.";
  }
  leaf destination {
    type inet:ip-address;
    description
      "For a p2p tunnel, this is the tunnel endpoint address
      extracted from SESSION object;
      for a p2mp tunnel, this identifies the destination
      group, or p2mp-id;
      for a mp2mp tunnel identified by root and opaque-value,
      this value is set to '0.0.0.0'.";
    reference "RFC3209, RFC4875, RFC6388, RFC7582.";
  }
  leaf tunnel-id {
    type binary;
    description
      "For a p2p or p2mp tunnel, this is the tunnel identifier
      used in the SESSION that remains constant over the life
      of the tunnel;
      for a mp2mp tunnel, this is the opaque-value in the
      FEC element.";
    reference "RFC3209, RFC4875, RFC6388, RFC7582.";
  }
  container encapsulation {
    description
      "The encapsulation information used by the tunnel.";
  }
}

```

```
leaf type {
    type identityref {
        base service-encapsulation-type;
    }
    description
        "The encapsulation type used by the tunnel.";
}
leaf value {
    type binary;
    description
        "The encapsulation value pushed to the tunnel data to
        identify the traffic in this tunnel.
        If not specified, the system decides what
        value to be used.";
}
}
}
}
}
<CODE ENDS>
```

7. 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-tunnel-steering
 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-tunnel-steering
namespace:     urn:ietf:params:xml:ns:yang:ietf-tunnel-steering
prefix:        tnl-steer
reference:      RFC XXXX
```


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 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:

/tunnel-pools/tunnel-pool

This subtree specifies a list of tunnel pools. Modifying the configurations cause interruption to related services and tunnels.

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:

/tunnel-pools/tunnel-pool

Unauthorized access to this subtree can disclose the information of related services and tunnels.

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>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.
- [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-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-22 (work in progress), June 2019.
- [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 and Interfaces", draft-ietf-teas-yang-te-22 (work in progress), November 2019.

9.2. Informative References

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

Authors' Addresses

Igor Bryskin
Individual

E-Mail: i_bryskin@yahoo.com

Vishnu Pavan Beeram
Juniper Networks

E-Mail: vbeeram@juniper.net

Tarek Saad
Juniper Networks

E-Mail: tsaad@juniper.net

Xufeng Liu
Volta Networks

E-Mail: xufeng.liu.ietf@gmail.com

Young Lee
Sung Kyun Kwan University

E-Mail: younglee.tx@gmail.com

Aihua Guo
Futurewei

E-Mail: aihuaguo@futurewei.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
Huawei Technologies
D. Ceccarelli
Ericsson
I. Bryskin
Individual
B. Yoon
ETRI
7 March 2022

A YANG Data Model for VN Operation
draft-ietf-teas-actn-vn-yang-14

Abstract

This document provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

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 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	4
1.1.1. Requirements Language	4
1.2. Tree diagram	4
1.3. Prefixes in Data Node Names	4
2. Use-case of VN YANG Model in the ACTN context	5
2.1. Type 1 VN	5
2.2. Type 2 VN	6
3. High-Level Control Flows with Examples	7
3.1. Type 1 VN Illustration	7
3.2. Type 2 VN Illustration	8
3.2.1. VN and AP Usage	11
4. VN Model Usage	12
4.1. Customer view of VN	12
4.2. Auto-creation of VN by MDSC	12
4.3. Innovative Services	12
4.3.1. VN Compute	12
4.3.2. Multi-sources and Multi-destinations	16
4.3.3. Others	17
4.3.4. Summary	18
5. VN YANG Model (Tree Structure)	18
6. VN YANG Model	22
7. JSON Example	33
7.1. VN JSON	34
7.2. TE-topology JSON	39
8. Security Considerations	50
9. IANA Considerations	51
10. Acknowledgments	52
11. References	52
11.1. Normative References	52
11.2. Informative References	54
Appendix A. Performance Constraints	55
Appendix B. Contributors Addresses	55
Authors' Addresses	56

1. Introduction

This document provides a YANG [RFC7950] data model generally applicable to any mode of Virtual Network (VN) operation.

The VN model defined in this document is applicable in generic sense as an independent model in and of itself. The VN model defined in this document can also work together with other customer service models such as L3SM [RFC8299], L2SM [RFC8466] and L1CSM [I-D.ietf-ccamp-llcsm-yang] to provide a complete life-cycle service 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 VN in terms of multiple VN Members comprising a VN, multi-source and/or multi-destination characteristics of the VN Member, the VN's reference to TE-topology's Abstract Node;

The actual VN instantiation and computation is performed with Connectivity Matrices sub-module of TE-Topology Model [RFC8795] which provides TE network topology abstraction and management operation. Once TE-topology Model is used in triggering VN instantiation over the networks, TE-tunnel [I-D.ietf-teas-yang-te] Model will inevitably interact with TE-Topology model for setting up actual tunnels and LSPs under the tunnels.

Abstraction and Control of Traffic Engineered Networks (ACTN) describes a set of management and control functions used to operate one or more TE networks to construct virtual networks that can be represented to customers and that are built from abstractions of the underlying TE networks [RFC8453]. ACTN is the primary example of the usage of the VN YANG model.

Sections 2 and 3 provide the discussion of how the VN YANG model is applicable to the ACTN context where Virtual Network Service (VNS) operation is 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 [RFC8309]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN instantiation, VN computation, and its life-cycle service management and operations.

The VN operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture (NMDA) [RFC8342]. The origin of the data is indicated as per the origin metadata annotation.

1.1. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

1.1.1. 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.

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
vn	ietf-vn	[RFCXXXX]
yang	ietf-yang-types	[RFC6991]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[RFC8776]
tet	ietf-te-topology	[RFC8795]

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. Use-case of VN YANG Model in the ACTN context

In this section, ACTN is being used to illustrate the general usage of the VN YANG model. The model presented in this section has the following ACTN context.

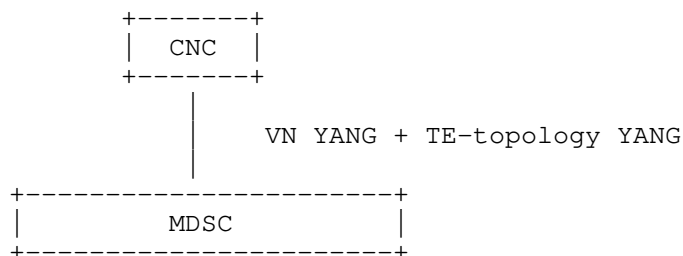


Figure 1: ACTN CMI

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 [RFC8453], a Virtual Network is a customer view of the TE network. To recapitulate VN types from [RFC8453], Type 1 VN is defined as follows:

The VN can be seen as a set of edge-to-edge abstract links (a Type 1 VN). Each abstract 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:

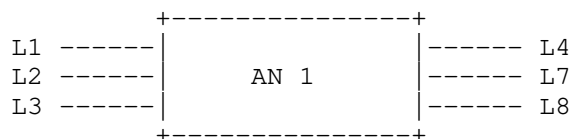


Figure 2: Abstract Node (One node topology)

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.

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 1 VN is a higher abstraction of a Type 2 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 under the Type 1 VN.

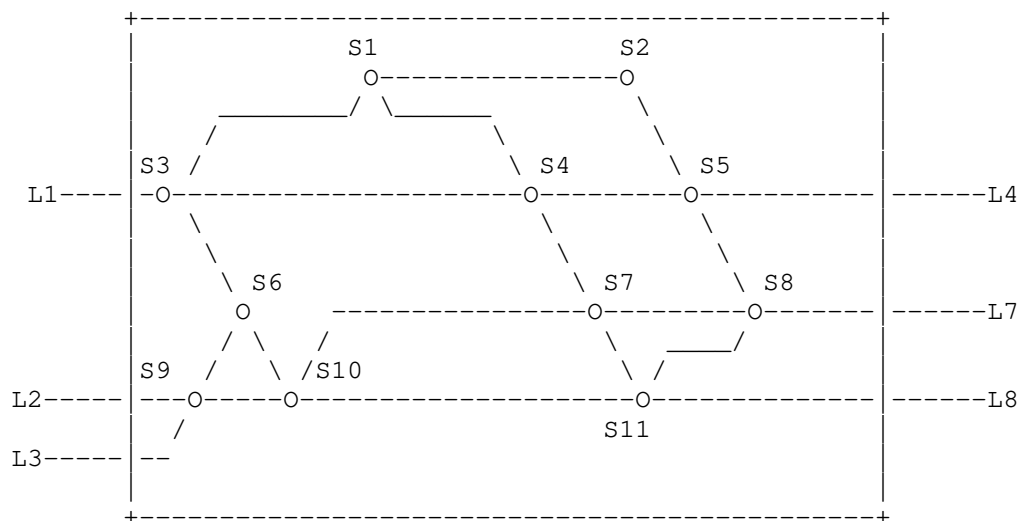


Figure 3: Type 2 topology

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

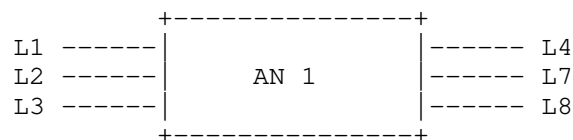
3.1. Type 1 VN Illustration

If we were to create a VN where we have four VN-members as follows:

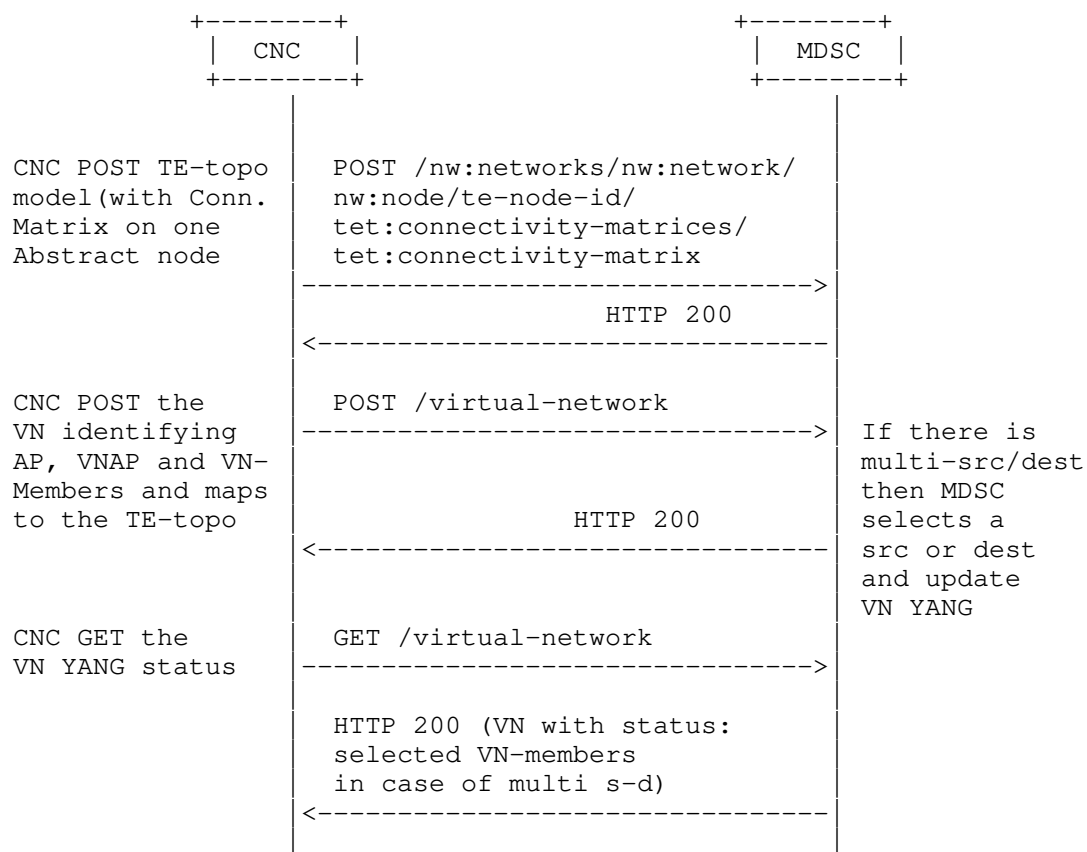
VN-Member 1	L1-L4
VN-Member 2	L1-L7
VN-Member 3	L2-L4
VN-Member 4	L3-L8

Where L1, L2, L3, L4, L7 and L8 correspond to Access Points.

This VN can be modeled as one abstract node representation as follows:



If this VN is Type 1, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.



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.

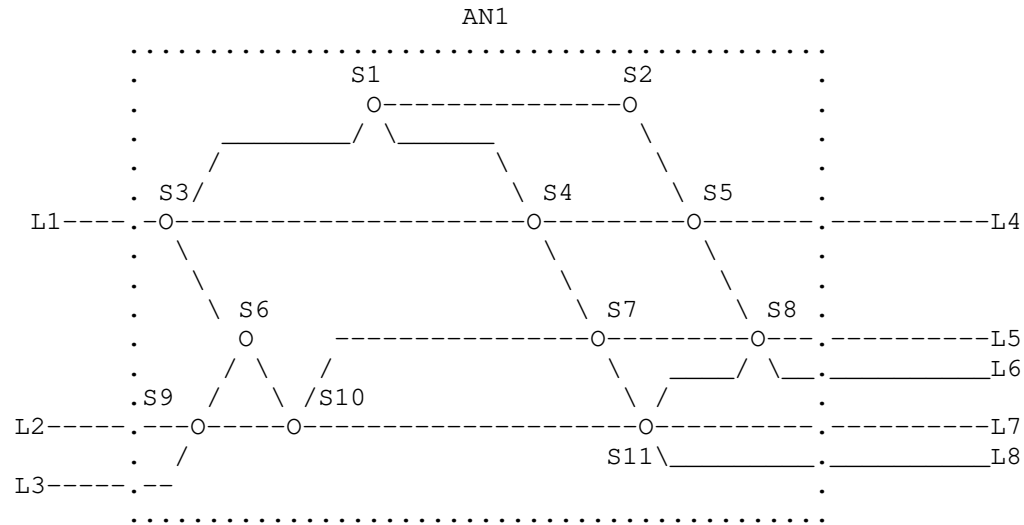
VN-Member 1 L1-L4 (via S3, S4, and S5)

VN-Member 2 L1-L7 (via S3, S4, S7 and S8)

VN-Member 3 L2-L7 (via S9, S10, and S11)

VN-Member 4 L3-L8 (via S9, S10 and S11)

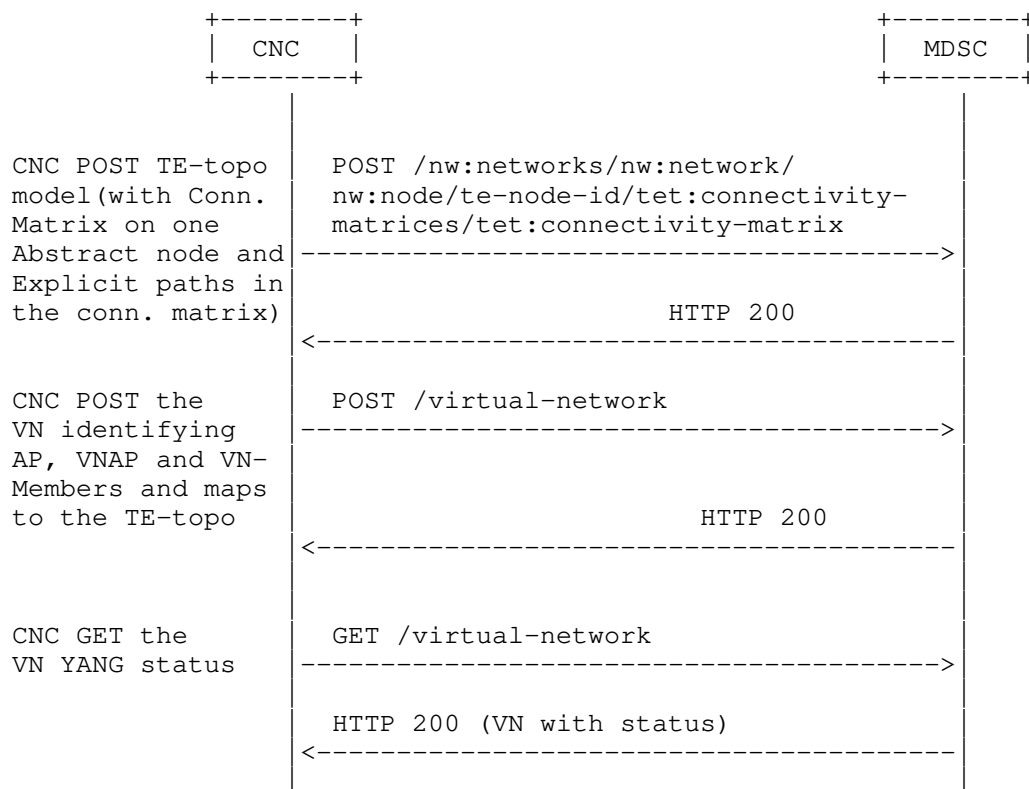
Where the following topology is the underlay for Abstraction Node 1 (AN1).



There are two options depending on whether CNC or MDSC creates the single abstract node topology.

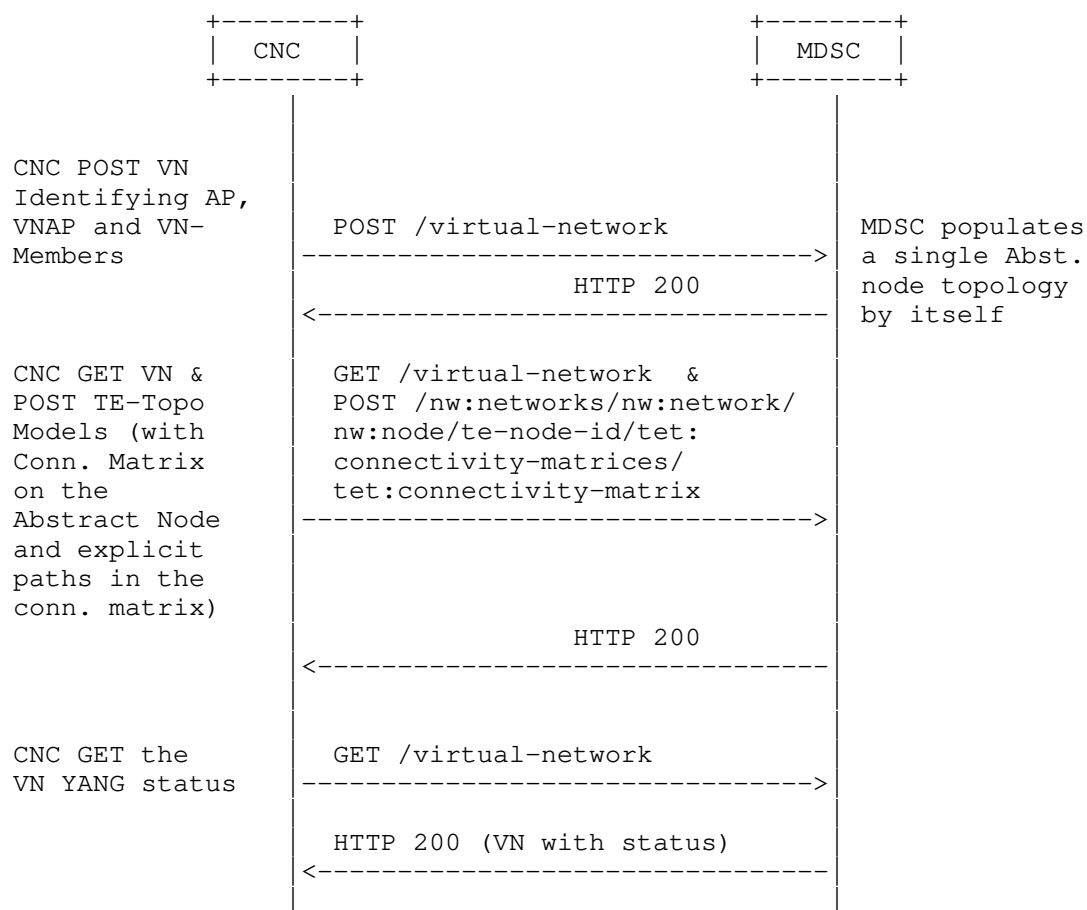
Case 1:

If CNC creates the single abstract node topology, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Model.



Case 2:

On the other hand, if MDSC create the single abstract node topology based VN YANG posted by the CNC, the following diagram shows the message flow between CNC and MDSC to instantiate this VN using VN and TE-Topology Models.



Section 7 provides JSON examples for both VN model and TE-topology Connectivity Matrix sub-model to illustrate how a VN can be created by the CNC making use of the VN module as well as the TE-topology Connectivity Matrix module.

3.2.1. VN and AP Usage

The customer access information may be known at the time of VN creation. A shared logical AP identifier is used between the customer and the operator to identify the access link between Customer Edge (CE) and Provider Edge (PE) . This is described in Section 6 of [RFC8453].

In some VN operations, the customer access may not be known at the initial VN creation. The VN operation allow a creation of VN with only PE identifier as well. The customer access information could be added later.

To achieve this the 'ap' container has a leaf for 'pe' node that allows AP to be created with PE information. The vn-member (and vn) could use APs that only have PE information initially.

4. VN Model Usage

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 [RFC8454]. 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 [RFC8309], which states that service models do not make any assumption of how a service is actually engineered and delivered for a customer.

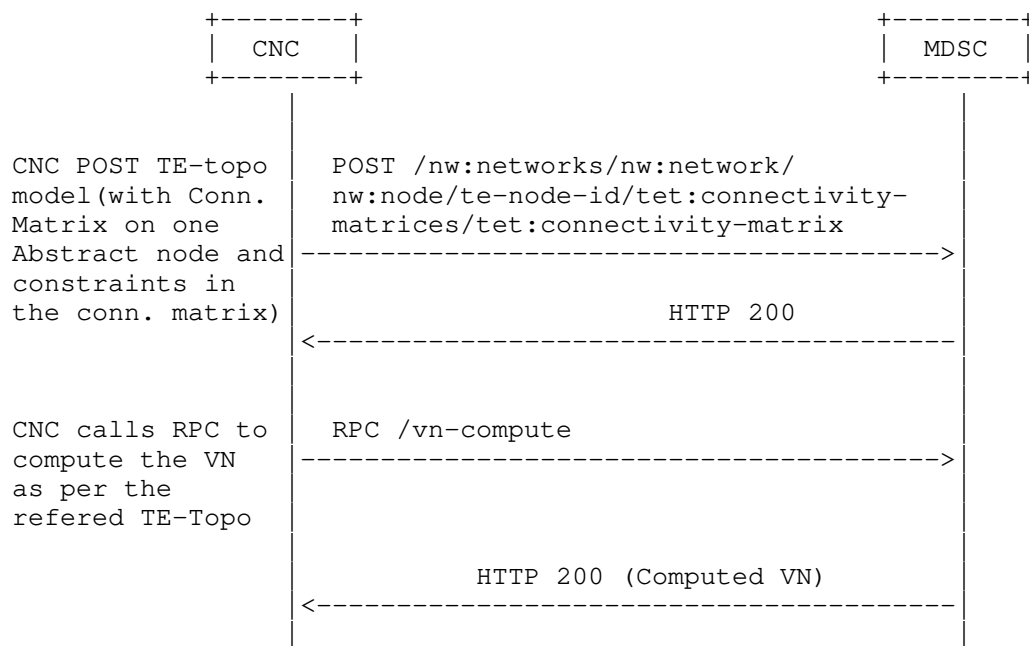
4.2. Auto-creation of VN by MDSC

The VN could be configured at the MDSC explicitly by the CNC using the VN YANG model. In some other cases, the VN is not explicitly configured, but created automatically by the MDSC based on the customer service model and local policy, even in these case the VN YANG model can be used by the CNC to learn details of the underlying VN created to meet the requirements of customer service model.

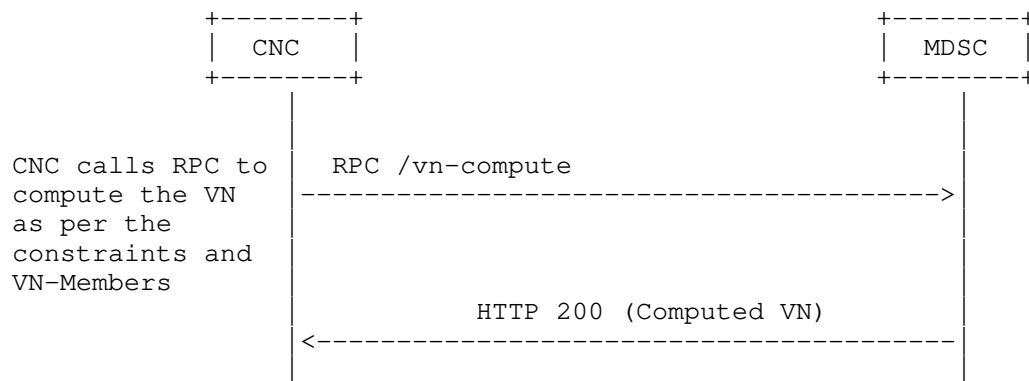
4.3. Innovative Services

4.3.1. VN Compute

VN Model 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.



The VN compute RPC allow you to optionally include the constraints and the optimization criteria at the VN as well as at the individual VN-member level. Thus, the RPC can be used independently to get the computed VN result without creating an abstract topology first.



In either case the output includes a reference to the single node abstract topology with each VN-member including a reference to the connectivity-matrix-id where the path properties could be found.

To achieve this the VN-compute RPC reuses the following common groupings:

- * `te-types:generic-path-constraints`: This is used optionally in the RPC input at the VN and/or VN-member level. The VN-member level overrides the VN-level data. This also overrides any constraints in the referred abstract node in the TE topology.
- * `te-types:generic-path-optimization`: This is used optionally in the RPC input at the VN and/or VN-member level. The VN-member level overrides the VN-level data. This also overrides any optimization in the referred abstract node in the TE topology.
- * `vn-member`: This identifies the VN member in both RPC input and output.
- * `vn-policy`: This is used optionally in the RPC input to apply any VN level policies.

When MDSC receives this RPC it computes the VN based on the input provided in the RPC call. This computation does not create a VN or reserve any resources in the system, it simply computes the resulting VN based on information at the MDSC or in coordination with the CNC. A single node abstract topology is used to convey the result of the each VN member as a reference to the `connectivity-matrix-id`. In case of error, the error information is included.

```

rpcs:
  +---x vn-compute
    +---w input
      +---w te-topology-identifier
        +---w provider-id?    te-global-id
        +---w client-id?     te-global-id
        +---w topology-id?   te-topology-id
      +---w abstract-node?
        -> /nw:networks/network/node/tet:te-node-id
      +---w path-constraints
        +---w te-bandwidth
          +---w (technology)?
            ...
        +---w link-protection?      identityref
        +---w setup-priority?       uint8
        +---w hold-priority?        uint8
        +---w signaling-type?       identityref
        +---w path-metric-bounds
          +---w path-metric-bound* [metric-type]
            ...
        +---w path-affinities-values
          +---w path-affinities-value* [usage]
            ...
        +---w path-affinity-names

```

```

    +---w path-affinity-name* [usage]
    |   ...
+---w path-srlgs-lists
    |   +---w path-srlgs-list* [usage]
    |   |   ...
+---w path-srlgs-names
    |   +---w path-srlgs-name* [usage]
    |   |   ...
+---w disjointness?                te-path-disjointness
+---w cos?                        te-types:te-ds-class
+---w optimizations
    |   +---w (algorithm)?
    |   |   +---:(metric) {path-optimization-metric}?
    |   |   |   ...
    |   |   +---:(objective-function)
    |   |   |   {path-optimization-objective-function}?
    |   |   |   ...
+---w vn-member-list* [vnm-id]
    |   +---w vnm-id                vnm-id
    |   +---w src
    |   |   +---w src?                -> /access-point/ap/ap-id
    |   |   +---w src-vn-ap-id?
    |   |   |   -> /access-point/ap/vn-ap/vn-ap-id
    |   |   +---w multi-src?          boolean {multi-src-dest}?
    |   +---w dest
    |   |   +---w dest?                -> /access-point/ap/ap-id
    |   |   +---w dest-vn-ap-id?
    |   |   |   -> /access-point/ap/vn-ap/vn-ap-id
    |   |   +---w multi-dest?          boolean {multi-src-dest}?
    |   +---w connectivity-matrix-id? leafref
    |   +---w underlay
    |   +---w path-constraints
    |   |   +---w te-bandwidth
    |   |   |   ...
    |   |   +---w link-protection?      identityref
    |   |   +---w setup-priority?        uint8
    |   |   +---w hold-priority?         uint8
    |   |   +---w signaling-type?        identityref
    |   |   +---w path-metric-bounds
    |   |   |   ...
    |   |   +---w path-affinities-values
    |   |   |   ...
    |   |   +---w path-affinity-names
    |   |   |   ...
    |   |   +---w path-srlgs-lists
    |   |   |   ...
    |   |   +---w path-srlgs-names
    |   |   |   ...
    |   |   ...

```

```

|   |   | +---w disjointness?          te-path-disjointness
|   |   | +---w cos?                  te-types:te-ds-class
|   |   | +---w optimizations
|   |   |   +---w (algorithm)?
|   |   |   ...
|   |   +---w vn-level-diversity?
|   |       te-types:te-path-disjointness
+--ro output
+--ro te-topology-identifier
|   +--ro provider-id?    te-global-id
|   +--ro client-id?      te-global-id
|   +--ro topology-id?    te-topology-id
+--ro abstract-node?
|   -> /nw:networks/network/node/tet:te-node-id
+--ro vn-member-list* [vnm-id]
|   +--ro vnm-id          vnm-id
|   +--ro src
|   |   +--ro src?          -> /access-point/ap/ap-id
|   |   +--ro src-vn-ap-id?
|   |   |   -> /access-point/ap/vn-ap/vn-ap-id
|   |   +--ro multi-src?    boolean {multi-src-dest}?
|   +--ro dest
|   |   +--ro dest?          -> /access-point/ap/ap-id
|   |   +--ro dest-vn-ap-id?
|   |   |   -> /access-point/ap/vn-ap/vn-ap-id
|   |   +--ro multi-dest?    boolean {multi-src-dest}?
+--ro connectivity-matrix-id? leafref
+--ro underlay
+--ro if-selected?          boolean
|   {multi-src-dest}?
+--ro compute-status?       vn-compute-status
+--ro error-info
|   +--ro error-description? string
|   +--ro error-timestamp?   yang:date-and-time
|   +--ro error-reason?      identityref

```

4.3.2. 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.

```

module: ietf-vn
  +--rw virtual-network
    +--rw vn* [vn-id]
      +--rw vn-id vn-id
      +--rw te-topology-identifier
        | +--rw provider-id? te-global-id
        | +--rw client-id? te-global-id
        | +--rw topology-id? te-topology-id
      +--rw abstract-node?
        | -> /nw:networks/network/node/tet:te-node-id
      +--rw vn-member* [vnm-id]
        +--rw vnm-id vnm-id
        +--rw src
          | +--rw src? -> /access-point/ap/ap-id
          | +--rw src-vn-ap-id?
          | | -> /access-point/ap/vn-ap/vn-ap-id
          | +--rw multi-src? boolean {multi-src-dest}?
        +--rw dest
          | +--rw dest? -> /access-point/ap/ap-id
          | +--rw dest-vn-ap-id?
          | | -> /access-point/ap/vn-ap/vn-ap-id
          | +--rw multi-dest? boolean {multi-src-dest}?
        +--rw connectivity-matrix-id? leafref
        +--rw underlay
          +--ro oper-status? te-types:te-oper-status
        +--ro if-selected? boolean {multi-src-dest}?
        +--rw admin-status? te-types:te-admin-status
        +--ro oper-status? te-types:te-oper-status
        +--rw vn-level-diversity? te-types:te-path-disjointness

```

4.3.3. 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 [I-D.ietf-teas-te-service-mapping-yang].

The VN YANG model can be extended to support telemetry, performance monitoring and network autonomies as described in [I-D.ietf-teas-actn-pm-telemetry-autonomics].

Note that the YANG model is tightly coupled with the TE Topology model [RFC8795]. Any underlay technology not supported by [RFC8795] is also not supported by this model. The model does include an empty container called "underlay" that can be augmented. For example the SR-policy information can be augmented for the SR underlay by a future model.

Apart from the `te-types:generic-path-constraints` and `te-types:generic-path-optimization`, an additional leaf `cos` for class of service [RFC4124] is added to represent the Class-Type of traffic to be used as one of the path constraints.

4.3.4. Summary

This section summarizes the innovative service features of the 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. VN YANG Model (Tree Structure)

```

module: ietf-vn
+--rw access-point
|   +--rw ap* [ap-id]
|       +--rw ap-id          ap-id
|       +--rw pe?
|       |   -> /nw:networks/network/node/tet:te-node-id
|       +--rw max-bandwidth? te-types:te-bandwidth
|       +--rw avl-bandwidth? te-types:te-bandwidth
|       +--rw vn-ap* [vn-ap-id]
|           +--rw vn-ap-id      ap-id
|           +--rw vn?          -> /virtual-network/vn/vn-id

```

```

|         +---rw abstract-node?
|         |         -> /nw:networks/network/node/tet:te-node-id
|         +---rw ltp?          leafref
|         +---ro max-bandwidth? te-types:te-bandwidth
+---rw virtual-network
  +---rw vn* [vn-id]
    +---rw vn-id          vn-id
    +---rw te-topology-identifier
    |   +---rw provider-id?  te-global-id
    |   +---rw client-id?    te-global-id
    |   +---rw topology-id?  te-topology-id
    +---rw abstract-node?
    |   -> /nw:networks/network/node/tet:te-node-id
  +---rw vn-member* [vnm-id]
    +---rw vnm-id          vnm-id
    +---rw src
    |   +---rw src?          -> /access-point/ap/ap-id
    |   +---rw src-vn-ap-id?
    |   |   -> /access-point/ap/vn-ap/vn-ap-id
    |   +---rw multi-src?    boolean {multi-src-dest}?
    +---rw dest
    |   +---rw dest?         -> /access-point/ap/ap-id
    |   +---rw dest-vn-ap-id?
    |   |   -> /access-point/ap/vn-ap/vn-ap-id
    |   +---rw multi-dest?    boolean {multi-src-dest}?
    +---rw connectivity-matrix-id? leafref
    +---rw underlay
    |   +---ro oper-status?    te-types:te-oper-status
  +---ro if-selected?         boolean {multi-src-dest}?
  +---rw admin-status?        te-types:te-admin-status
  +---ro oper-status?         te-types:te-oper-status
  +---rw vn-level-diversity?   te-types:te-path-disjointness

rpcs:
  +---x vn-compute
    +---w input
      +---w te-topology-identifier
      |   +---w provider-id?  te-global-id
      |   +---w client-id?    te-global-id
      |   +---w topology-id?  te-topology-id
      +---w abstract-node?
      |   -> /nw:networks/network/node/tet:te-node-id
      +---w path-constraints
      |   +---w te-bandwidth
      |   |   +---w (technology)?
      |   |   ...
      |   +---w link-protection? identityref
      +---w setup-priority?      uint8

```

```

+---w hold-priority?                uint8
+---w signaling-type?              identityref
+---w path-metric-bounds
|   +---w path-metric-bound* [metric-type]
|   ...
+---w path-affinities-values
|   +---w path-affinities-value* [usage]
|   ...
+---w path-affinity-names
|   +---w path-affinity-name* [usage]
|   ...
+---w path-srlgs-lists
|   +---w path-srlgs-list* [usage]
|   ...
+---w path-srlgs-names
|   +---w path-srlgs-name* [usage]
|   ...
+---w disjointness?                te-path-disjointness
+---w cos?                        te-types:te-ds-class
+---w optimizations
|   +---w (algorithm)?
|   |   +--:(metric) {path-optimization-metric}?
|   |   |   ...
|   |   +--:(objective-function)
|   |   |   {path-optimization-objective-function}?
|   |   ...
+---w vn-member-list* [vnm-id]
|   +---w vnm-id                    vnm-id
|   +---w src
|   |   +---w src?                  -> /access-point/ap/ap-id
|   |   +---w src-vn-ap-id?
|   |   |   -> /access-point/ap/vn-ap/vn-ap-id
|   |   +---w multi-src?            boolean {multi-src-dest}?
|   +---w dest
|   |   +---w dest?                  -> /access-point/ap/ap-id
|   |   +---w dest-vn-ap-id?
|   |   |   -> /access-point/ap/vn-ap/vn-ap-id
|   |   +---w multi-dest?            boolean {multi-src-dest}?
+---w connectivity-matrix-id?      leafref
+---w underlay
+---w path-constraints
|   +---w te-bandwidth
|   |   ...
|   +---w link-protection?          identityref
|   +---w setup-priority?           uint8
|   +---w hold-priority?            uint8
|   +---w signaling-type?           identityref
+---w path-metric-bounds

```

```

|
|
|
|      ...
|      +---w path-affinities-values
|      |      ...
|      +---w path-affinity-names
|      |      ...
|      +---w path-srlgs-lists
|      |      ...
|      +---w path-srlgs-names
|      |      ...
|      +---w disjointness?                te-path-disjointness
+---w cos?                                te-types:te-ds-class
+---w optimizations
|      +---w (algorithm)?
|      |      ...
+---w vn-level-diversity?
|      te-types:te-path-disjointness
+--ro output
+--ro te-topology-identifier
|   +--ro provider-id?    te-global-id
|   +--ro client-id?     te-global-id
|   +--ro topology-id?   te-topology-id
+--ro abstract-node?
|   -> /nw:networks/network/node/tet:te-node-id
+--ro vn-member-list* [vnm-id]
|   +--ro vnm-id          vnm-id
+--ro src
|   +--ro src?            -> /access-point/ap/ap-id
|   +--ro src-vn-ap-id?
|   |   -> /access-point/ap/vn-ap/vn-ap-id
|   +--ro multi-src?     boolean {multi-src-dest}?
+--ro dest
|   +--ro dest?          -> /access-point/ap/ap-id
|   +--ro dest-vn-ap-id?
|   |   -> /access-point/ap/vn-ap/vn-ap-id
|   +--ro multi-dest?    boolean {multi-src-dest}?
+--ro connectivity-matrix-id? leafref
+--ro underlay
+--ro if-selected?        boolean
|   {multi-src-dest}?
+--ro compute-status?     vn-compute-status
+--ro error-info
|   +--ro error-description? string
|   +--ro error-timestamp?  yang:date-and-time
|   +--ro error-reason?     identityref

```


6. VN YANG Model

The YANG model is as follows:

```
<CODE BEGINS> file "ietf-vn@2022-03-07.yang"
module ietf-vn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn";
  prefix vn;

  /* Import network */

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  /* Import network topology */

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

  /* Import TE Common types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }

  /* Import TE Topology */

  import ietf-te-topology {
    prefix tet;
    reference
      "RFC 8795: YANG Data Model for Traffic Engineering (TE)
        Topologies";
  }
}
```

```
organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
  WG List:  <mailto:teas@ietf.org>
  Editor: Young Lee <younglee.tx@gmail.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>";
```

```
description
  "This module contains a YANG module for the Virtual Network
  (VN). It describes a VN operation module that takes place
  in the context of the Customer Network Controller (CNC)-
  Multi-Domain Service Coordinator (MSDC) interface (CMI) of
  the Abstraction and Control of Traffic Engineered Networks
  (ACTN) architecture where the CNC is the actor of a VN
  Instantiation/modification/deletion as per RFC 8453.
```

Copyright (c) 2022 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; see the RFC itself for full legal notices.

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.";

```
revision 2022-03-07 {
  description
    "initial version.";
  reference
    "RFC XXXX: A YANG Data Model for VN Operation";
}
```

```
/* Features */
```

```
feature multi-src-dest {
  description
    "Support for selection of one src or destination
```

```
        among multiple.";
    reference
        "RFC 8453: Framework for Abstraction and Control of TE
        Networks (ACTN)";
}

/* Typedef */

typedef vn-id {
    type string;
    description
        "Defines a type of Virtual Network (VN) identifier.";
}

typedef ap-id {
    type string;
    description
        "Defines a type of Access Point (AP) identifier.";
}

typedef vnm-id {
    type string;
    description
        "Defines a type of VN member identifier.";
}

typedef vn-compute-status {
    type te-types:te-common-status;
    description
        "Defines a type representing the VN compute status. Note
        that all status apart from up and down are considered as
        unknown.";
}

/* identities */

identity vn-computation-error-reason {
    description
        "Base identity for VN computation error reasons.";
}

identity vn-computation-error-not-ready {
    base vn-computation-error-reason;
    description
        "VN computation has failed because the MDSC is not
        ready";
}
```

```
identity vn-computation-error-no-cnc {
  base vn-computation-error-reason;
  description
    "VN computation has failed because one or more dependent
    CNC are unavailable.";
}

identity vn-computation-error-no-resource {
  base vn-computation-error-reason;
  description
    "VN computation has failed because there is no
    available resource in one or more domains.";
}

identity vn-computation-error-path-not-found {
  base vn-computation-error-reason;
  description
    "VN computation failed as no path found.";
}

identity vn-computation-ap-unknown {
  base vn-computation-error-reason;
  description
    "VN computation failed as source or destination AP not
    known.";
}

/* Groupings */

grouping vn-ap {
  description
    "VNAP related information";
  leaf vn-ap-id {
    type ap-id;
    description
      "A unique identifier for the referred VNAP";
  }
  leaf vn {
    type leafref {
      path "/virtual-network/vn/vn-id";
    }
    description
      "A 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 that
       represent the VN";
  }
  leaf ltp {
    type leafref {
      path "/nw:networks/nw:network/nw:node/"
        + "nt:termination-point/tet:te-tp-id";
    }
    description
      "A reference to Link Termination Point (LTP) in the
       TE-topology";
    reference
      "RFC 8795: YANG Data Model for Traffic Engineering (TE)
       Topologies";
  }
  leaf max-bandwidth {
    type te-types:te-bandwidth;
    config false;
    description
      "The max bandwidth of the VNAP";
  }
  reference
    "RFC 8453: Framework for Abstraction and Control of TE
     Networks (ACTN), Section 6";
} //vn-ap

grouping access-point {
  description
    "AP related information";
  leaf ap-id {
    type ap-id;
    description
      "A unique identifier for the referred access point";
  }
  leaf pe {
    type leafref {
      path "/nw:networks/nw:network/nw:node/tet:te-node-id";
    }
    description
      "A reference to the PE node in the native TE Topology";
  }
  leaf max-bandwidth {
    type te-types:te-bandwidth;
    description
      "The max bandwidth of the AP";
  }
  leaf avl-bandwidth {
```

```
    type te-types:te-bandwidth;
    description
      "The 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";
  }
  reference
    "RFC 8453: Framework for Abstraction and Control of TE
    Networks (ACTN), Section 6";
} //access-point

grouping vn-member {
  description
    "The vn-member is described by this grouping";
  leaf vnm-id {
    type vnm-id;
    description
      "A vn-member identifier";
  }
  container src {
    description
      "The source of VN Member";
    leaf src {
      type leafref {
        path "/access-point/ap/ap-id";
      }
      description
        "A reference to source AP";
    }
    leaf src-vn-ap-id {
      type leafref {
        path "/access-point/ap/vn-ap/vn-ap-id";
      }
      description
        "A reference to source VNAP";
    }
    leaf multi-src {
      if-feature "multi-src-dest";
      type boolean;
      default "false";
    }
  }
}
```

```
        description
            "Is the 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 "/access-point/ap/ap-id";
        }
        description
            "A reference to destination AP";
    }
    leaf dest-vn-ap-id {
        type leafref {
            path "/access-point/ap/vn-ap/vn-ap-id";
        }
        description
            "A reference to dest VNAP";
    }
    leaf multi-dest {
        if-feature "multi-src-dest";
        type boolean;
        default "false";
        description
            "Is destination part of multi-destination, where only one
            of the destination is enabled";
    }
}
leaf connectivity-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
        "A reference to connectivity-matrix";
    reference
        "RFC 8795: YANG Data Model for Traffic Engineering (TE)
        Topologies";
}
container underlay {
    description
        "An empty container that can be augmented with underlay
        technology information not supported by RFC 8795 (for
```

```
        example - Segement Routing (SR). ";
    }
    reference
        "RFC 8454: Information Model for Abstraction and Control of TE
        Networks (ACTN)";
    } //vn-member

    grouping vn-policy {
        description
            "policy for VN-level diverisity";
        leaf vn-level-diversity {
            type te-types:te-path-disjointness;
            description
                "The type of disjointness on the VN level (i.e., across all
                VN members)";
        }
    }
}

/* Configuration data nodes */

container access-point {
    description
        "AP configurations";
    list ap {
        key "ap-id";
        description
            "access-point identifier";
        uses access-point {
            description
                "The access-point information";
        }
    }
}

reference
    "RFC 8453: Framework for Abstraction and Control of TE
    Networks (ACTN), Section 6";
}

container virtual-network {
    description
        "VN configurations";
    list vn {
        key "vn-id";
        description
            "A virtual network is identified by a vn-id";
        leaf vn-id {
            type vn-id;
            description
                "A unique VN identifier";
        }
    }
}
```



```
/*An optional identifier to the TE Topology Model
   where the abstract nodes and links of the Topology
   can be found for Type 2 VNS*/
uses te-types:te-topology-identifier;
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 {
  key "vnm-id";
  description
    "List of vn-members in a VN";
  uses vn-member;
  leaf oper-status {
    type te-types:te-oper-status;
    config false;
    description
      "The 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";
}
leaf admin-status {
  type te-types:te-admin-status;
  default "up";
  description
    "VN administrative state.";
}
leaf oper-status {
  type te-types:te-oper-status;
  config false;
  description
    "VN operational state.";
}
uses vn-policy;
} //vn
reference
  "RFC 8453: Framework for Abstraction and Control of TE
```

```
    Networks (ACTN)";
} //vn

/* RPC */

rpc vn-compute {
  description
    "The VN computation without actual instantiation. This is
    used by the CNC to get the VN results without actually
    creating it in the network.

    The input could include a reference to the single node
    abstract topology. It could optionally also include
    constraints and optimization criteria. The computation
    is done based on the list of VN-members.

    The output includes a reference to the single node
    abstract topology with each VN-member including a
    reference to the connectivity-matrix-id where the
    path properties could be found. Error information is
    also included.";
  input {
    uses te-types:te-topology-identifier;
    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";
    }
    uses te-types:generic-path-constraints;
    leaf cos {
      type te-types:te-ds-class;
      description
        "The class of service";
    }
  }
  uses te-types:generic-path-optimization;
  list vn-member-list {
    key "vnm-id";
    description
      "List of VN-members in a VN";
    uses vn-member;
    uses te-types:generic-path-constraints;
    leaf cos {
      type te-types:te-ds-class;
      description
        "The class of service";
      reference

```

```
        "RFC 4124: Protocol Extensions for Support of
        Diffserv-aware MPLS Traffic Engineering,
        Section 4.3.1";
    }
    uses te-types:generic-path-optimization;
}
uses vn-policy;
}
output {
    uses te-types:te-topology-identifier;
    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 "vnm-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";
            reference
                "RFC 8453: Framework for Abstraction and Control of TE
                Networks (ACTN), Section 7";
        }
        leaf compute-status {
            type vn-compute-status;
            description
                "The VN-member compute state.";
        }
    }
    container error-info {
        description
            "Error information related to the VN member";
        leaf error-description {
            type string;
            description
                "Textual representation of the error occurred during
                VN compute.";
        }
        leaf error-timestamp {
```

```

        type yang:date-and-time;
        description
            "Timestamp of the attempt.";
    }
    leaf error-reason {
        type identityref {
            base vn-computation-error-reason;
        }
        description
            "Reason for the VN computation error.";
    }
}
}
}
} //vn-compute
}
<CODE ENDS>

```

7. JSON Example

This section provides json implementation examples as to how 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 VN YANG model also include the AP and VNAP which shows various VN using the same AP.

7.1. VN JSON

```
{
  "access-point": {
    "ap": [
      {
        "ap-id": "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"
          }
        ]
      },
      {
        "ap-id": "202",
        "vn-ap": [
          {
            "vn-ap-id": "20201",
            "vn": "1",
            "abstract-node": "D1",
            "ltp": "2-0-2"
          }
        ]
      },
      {
        "ap-id": "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"
        }
      ]
    },
    {
      "ap-id": "440",
      "vn-ap": [
        {
          "vn-ap-id": "44001",
          "vn": "1",
          "abstract-node": "D1",
          "ltp": "4-4-0"
        }
      ]
    },
    {
      "ap-id": "550",
      "vn-ap": [
        {
          "vn-ap-id": "55002",
          "vn": "2",
          "abstract-node": "D2",
          "ltp": "5-5-0"
        }
      ]
    },
    {
      "ap-id": "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"
        }
      ]
    },
    {

```

```
    "ap-id": "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"
      }
    ]
  }
],
"virtual-network": {
  "vn": [
    {
      "vn-id": "1",
      "te-topology-identifier": {
        "topology-id": "abstract1"
      },
      "abstract-node": "D1",
      "vn-member": [
        {
          "vnm-id": "104",
          "src": {
            "src": "101",
            "src-vn-ap-id": "10101"
          },
          "dest": {
            "dest": "440",
            "dest-vn-ap-id": "44001"
          },
          "connectivity-matrix-id": "104"
        },
        {
          "vnm-id": "107",
          "src": {
            "src": "101",
            "src-vn-ap-id": "10101"
          },
          "dest": {
            "dest": "770",
            "dest-vn-ap-id": "77001"
          }
        }
      ]
    }
  ]
}
```

```
    },
    "connectivity-matrix-id": "107"
  },
  {
    "vnm-id": "204",
    "src": {
      "src": "202",
      "dest-vn-ap-id": "20401"
    },
    "dest": {
      "dest": "440",
      "dest-vn-ap-id": "44001"
    },
    "connectivity-matrix-id": "204"
  },
  {
    "vnm-id": "308",
    "src": {
      "src": "303",
      "src-vn-ap-id": "30301"
    },
    "dest": {
      "dest": "880",
      "src-vn-ap-id": "88001"
    },
    "connectivity-matrix-id": "308"
  },
  {
    "vnm-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",
  "te-topology-identifier": {
    "topology-id": "abstract2"
  },
  "abstract-node": "D2",
  "vn-member": [
```



```
    {
      "vnm-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",
  "te-topology-identifier": {
    "topology-id": "abstract3"
  },
  "abstract-node": "D3",
  "vn-member": [
    {
      "vnm-id": "104",
      "src": {
        "src": "101"
      },
      "dest": {
        "dest": "440",
        "multi-dest": true
      }
    },
    {
      "vnm-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
    },
    {
      "vnm-id": "204",
      "src": {
```

```

        "src": "202",
        "multi-src": true
    },
    "dest": {
        "dest": "440"
    }
},
{
    "vnm-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",
        "te-topology-identifier": {
          "provider-id": 0,
          "client-id": 0,
          "topology-id": "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": {
    "te-bandwidth": {
      "generic": "0x1p10"
    },
    "disjointness": "node link srlg"
  },
"connectivity-matrix": [
  {
    "id": 104,
    "from": {
      "tp-ref": "1-0-1"
    },
    "to": {
      "tp-ref": "4-4-0"
    }
  },
  {
    "id": 107,
    "from": {
      "tp-ref": "1-0-1"
    },
    "to": {
      "tp-ref": "7-7-0"
    }
  },
  {
    "id": 204,
    "from": {
      "tp-ref": "2-0-2"
    },
    "to": {
      "tp-ref": "4-4-0"
    }
  },
  {
    "id": 308,
    "from": {
      "tp-ref": "3-0-3"
    },
    "to": {
      "tp-ref": "8-8-0"
    }
  }
]
```

```
    },
    {
      "id": 108,
      "from": {
        "tp-ref": "1-0-1"
      },
      "to": {
        "tp-ref": "8-8-0"
      }
    }
  ]
}
},
"tunnel-termination-point": [
  {
    "name": "1-0-1",
    "tunnel-tp-id": 10001,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "1-1-0",
    "tunnel-tp-id": 10100,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "2-0-2",
    "tunnel-tp-id": 20002,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "2-2-0",
    "tunnel-tp-id": 20200,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "3-0-3",
    "tunnel-tp-id": 30003,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "3-3-0",
    "tunnel-tp-id": 30300,
    "switching-capability": "switching-otn",
```

```
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "4-0-4",
    "tunnel-tp-id": 40004,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "4-4-0",
    "tunnel-tp-id": 40400,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "5-0-5",
    "tunnel-tp-id": 50005,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "5-5-0",
    "tunnel-tp-id": 50500,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "6-0-6",
    "tunnel-tp-id": 60006,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "6-6-0",
    "tunnel-tp-id": 60600,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "7-0-7",
    "tunnel-tp-id": 70007,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  },
  {
    "name": "7-7-0",
    "tunnel-tp-id": 70700,
    "switching-capability": "switching-otn",
```

```

        "encoding": "lsp-encoding-oduk"
    },
    {
        "name": "8-0-8",
        "tunnel-tp-id": 80008,
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
    },
    {
        "name": "8-8-0",
        "tunnel-tp-id": 80800,
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
    }
  ]
}
]
},
{
  "network-types": {
    "te-topology": {}
  },
  "network-id": "abstract2",
  "te-topology-identifier": {
    "provider-id": 0,
    "client-id": 0,
    "topology-id": "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": {
          "te-bandwidth": {
            "generic": "0x1p10"
          }
        }
      }
    }
  ]
}

```

```

    },
    "optimizations": {
      "objective-function": {
        "objective-function-type": "of-maximize-residual-bandwidth"
      }
    },
    "connectivity-matrix": [
      {
        "id": 105,
        "from": {
          "tp-ref": "1-0-1"
        },
        "to": {
          "tp-ref": "5-5-0"
        },
        "underlay": {
          "enabled": true,
          "primary-path": {
            "network-ref": "absolute",
            "path-element": [
              {
                "path-element-id": 1,
                "numbered-node-hop": {
                  "node-id": "4.4.4.4",
                  "hop-type": "strict"
                }
              },
              {
                "path-element-id": 2,
                "numbered-hop": {
                  "node-id": "7.7.7.7",
                  "hop-type": "strict"
                }
              }
            ]
          }
        }
      }
    ]
  },
  "tunnel-termination-point": [
    {
      "name": "1-0-1",
      "tunnel-tp-id": 10001,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ],

```

```
{
  "name": "1-1-0",
  "tunnel-tp-id": 10100,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "2-0-2",
  "tunnel-tp-id": 20002,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "2-2-0",
  "tunnel-tp-id": 20200,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "3-0-3",
  "tunnel-tp-id": 30003,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "3-3-0",
  "tunnel-tp-id": 30300,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "4-0-4",
  "tunnel-tp-id": 40004,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "4-4-0",
  "tunnel-tp-id": 40400,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
{
  "name": "5-0-5",
  "tunnel-tp-id": 50005,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
},
},
```



```

    {
      "name": "5-5-0",
      "tunnel-tp-id": 50500,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "6-0-6",
      "tunnel-tp-id": 60006,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "6-6-0",
      "tunnel-tp-id": 60600,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "7-0-7",
      "tunnel-tp-id": 70007,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "7-7-0",
      "tunnel-tp-id": 70700,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "8-0-8",
      "tunnel-tp-id": 80008,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "8-8-0",
      "tunnel-tp-id": 80800,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
}
]
},
{

```

```
"network-types": {
  "te-topology": {}
},
"network-id": "abstract3",
"te-topology-identifier": {
  "provider-id": 0,
  "client-id": 0,
  "topology-id": "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": {
          "te-bandwidth": {
            "generic": "0x1p10"
          }
        }
      },
      "connectivity-matrix": [
        {
          "id": 107,
          "from": {
            "tp-ref": "1-0-1"
          },
          "to": {
            "tp-ref": "7-7-0"
          }
        },
        {
          "id": 308,
          "from": {
            "tp-ref": "3-0-3"
          },
          "to": {
            "tp-ref": "8-8-0"
          }
        }
      ]
    }
  }
],
},
```

```
"tunnel-termination-point": [  
  {  
    "name": "1-0-1",  
    "tunnel-tp-id": 10001,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "1-1-0",  
    "tunnel-tp-id": 10100,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "2-0-2",  
    "tunnel-tp-id": 20002,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "2-2-0",  
    "tunnel-tp-id": 20200,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "3-0-3",  
    "tunnel-tp-id": 30003,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "3-3-0",  
    "tunnel-tp-id": 30300,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "4-0-4",  
    "tunnel-tp-id": 40004,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  },  
  {  
    "name": "4-4-0",  
    "tunnel-tp-id": 40400,  
    "switching-capability": "switching-otn",  
    "encoding": "lsp-encoding-oduk"  
  }  
]
```

```
    },
    {
      "name": "5-0-5",
      "tunnel-tp-id": 50005,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "5-5-0",
      "tunnel-tp-id": 50500,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "6-0-6",
      "tunnel-tp-id": 60006,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "6-6-0",
      "tunnel-tp-id": 60600,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "7-0-7",
      "tunnel-tp-id": 70007,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "7-7-0",
      "tunnel-tp-id": 70700,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "8-0-8",
      "tunnel-tp-id": 80008,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    },
    {
      "name": "8-8-0",
      "tunnel-tp-id": 80800,
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ],
  {
    "name": "8-8-0",
    "tunnel-tp-id": 80800,
    "switching-capability": "switching-otn",
    "encoding": "lsp-encoding-oduk"
  }
],
{
  "name": "8-8-0",
  "tunnel-tp-id": 80800,
  "switching-capability": "switching-otn",
  "encoding": "lsp-encoding-oduk"
}
```

```
}
}
}
}
}
}
}
}
```

8. Security Considerations

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] 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 model presented in this document is used in the interface between the Customer Network Controller (CNC) and Multi-Domain Service Coordinator (MDSC), which is referred to as CNC-MDSC Interface (CMI). Therefore, many security risks such as malicious attack and rogue elements attempting to connect to various ACTN components. Furthermore, some ACTN components (e.g., MSDC) represent a single point of failure and threat vector and must also manage policy conflicts and eavesdropping of communication between different ACTN components.

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.

These are the subtrees and data nodes and their sensitivity/vulnerability:

* ap:

- ap-id
- max-bandwidth
- avl-bandwidth

- * vn-ap:
 - vn-ap-id
 - vn
 - abstract-node
 - ltp
- * vn
 - vn-id
 - vn-topology-id
 - abstract-node
- * vnm-id
 - src
 - src-vn-ap-id
 - dest
 - dest-vn-ap-id
 - connectivity-matrix-id

9. IANA Considerations

IANA is requested to make the following allocation for the URIs in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

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

IANA is requested to make the following allocation for the YANG module in the "YANG Module Names" registry [RFC6020]:

```
-----  
name:      ietf-vn  
namespace: urn:ietf:params:xml:ns:yang:ietf-vn  
prefix:    vn  
reference:  RFC XXXX  
-----
```

10. Acknowledgments

The authors would like to thank Xufeng Liu, Adrian Farrel, and Tom Petch for their helpful comments and valuable suggestions.

Thanks to Andy Bierman for YANGDIR review.

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>>.
- [RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering", RFC 4124, DOI 10.17487/RFC4124, June 2005, <<https://www.rfc-editor.org/info/rfc4124>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.

- [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>>.

11.2. Informative References

- [I-D.ietf-ccamp-llcsm-yang]
Lee, Y., Lee, K., Zheng, H., Dios, O. G. D., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", Work in Progress, Internet-Draft, draft-ietf-ccamp-llcsm-yang-16, 13 December 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-ccamp-llcsm-yang-16>>.
- [I-D.ietf-teas-actn-pm-telemetry-autonomics]
Lee, Y., Dhody, D., Karunanithi, S., Vilalta, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", Work in Progress, Internet-Draft, draft-ietf-teas-actn-pm-telemetry-autonomics-07, 23 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-actn-pm-telemetry-autonomics-07>>.
- [I-D.ietf-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Fioccola, G., Wu, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping YANG Model", Work in Progress, Internet-Draft, draft-ietf-teas-te-service-mapping-yang-09, 24 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-te-service-mapping-yang-09>>.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. D. Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-29, 7 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-29>>.
- [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>>.

- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.

Appendix A. Performance Constraints

At the time of creation of VN, it is natural to provide VN level constraints and optimization criteria. It should be noted that this YANG model rely on the TE-Topology Model [RFC8795] by using a reference to an abstract node to achieve this. Further, connectivity-matrix structure is used to assign the constraints and optimization criteria include delay, jitter etc. [RFC8776] define some of the metric-types already and future documents are meant to augment it.

Note that the VN compute allows inclusion of the constraints and the optimization criteria directly in the RPC to allow it to be used independently.

Appendix B. Contributors Addresses

Qin Wu
Huawei Technologies
Email: bill.wu@huawei.com

Peter Park
KT
Email: peter.park@kt.com

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

Takuya Miyasaka
KDDI
Email: ta-miyasaka@kddi.com

Kenichi Ogaki
KDDI
Email: ke-oogaki@kddi.com

Authors' Addresses

Young Lee (editor)
Samsung Electronics
Email: younglee.tx@gmail.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore 560066
Karnataka
India
Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan, 48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Igor Bryskin
Individual
Email: i_bryskin@yahoo.com

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

TEAS WG
Internet Draft
Intended status: Informational
Expires: September 7, 2022

Young Lee
Samsung
Haomian Zheng
Huawei
Daniele Ceccarelli
Ericsson
Bin Yeong Yoon
ETRI
Sergio Belotti
Nokia

March 7, 2022

Applicability of YANG models for Abstraction and Control of Traffic
Engineered Networks

draft-ietf-teas-actn-yang-09

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.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 8, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Conventions Used in This Document	3
2. Abstraction and Control of TE Networks (ACTN) Architecture ..	3
3. Service Models	5
4. Service Model Mapping to ACTN	7
4.1. Customer Service Models in the ACTN Architecture (CMI) ..	7
4.2. Service Delivery Models in ACTN Architecture	8
4.3. Network Configuration Models in ACTN Architecture (MPI) ..	8
4.4. Device Models in ACTN Architecture (SBI)	9
5. Examples of Using Different Types of YANG Models	10
5.1. Topology Collection	10
5.2. Connectivity over Two Nodes	10
5.3. VN example	11
5.4. Data Center-Interconnection Example	12

5.4.1. CMI (CNC-MDSC Interface)	14
5.4.2. MPI (MDSC-PNC Interface)	14
5.4.3. SBI (Southbound interface between PNC and devices)	14
6. Security	15
7. IANA Considerations	15
8. Acknowledgements	15
9. References	15
9.1. Informative References.....	15
10. Contributors	18
Authors' Addresses	18

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 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 [RFC6241] 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 [RFC8199] and [RFC8309].

This document shows how the ACTN architecture can be satisfied using various 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.

1.1. 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.

2. Abstraction and Control of TE Networks (ACTN) Architecture

[RFC8453] describes the architecture model for ACTN including the entities (Customer Network Controller (CNC), Multi-domain Service

Coordinator (MDSC), and Provisioning 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 [RFC8453]. A number of key ACTN interfaces exist for deployment and operation of ACTN-based networks. These are highlighted in Figure 1 (ACTN Interfaces) below:

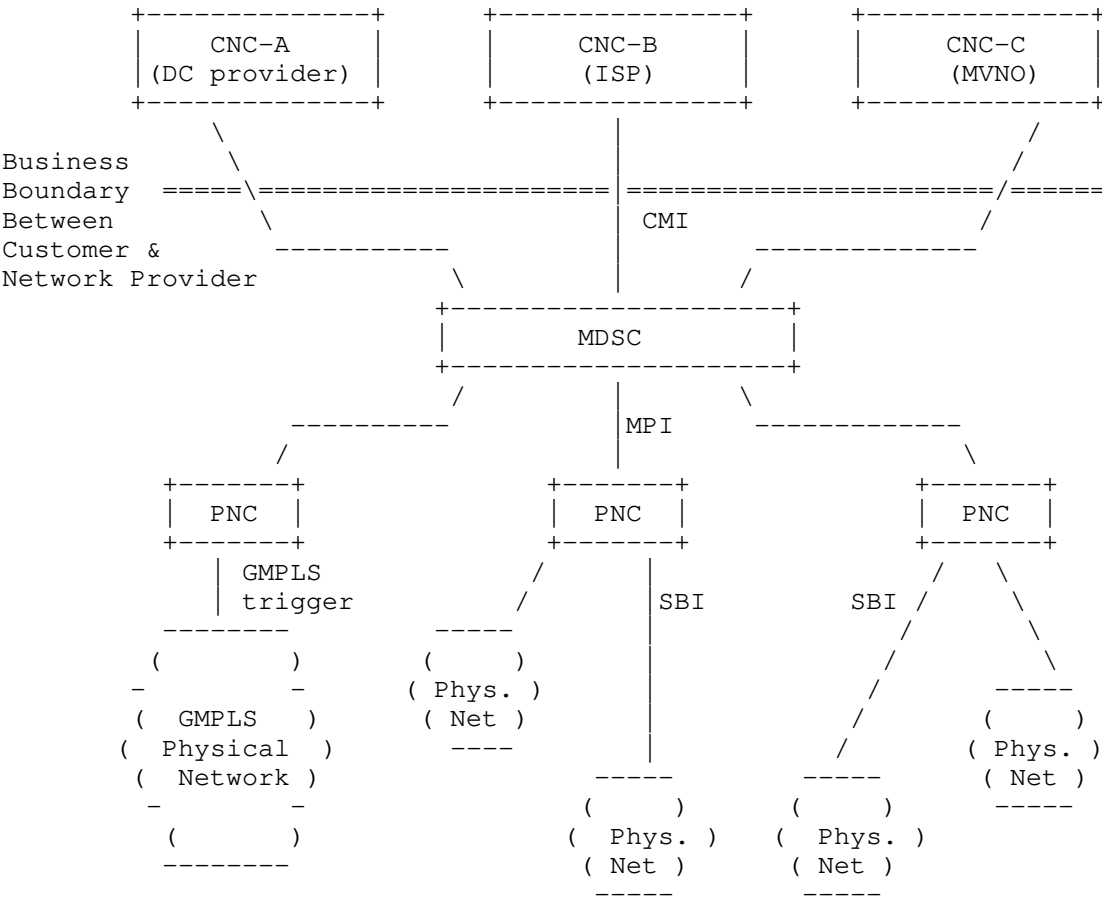


Figure 1 : ACTN Interfaces

The interfaces and functions are described below (without modifying the definitions) in [RFC8453]:

The CNC-MDSC Interface (CMI) is an interface between a CNC and an MDSC. This interface is used to 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 together with 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. Performance monitoring is also applicable in CMI, which enables the MDSC to report network parameters/telemetries that may guide the CNC to create/change their services.

The MDSC-PNC Interface (MPI) is an interface between a MDSC and a PNC. 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. MPI plays an important role for multi-vendor mechanism, interoperability can be achieved by standardized interface modules.

The South-Bound Interface (SBI) is the provisioning interface for creating forwarding state in the physical network, requested via the PNC. 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 [RFC8309].

3. Service Models

[RFC8309] 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 [RFC8309]. 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.

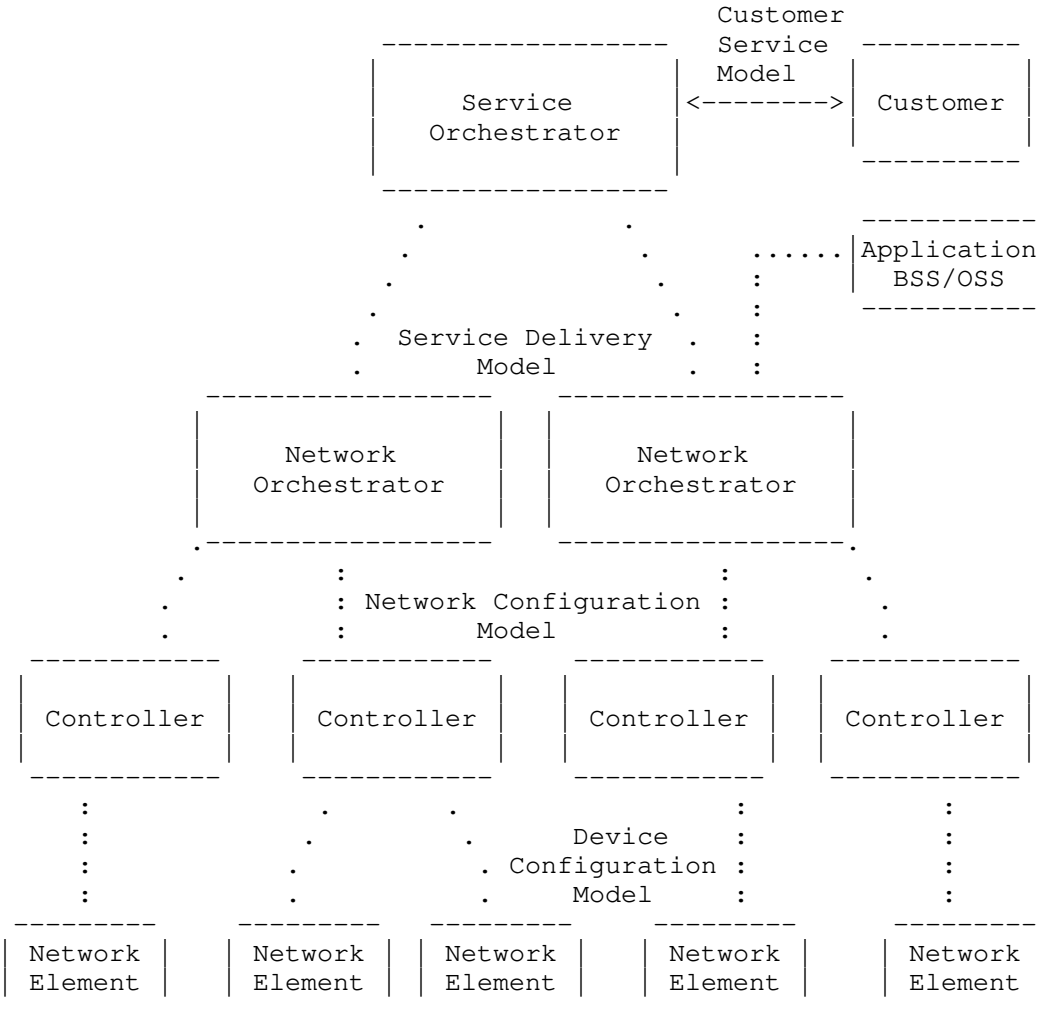


Figure 2: An SDN Architecture with a Service Orchestrator

4. Service Model Mapping to ACTN

YANG models coupled with the RESTCONF/NETCONF protocol [RFC6241][RFC8040] 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 [RFC8453] for details of the mapping between ACTN functions and service models. In summary, the following mappings are held between and Service Yang Models in [RFC8309] and the ACTN interfaces in [RFC8453].

- 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 [RFC8309], 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

VN Service Request	[ACTN-VN-YANG]
VN Computation Request	[ACTN-VN-YANG]*
TE & Service Mapping	[TE-Service-Mapping]**
VN Performance Monitoring Telemetry	[ACTN-PM-Telemetry]***
Topology Abstraction	[RFC8795]****
Layer 1 Connectivity Service Model	[L1CSM]
Layer 2 VPN Service Model	[RFC8466]
Layer 3 VPN Service Model	[RFC8299]

*VN 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.

**[TE-Service-Mapping] provides a mapping and cross-references between service models (e.g., L3SM, L2SM, L1CSM) and TE model via [ACTN-VN-YANG] and [RFC8795]. This model can be used as either Customer Service Models, or Service Delivery model described in Section 4.2.

***ietf-actn-te-kpi-telemetry model in [ACTN-PM-Telemetry] 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 autonomies 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.

****RFC8795's Connectivity Matrices/Matrix construct can be used to instantiate VN Service via a suitable referencing and mapping with [ACTN-VN-YANG].

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 [RFC8309]. 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 [RFC8309]. 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]
Tunnel/LSP Provisioning	[TE-tunnel]
Topology Abstraction	[RFC8795]
Service Provisioning	[Client-signal]&[TE-tunnel]*
Client Topology Abstraction	[Client-topo]
OTN Topology Abstraction	[OTN-topo]
WSON Topology Abstraction	[WSON-topo]
Flexi-grid Topology Abstraction	[Flexi-topo]
Microwave Topology Abstraction	[MW-topo]
Client Signal Description	[Client-signal]
OTN Tunnel Model	[OTN-Tunnel]
WSON TE Tunnel Model	[WSON-Tunnel]
Flexi-grid Tunnel Model	[Flexigrid-Tunnel]

* This function is a combination of tunnel set up and client signal description. Usually a tunnel is setting up first to get prepared to carry a client signal, in order to do the service provisioning. Then the client signal is adapted to the established tunnel. It is worth noting that various tunnel models such as [OTN-Tunnel] and [WSON-Tunnel] can be used together with the [TE-tunnel] model to construct technology-specific tunnels, and carry different types of client signals. More details can be found in [Client-signal].

[TE-topo-tunnel] provides the clarification and example usage for TE topology model [RFC8795] and TE tunnel model [TE-tunnel]. [T-NBI Applicability] provides a summary on the applicability of existing YANG model usage in the current network configuration, especially for transport network.

4.4. Device Models in ACTN Architecture (SBI)

Note that SBI is not in the scope of ACTN, as there is already mature protocol solutions for various purpose on the device level of ACTN architecture, such as RSVP-TE, OSPF-TE and so on. The interworking of such protocols and ACTN controller hierarchies can be found in [gmpls-controller-inter-work].

For the device YANG models are used for per-device configuration purpose, they can be used between the PNC and the physical network/devices. One example of Device Models is ietf-te-device yang module defined in [TE-tunnel].

5. Examples of Using Different Types of YANG Models

This section provides some examples on the usage of IETF YANG models in the network operation. A few typical generic scenarios are involved. In [T-NBI Applicability], there are more transport-related scenarios and examples.

5.1. Topology Collection

Before any connection is requested and delivered, the controller needs to understand the network topology. The topology information is exchanged among controllers with topology models, such as [RFC8795]. Moreover, technology-specific topology reporting may use the model described in [OTN-topo] [WSON-topo], and [Flexi-topo] for OTN, WSON and Flexi-grid, respectively. By collecting the network topology, each controller can therefore construct a local database, which can be used for the further service deployment.

There can be different types of abstraction applied between each pair of controllers, corresponding method can be found in [RFC8453]. The technology-specific features may be hidden after abstraction, to make the network easier for the user to operate.

When there is a topology change in the physical network, the PNC should report the change to upper level of controllers via updating messages using topology models. Accordingly, such changes is propagated between different controllers for further synchronization.

5.2. Connectivity over Two Nodes

The service models, such as described in [RFC8299], [RFC8466] and [L1CSM] provide a customer service model which can be used in provider networks.

It would be used as follows in the ACTN architecture:

A CNC uses the service models 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.

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]. Technology-specific tunnel configuration may use the model described in [OTN-Tunnel] [WSON-Tunnel], and [Flexigrid-Tunnel] for OTN, WSON and Flexi-grid, respectively.

Then, the PNCs need to decide the explicit route of such a tunnel or tunnel segment (in case of multiple domains) for each domain, and then create such a tunnel using protocols such as PCEP and RSVP-TE or using per-hop configuration.

5.3. VN 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 evaluate each candidate and then choose the best one 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 example in section 5.2.

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].

5.4. 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 [RFC8454].

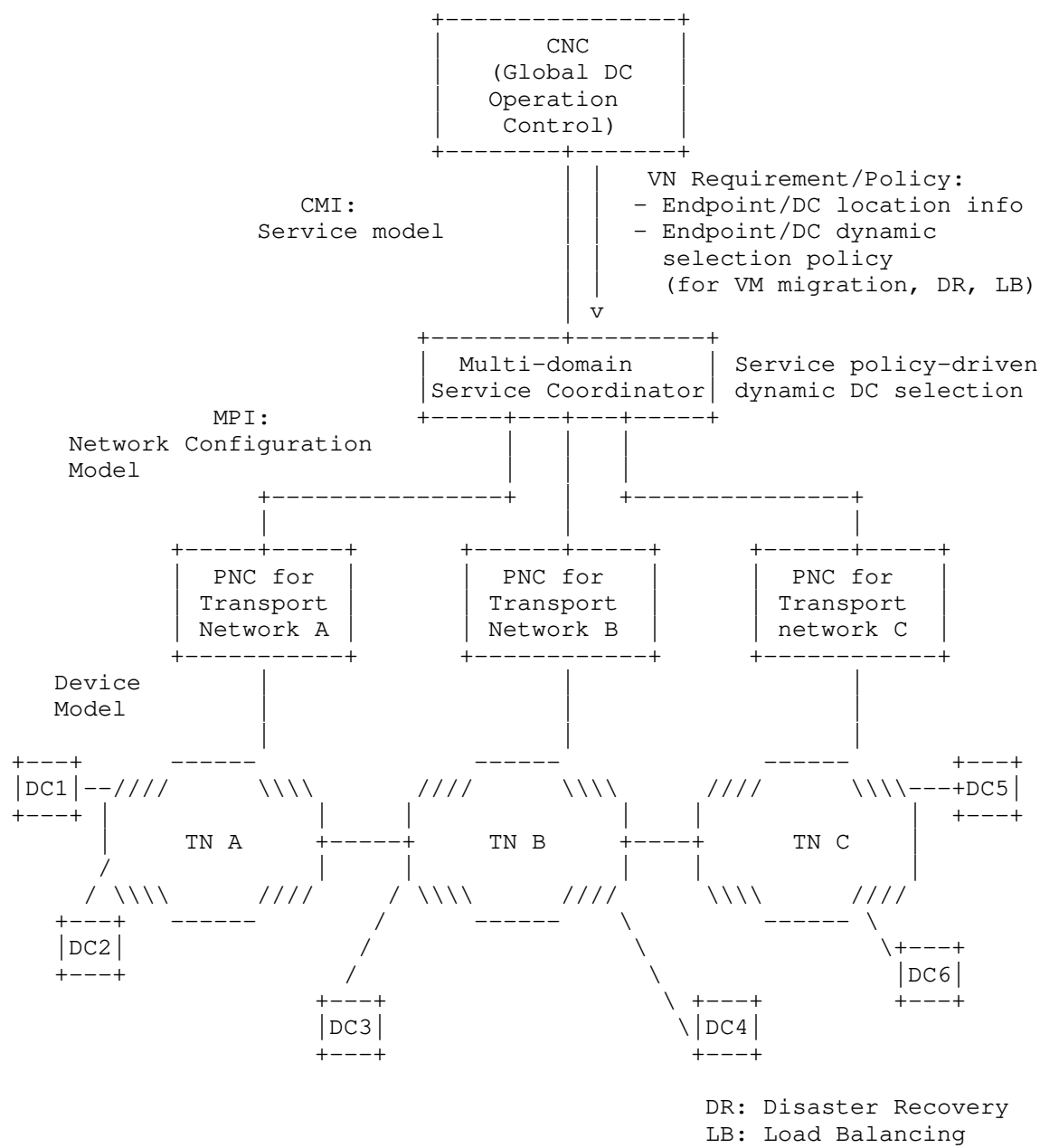


Figure 3: Service Policy-driven Data Center Selection

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.

5.4.1. 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.

5.4.2. 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. [RFC8795] is used to for the purpose of underlying domain network abstraction from the PNC to the MDSC.

5.4.3. SBI (Southbound interface between PNC and devices)

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 employs (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. IANA Considerations

This document requires no IANA actions.

8. Acknowledgements

We thank Adrian Farrel for providing useful comments and suggestions for this draft.

9. References

9.1. Informative 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>>.
- [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>>.
- [RFC8309] Q. Wu, W. Liu and A. Farrel, "Service Models Explained", RFC 8309.

- [RFC8199] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification", RFC 8199.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241.
- [RFC8040] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol", RFC 8040.
- [RFC8453] D. Ceccarelli and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC8453.
- [RFC8466] B. Wen, G. Fioccola, C. Xie, L. Jalil, "A YANG Data Model for L2VPN Service Delivery", RFC8466.
- [RFC8299] Q. Wu, S. Litkowski, L. Tomotaki, K.Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC8299.
- [RFC8454] Y. Lee & S. Belotti, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC8454.
- [RSVP-TE-YANG] T. Saad (Editor), "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp, work in progress.
- [RFC8795] X. Liu, et. al., "YANG Data Model for TE Topologies", RFC8795.
- [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.
- [L1CSM] G. Fioccola, K. Lee, Y. Lee, D. Dhody, O. Gonzalez de-Dios, D. Ceccarelli, "A Yang Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang, work in progress.
- [Schedule] X. Liu, et. al., "A YANG Data Model for Configuration Scheduling", draft-liu-netmod-yang-schedule, work in progress.

- [OTN-topo] H. Zheng, et. al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.
- [RFC9094] Y. Lee, et. al., "A Yang Data Model for WSON Optical Networks", RFC 9094.
- [Flexi-topo] J.E. Lopez de Vergara, et. al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.
- [MW-topo] M. Ye, et. al., "A YANG Data Model for Microwave Topology", draft-ietf-ccamp-mw-topo-yang, work in progress.
- [OTN-Tunnel] H. Zheng, et. al., "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-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-ietf-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-ietf-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-ietf-ccamp-flexigrid-media-channel-yang, work in progress.
- [Client-signal] H. Zheng, et al, "A YANG Data Model for Optical Transport Network Client Signals", draft-ietf-ccamp-client-signal-yang, work in progress.
- [Client-topo] H. Zheng, et al, "A YANG Data Model for Ethernet TE Topology", draft-ietf-ccamp-eth-client-te-topo-yang, work in progress.
- [TE-topo-tunnel] I. Bryskin, et. al., "TE Topology and Tunnel Modeling for Transport Networks", draft-ietf-teas-te-topo-and-tunnel-modeling, work in progress.

[T-NBI Applicability] I. Busi, et al, "Transport Northbound Interface Applicability Statement and Use Cases", draft-ietf-ccamp-transport-nbi-app-statement, work in progress.

[gmpls-controller-inter-work] H. Zheng, et al, "Interworking of GMPLS Control and Centralized Controller System", draft-ietf-teas-gmpls-controller-inter-work, work in progress.

10. Contributors

Contributor's Addresses

Dhruv Dhody
Huawei Technologies

Email: dhruv.ietf@gmail.com

Xian Zhang
Huawei Technologies

Email: zhang.xian@huawei.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Jong Yoon Shin
SKT
Email: jongyoon.shin@sk.com

Authors' Addresses

Young Lee
Samsung
Korea
Email: younglee.tx@gmail.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

Sergio Belotti
Nokia
Email: sergio.belotti@nokia.com

TEAS Working Group
Internet-Draft
Intended status: Informational
Expires: 8 September 2022

J. Dong
Huawei
S. Bryant
University of Surrey
Z. Li
China Mobile
T. Miyasaka
KDDI Corporation
Y. Lee
Samsung
7 March 2022

A Framework for Enhanced Virtual Private Network (VPN+) Services
draft-ietf-teas-enhanced-vpn-10

Abstract

This document describes the framework for Enhanced Virtual Private Network (VPN+) services. The purpose of enhanced VPNs is to support the needs of new applications, particularly applications that are associated with 5G services, by utilizing an approach that is based on the VPN and Traffic Engineering (TE) technologies and adds characteristics that specific services require over those provided by traditional VPNs.

Typically, VPN+ will be used to underpin network slicing, but could also be of use in its own right providing enhanced connectivity services between customer sites.

It is envisaged that enhanced VPNs will be delivered using a combination of existing, modified, and new networking technologies. This document provides an overview of relevant technologies and identifies some areas for potential new work.

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 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	6
3. Overview of the Requirements	7
3.1. Performance Guarantees	7
3.2. Isolation between Enhanced VPN Services	8
3.2.1. A Pragmatic Approach to Isolation	10
3.3. Integration	11
3.3.1. Abstraction	11
3.4. Dynamic Changes	12
3.5. Customized Control	12
3.6. Applicability to Overlay Technologies	13
3.7. Inter-Domain and Inter-Layer Network	13
4. Architecture of Enhanced VPNs	14
4.1. Layered Architecture	16
4.2. Multi-Point to Multi-Point (MP2MP) Connectivity	18
4.3. Application Specific Data Types	18
4.4. Scaling Considerations	18
5. Candidate Technologies	19
5.1. Packet Forwarding Plane Technologies	20
5.1.1. Flexible Ethernet	20
5.1.2. Dedicated Queues	20
5.1.3. Time Sensitive Networking	21
5.2. Layer Three Data Plane	21
5.2.1. Deterministic Networking	21
5.2.2. MPLS Traffic Engineering (MPLS-TE)	22

5.2.3. Segment Routing	22
5.3. Non-Packet Data Plane	23
5.4. Control Plane	23
5.5. Management Plane	24
5.6. Applicability of Service Data Models to Enhanced VPN . .	26
6. Applicability to Network Slice Realization	26
6.1. VTN Planning	27
6.2. VTN Instantiation	27
6.3. VPN+ Service Provisioning	27
6.4. Network Slice Traffic Steering and Forwarding	28
7. Scalability Considerations	28
7.1. Maximum Stack Depth of SR	29
7.2. RSVP-TE Scalability	29
7.3. SDN Scaling	29
8. OAM Considerations	30
9. Telemetry Considerations	30
10. Enhanced Resiliency	31
11. Operational Considerations	32
12. Security Considerations	32
13. IANA Considerations	33
14. Contributors	33
15. Acknowledgements	33
16. References	34
16.1. Normative References	34
16.2. Informative References	34
Authors' Addresses	40

1. Introduction

Virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated connectivity over a common network. The common or base network that is used to provide the VPNs is often referred to as the underlay, and the VPN is often called an overlay.

Customers of a network operator may request a connectivity services with advanced characteristics such as low latency guarantees, bounded jitter, or isolation from other services or customers so that changes in some other service (such as changes in network load, or events such as congestion or outages) have no or only acceptable effect on the throughput or latency of the services provided to the customer. These services are referred to as "enhanced VPNs" (known as VPN+) in that they are similar to VPN services providing the customer with the required connectivity, but in addition they have enhanced characteristics.

The concept of network slicing has gained traction driven largely by needs surfacing from 5G [NGMN-NS-Concept] [TS23501] [TS28530]. According to [TS28530], a 5G end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN), and Mobile Core Network (CN). The transport network provides the connectivity between different entities in RAN and CN segments of a 5G end-to-end network slice, with specific performance commitment.

[I-D.ietf-teas-ietf-network-slices] defines the terminologies and the characteristics of IETF network slices. It also discusses the general framework, the components and interfaces for requesting and operating IETF network slices. An IETF Network Slice Service enables connectivity between a set of CEs with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network. An IETF Network Slice can be realized as a logical network connecting a number of endpoints and is associated with a set of shared or dedicated network resources that are used to satisfy the Service Level Objectives (SLOs) and Service Level Expectations (SLEs) requirements. In this document (which is solely about IETF technologies) we refer to an "IETF network slice" simply as a "network slice": a network slice is considered one possible use case of an enhanced VPN.

A network slice could span multiple technologies (such as IP or Optical) and multiple administrative domains. Depending on the customer's requirement, a network slice could be isolated from other network slices in terms of data plane, control plane, and management plane resources.

Network slicing builds on the concepts of resource management, network virtualization, and abstraction to provide performance assurance, flexibility, programmability, and modularity. It may use techniques such as Software Defined Networking (SDN) [RFC7149], network abstraction [RFC7926] and Network Function Virtualization (NFV) [RFC8172] [RFC8568] to create multiple logical (virtual) networks, each tailored for use by a set of services or by a particular tenant or a group of tenants that share the same or similar requirements. These logical networks are created on top of a common underlay network. How the network slices are engineered can be deployment-specific.

The requirements of enhanced VPN services cannot be met by simple overlay networks, as these services require tighter coordination and integration between the underlay and the overlay network. VPN+ is built from a VPN overlay and an underlying Virtual Transport Network (VTN) which has a customized network topology and a set of dedicated or shared resources in the underlay network. The enhanced VPN may

also include a set of invoked service functions located within the underlay network. Thus, an enhanced VPN can achieve greater isolation with strict performance guarantees. These new properties, which have general applicability, are also of interest as part of a network slicing solution.

VPN+ can be used to instantiate a network slice service, and the technique can also be of use in general cases to provide enhanced connectivity services between customer sites or service end points. [I-D.ietf-teas-ietf-network-slices] introduces the concept Network Resource Partition (NRP) as a set of network resources that are available to carry traffic and meet the SLOs and SLEs. An NRP is associated with a network topology to define the set of links and nodes. Thus VTN and NRP are considered as similar concepts, and NRP can be seen as an instantiation of VTN in the context of network slicing.

It is not envisaged that VPN+ services will replace traditional VPN services. Traditional VPN services will continue to be delivered using pre-existing mechanisms and can co-exist with VPN+ services.

This document describes a framework for using existing, modified, and potential new technologies as components to provide a VPN+ service. Specifically, we are concerned with:

- * The functional requirements and service characteristics of an enhanced VPN.
- * The design of the enhanced data plane.
- * The necessary control and management protocols in both the underlay and the overlay of the enhanced VPN.
- * The mechanisms to achieve integration between overlay and underlay.
- * The necessary Operation, Administration, and Management (OAM) methods to instrument an enhanced VPN to make sure that the required Service Level Agreement (SLA) between the customer and the network operator is met, and to take any corrective action (such as switching traffic to an alternate path) to avoid SLA violation.

The required layered network structure to achieve this is shown in Section 4.1.

2. Terminology

In this document, the relationship of the four terms "VPN", "VPN+", "VTN", and "Network Slice" are as follows:

- * A Virtual Private Network (VPN) refers to the overlay network service that provides the connectivity between different customer sites, and that maintains traffic separation between different customers. The typical VPN technologies are: IPVPN [RFC2764], L2VPN [RFC4664], L3VPN [RFC4364], and EVPN [RFC7209].
- * An enhanced VPN (VPN+) is an evolution of the VPN service that makes additional service-specific commitments. An enhanced VPN is made by integrating an overlay VPN with a set of network resources allocated in the underlay network.
- * A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VTN has the capability to deliver the performance characteristics required by the VPN+ customers and to provide isolation between different VPN+ services.
- * A network slice could be provided by provisioning an enhanced VPN in the network. Other mechanisms for delivering network slices may exist but are not in scope for this document.

The term "tenant" is used in this document to refer to the customers and all of their associated enhanced VPNs.

The following terms are also used in this document. Some of them are newly defined, some others reference existing definitions.

ACTN: Abstraction and Control of Traffic Engineered Networks [RFC8453]

DetNet: Deterministic Networking. See [DETNET] and [RFC8655]

FlexE: Flexible Ethernet [FLEXE]

TSN: Time Sensitive Networking [TSN]

VN: Virtual Network [I-D.ietf-teas-actn-vn-yang]

VTP: Virtual Transport Path. A VTP is a path through the VTN which provides the required connectivity and performance between two or more customer sites.

3. Overview of the Requirements

This section provides an overview of the requirements of an enhanced VPN service.

3.1. Performance Guarantees

Performance guarantees are made by network operators to their customers in relation to the services provided to the customers. They are usually expressed in SLAs as a set of SLOs.

There are several kinds of performance guarantee, including guaranteed maximum packet loss, guaranteed maximum delay, and guaranteed delay variation. Note that these guarantees apply to conformance traffic, out-of-profile traffic will be handled according to a separate agreement with the customer.

Guaranteed maximum packet loss is usually addressed by setting packet priorities, queue size, and discard policy. However this becomes more difficult when the requirement is combined with latency requirements. The limiting case is zero congestion loss, and that is the goal of DetNet [DETNET] and TSN [TSN]. In modern optical networks, loss due to transmission errors already approaches zero, but there is the possibility of failure of the interface or the fiber itself. This type of fault can only be addressed by some form of signal duplication and transmission over diverse paths.

Guaranteed maximum latency is required by a number of applications particularly real-time control applications and some types of virtual reality applications. DetNet [DETNET] is relevant, however additional methods of enhancing the underlay to better support the delay guarantees may be needed, and these methods will need to be integrated with the overall service provisioning mechanisms.

Guaranteed maximum delay variation is a performance guarantee that may also be needed. [RFC8578] calls up a number of cases that need this guarantee, for example in electrical utilities. Time transfer is an example service that needs a performance guarantee, although it is in the nature of time that the service might be delivered by the underlay as a shared service and not provided through different enhanced VPNs. Alternatively, a dedicated enhanced VPN might be used to provide this as a shared service.

This suggests that a spectrum of service guarantees need to be considered when deploying an enhanced VPN. As a guide to understanding the design requirements we can consider four types of service:

- * Best effort
- * Assured bandwidth
- * Guaranteed latency
- * Enhanced delivery

The best effort service is the basic service as provided by current VPNs.

An assured bandwidth service is one in which the bandwidth over some period of time is assured. This can be achieved either simply based on a best effort service with over-capacity provisioning, or it can be based on MPLS traffic engineered label switching paths (TE-LSPs) with bandwidth reservations. Depending on the technique used, however, the bandwidth is not necessarily assured at any instant. Providing assured bandwidth to VPNs, for example by using per-VPN TE-LSPs, is not widely deployed at least partially due to scalability concerns. VPN+ aims to provide a more scalable approach for such services.

A guaranteed latency service has an upper bound to edge-to-edge latency. Assuring the upper bound is sometimes more important than minimizing latency. There are several new technologies that provide some assistance with this performance guarantee. Firstly, the IEEE TSN project [TSN] introduces the concept of scheduling of delay- and loss-sensitive packets. The DetNet work [DETNET] is also of relevance in assuring an upper bound of end-to-end packet latency. FlexE [FLEXE] is also useful to help provide these guarantees. The use of such underlying technologies to deliver VPN+ services needs to be considered.

An enhanced delivery service is one in which the underlay network (at Layer 3) attempts to deliver the packet through multiple paths in the hope of eliminating packet loss due to equipment or media failures. Such a mechanism may need to be used for VPN+ service.

3.2. Isolation between Enhanced VPN Services

One element of the SLA demanded for an enhanced VPN may be a guarantee that the service offered to the customer will not be affected by any other traffic flows in the network. This is termed "isolation" and a customer may express the requirement for isolation as an SLE [I-D.ietf-teas-ietf-network-slices].

One way for a network operator to meet the requirement for isolation is simply by setting and conforming to all the SLOs. For example, traffic congestion (interference from other services) might impact on the latency experienced by a VPN+ customer. Thus, in this example, conformance to a latency SLO would be the primary requirement for delivery of the VPN+ service, and isolation from other services might be only a means to that end.

Another way for a service provider to meet this SLE is to control the degree to which traffic from one service is isolated from other services in the network.

There is a fine distinction between how isolation is requested by a customer and how it is delivered by the service provider. In general, the customer is interested in service performance and not how it is delivered. Thus, for example, the customer wants specific quality guarantees and is not concerned about how the service provider delivers them. However, it should be noted that some aspects of isolation might be directly measurable by a customer if they have information about the traffic patterns on a number services supported by the same service provider. Furthermore, a customer may be nervous about disruption caused by other services, contamination by other traffic, or delivery of their traffic to the wrong destinations. In this way, the customer may want to specify (and pay for) the level of isolation provided by the service provider.

Isolation is achieved in the realization of a VPN+ through existing technologies that may be supplemented by new mechanisms. The service provider chooses which processes to use to meet this SLE just as they choose how to meet all other SLOs and SLEs. Isolation may be achieved in the network by various forms of resource partitioning ranging from simple separation of service traffic on delivery (ensuring that traffic is not delivered to the wrong customer), through sharing of resources with some form of safeguards, to dedicated allocation of resources for a specific enhanced VPN. For example, interference avoidance may be achieved by network capacity planning, allocating dedicated network resources, traffic policing or shaping, prioritizing in using shared network resources, etc.

The terms hard and soft isolation are used to indicate different levels of isolation. A service has soft isolation if the traffic of one service cannot be received by the customers of another service. The existing IP and MPLS VPNs are examples of services with soft isolation: the network delivers the traffic only to the required customer endpoints. However, with soft isolation, as the network resources are shared, traffic from some services may congest the network, resulting in packet loss and delay for other services. The ability for a service or a group of services to be sheltered from

this effect is called hard isolation. Hard isolation may be needed so that applications with exacting requirements can function correctly, despite other demands (perhaps a burst of traffic in another service) competing for the underlying resources. A customer may request different degrees of isolation ranging from soft isolation to hard isolation. In practice isolation may be delivered on a spectrum between soft and hard, and in some cases soft and hard isolation may be used in a hierarchical manner with one enhanced VPN being built on another.

To provide the required level of isolation, resources may need to be reserved in the data plane of the underlay network and dedicated to traffic from a specific enhanced VPN or a specific group of enhanced VPNs. This may introduce scalability concerns both in the implementation (as each enhanced VPN would need to be tracked in the network) and in how many resources need to be reserved and may be under-used (see Section 4.4). Thus, some trade-off needs to be considered to provide the isolation between enhanced VPNs while still allowing reasonable resource sharing.

An optical underlay can offer a high degree of isolation, at the cost of allocating resources on a long-term and end-to-end basis. On the other hand, where adequate isolation can be achieved at the packet layer, this permits the resources to be shared amongst a group of services and only dedicated to a service on a temporary basis.

The next section explores a pragmatic approach to isolation in packet networks.

3.2.1. A Pragmatic Approach to Isolation

A key question is whether it is possible to achieve hard isolation in packet networks that were designed to provide statistical multiplexing through sharing of data plane resources, a significant economic advantage when compared to a dedicated, or a Time Division Multiplexing (TDM) network. Clearly, there is no need to provide more isolation than is required by the applications, and an approximation to full hard isolation is sufficient in most cases. For example, pseudowires [RFC3985] emulate services that would have had hard isolation in their native form.

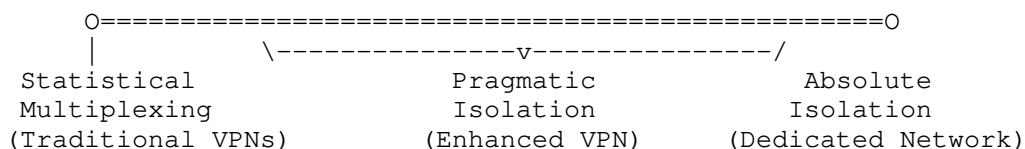


Figure 1: The Spectrum of Isolation

Figure 1 shows a spectrum of isolation that may be delivered by a network. At one end of the spectrum, we see statistical multiplexing technologies that support traditional VPNs. This is a service type that has served the industry well and will continue to do so. At the opposite end of the spectrum, we have the absolute isolation provided by dedicated transport networks. The goal of enhanced VPNs is "pragmatic isolation". This is isolation that is better than what is obtainable from pure statistical multiplexing, more cost effective and flexible than a dedicated network, but is a practical solution that is good enough for the majority of applications. Mechanisms for both soft isolation and hard isolation are needed to meet different levels of service requirement.

3.3. Integration

The way to achieve the characteristics demanded by an enhanced VPN (such as guaranteed or predictable performance) is by integrating the overlay VPN with a particular set of resources in the underlay network which are allocated to meet the service requirement. This needs to be done in a flexible and scalable way so that it can be widely deployed in operators' networks to support a reasonable number of enhanced VPN customers.

Taking mobile networks and in particular 5G into consideration, the integration of the network with service functions is likely a requirement. The IETF's work on service function chaining (SFC) [SFC] provides a foundation for this. Service functions can be considered as part of enhanced VPN services. The detailed mechanisms about the integration between service functions and enhanced VPNs are out of the scope of this document.

3.3.1. Abstraction

Integration of the overlay VPN and the underlay network resources does not always need to be a direct mapping. As described in [RFC7926], abstraction is the process of applying policy to a set of information about a traffic engineered (TE) network to produce selective information that represents the potential ability to connect across the network. The process of abstraction presents the connectivity graph in a way that is independent of the underlying network technologies, capabilities, and topology so that the graph can be used to plan and deliver network services in a uniform way.

Virtual networks can be built on top of an abstracted topology that represents the connectivity capabilities of the underlay TE based network as described in the framework for Abstraction and Control of TE Networks (ACTN) [RFC8453] as discussed further in Section 5.5. [I-D.ietf-teas-applicability-actn-slicing] describes the applicability of ACTN to network slicing and is, therefore, relevant to the consideration of using ACTN to enable enhanced VPNs.

3.4. Dynamic Changes

Enhanced VPNs need to be created, modified, and removed from the network according to service demands. An enhanced VPN that requires hard isolation (Section 3.2) must not be disrupted by the instantiation or modification of another enhanced VPN. Determining whether modification of an enhanced VPN can be disruptive to that VPN, and whether the traffic in flight will be disrupted can be a difficult problem.

The data plane aspects of this problem are discussed further in Section 5.1, Section 5.2, and Section 5.3.

The control plane aspects of this problem are discussed further in Section 5.4.

The management plane aspects of this problem are discussed further in Section 5.5.

Dynamic changes both to the enhanced VPN and to the underlay network need to be managed to avoid disruption to services that are sensitive to changes in network performance.

In addition to non-disruptively managing the network during changes such as the inclusion of a new VPN endpoint or a change to a link, VPN traffic might need to be moved because of changes to traffic patterns and volumes.

3.5. Customized Control

In many cases the customers are delivered with enhanced VPN services without knowing the information about the underlying VTNs. However, depends on the agreement between the operator and the customer, in some cases the customer may also be provided with some information about the underlying VTNs. Such information can be filtered or aggregated according to the operator's policy. This allows the customer of the enhanced VPN to have some visibility and even control over how the underlying topology and resources of the VTN are used. For example, the customers may be able to specify the service paths within the VTN for specific traffic flows of their enhanced VPNs.

Depending on the requirements, an enhanced VPN customer may have his own network controller, which may be provided with an interface to the control or management system run by the network operator. Note that such control is within the scope of the customer's enhanced VPN, any additional changes beyond this would require some intervention by the network operator.

A description of the control plane aspects of this problem are discussed further in Section 5.4. A description of the management plane aspects of this feature can be found in Section 5.5.

3.6. Applicability to Overlay Technologies

The concept of enhanced VPN can be applied to any existing and future multi-tenancy overlay technologies including but not limited to :

- * Layer-2 point-to-point services such as pseudowires [RFC3985]
- * Layer-2 VPNs [RFC4664]
- * Ethernet VPNs [RFC7209]
- * Layer-3 VPNs [RFC4364], [RFC2764]

Where such VPN service types need enhanced isolation and delivery characteristics, the technologies described in Section 5 can be used to provide an underlay with the required enhanced performance.

3.7. Inter-Domain and Inter-Layer Network

In some scenarios, an enhanced VPN service may span multiple network domains. A domain is considered to be any collection of network elements within a common realm of address space or path computation responsibility [RFC5151] for example, an Autonomous System. In some domains the network operator may manage a multi-layered network, for example, a packet network over an optical network. When VPN+ services are provisioned in such network scenarios, the technologies used in different network planes (data plane, control plane, and management plane) need to provide mechanisms to support multi-domain and multi-layer coordination and integration, so as to provide the required service characteristics for different enhanced VPNs, and improve network efficiency and operational simplicity.

4. Architecture of Enhanced VPNs

A number of VPN+ services will typically be provided by a common network infrastructure. Each VPN+ service is provisioned with an overlay VPN and a corresponding VTN, which has a specific set of network resources and functions allocated in the underlay to satisfy the needs of the customer. One VTN may support one or more VPN+ services. The integration between the overlay connectivity and the underlay resources ensures the required isolation between different VPN+ services, and achieves the guaranteed performance for different customers.

The VPN+ architecture needs to be designed with consideration given to:

- * An enhanced data plane.
- * A control plane to create enhanced VPNs, making use of the data plane isolation and performance guarantee techniques.
- * A management plane for enhanced VPN service life-cycle management.

These topics are expanded below.

- * The enhanced data plane:
 - Provides the required packet latency and jitter characteristics.
 - Provides the required packet loss characteristics.
 - Provides the required resource isolation capability, e.g., bandwidth guarantee.
 - Provides the mechanism to associate a packet with the set of resources allocated to a VTN which the VPN+ service packet is mapped to.
- * The control plane:
 - Collects information about the underlying network topology and network resources, and exports this to network nodes and/or a centralized controller as required.
 - Creates VTNs with the network resource and topology properties needed by the VPN+ services.

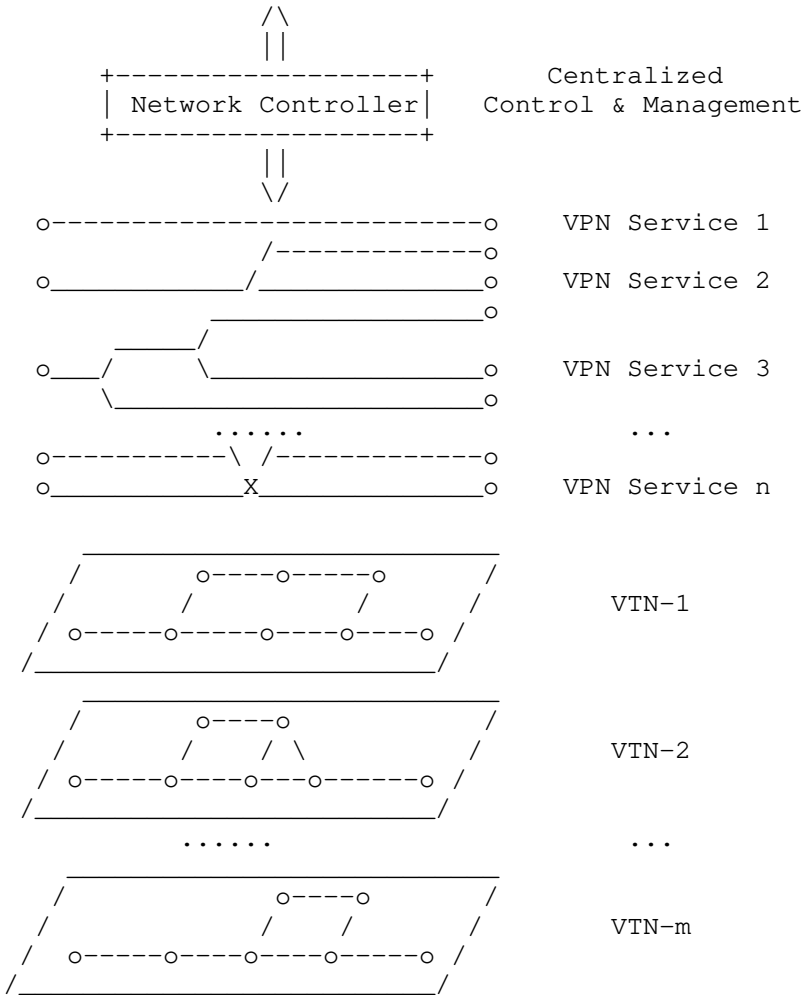
- Distribute the attributes of VTNs to network nodes which participate in the VTNs and/or a centralized controller.
 - Compute and set up network paths in each VTN.
 - Map VPN+ services to an appropriate VTN.
 - Determines the risk of SLA violation and takes appropriate avoiding action.
 - Consider the right balance of per-packet and per-node state according to the needs of the VPN+ services to scale to the required size.
- * The management plane:
- Provides an interface between the VPN+ service provider (e.g., operator's network management system) and the VPN+ customer (e.g., an organization or a service with enhanced VPN requirement) such that some of the operation requests can be met without interfering with other VPN+ customers.
 - Provides an interface between the VPN+ service provider and the VPN+ customers to expose the network capability information toward the customer.
 - Provides the service life-cycle management and operation of VPN+ services (e.g., creation, modification, assurance/monitoring, and decommissioning).
- * Operations, Administration, and Maintenance (OAM)
- Provides the tools to verify the connectivity and performance of the VPN+ service.
 - Provides the tools to verify whether the underlay network resources are correctly allocated and operating properly.
- * Telemetry
- Provides the mechanisms to collect network information about the operation of the data plane, control plane, and management plane. More specifically, telemetry provides the mechanisms to collect network data:
 - o from the underlay network for overall performance evaluation and for the planning VPN+ services.

- o from each VPN+ service for monitoring and analytics of the characteristics and SLA fulfillment of the VPN+ services.

4.1. Layered Architecture

The layered architecture of VPN+ is shown in Figure 2.

Underpinning everything is the physical network infrastructure layer which provide the underlying resources used to provision the separated VTNs. This layer is responsbile for the partitioning of link and/or node resources for different VTNs. Each subset of link or node resource can be considered as a virtual link or virtual node used to build the VTNs.



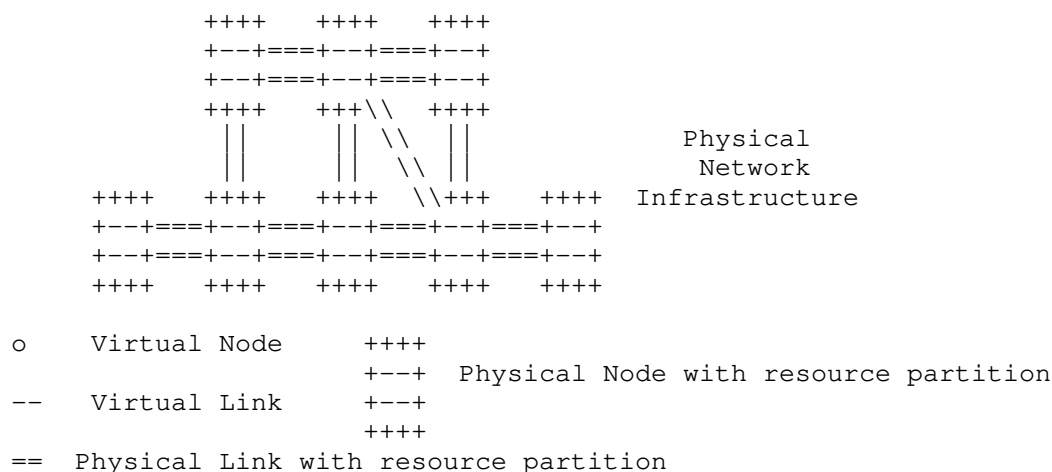


Figure 2: The Layered Architecture of VPN+

Various components and techniques discussed in Section 5 can be used to enable resource partitioning, such as FlexE, TSN, DetNet, dedicated queues, etc. These partitions may be physical or virtual so long as the SLA required by the higher layers is met.

Based on the network resource partitions provided by the physical network infrastructure, multiple VTNs can be created, each with a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology, so as to meet the requirements of different VPN+ services or different groups of VPN+ services. According to the associated logical network topology, each VTN needs to be instantiated on a set of network nodes and links which are involved in the logical topology. And on each node or link, each VTN is associated with a set of local resources which are allocated for the processing of traffic in the VTN. The VTN provides the integration between the virtual network topology and the required underlying network resources.

According to the service requirements on connectivity, performance and isolation, etc., VPN services can be mapped to the appropriate VTNs in the network. Different VPN services can be mapped to different VTNs, while it is also possible that multiple VPNs are mapped to the same VTN. Thus VTN is an essential scaling technique, as it has the potential of eliminating per-path state from the network. In addition, when a group of VPN+ services are mapped to a single VTN, only the network state of the single VTN needs to be maintained in the network (see Section 4.4 for more information).

The centralized network controller is responsible for creating a VTN, instructing the involved network nodes to allocate network resources to the VTN, and provisioning the VPN services on the VTN. A distributed control plane may be used for distributing the VTN resource and topology attributes among nodes in the VTN.

The process used to create VTNs and to allocate network resources for use by the VTNs needs to take a holistic view of the needs of all of its customers and to partition the resources accordingly. However, within a VTN these resources can, if required, be managed via a dynamic control plane. This provides the required scalability and isolation with some flexibility.

4.2. Multi-Point to Multi-Point (MP2MP) Connectivity

At the VPN service level, the required connectivity for an MP2MP VPN service is usually full or partial mesh. To support such VPN services, the corresponding VTN also needs to provide MP2MP connectivity among the end points.

Other service requirements may be expressed at different granularities, some of which can be applicable to the whole service, while some others may only be applicable to some pairs of end points. For example, when a particular level of performance guarantee is required, the point-to-point path through the underlying VTN of the VPN+ service may need to be specifically engineered to meet the required performance guarantee.

4.3. Application Specific Data Types

Although a lot of the traffic that will be carried over VPN+ will likely be IP based, the design must be capable of carrying other traffic types, in particular Ethernet traffic. This is easily accomplished through the various pseudowire (PW) techniques [RFC3985]. Where the underlay is MPLS, Ethernet traffic can be carried over VPN+ encapsulated according to the method specified in [RFC4448]. Where the underlay is IP, Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] can be used with Ethernet traffic carried according to [RFC4719]. Encapsulations have been defined for most of the common Layer-2 types for both PW over MPLS and for L2TPv3.

4.4. Scaling Considerations

VPNs are instantiated as overlays on top of an operator's network and offered as services to the operator's customers. An important feature of overlays is that they can deliver services without placing per-service state in the core of the underlay network.

VPN+ may need to install some additional state within the network to achieve the features that they require. Solutions must consider minimizing and controlling the scale of such state, and deployment architectures should constrain the number of VPN+ services so that the additional state introduced to the network is acceptable and under control. It is expected that the number of VPN+ services will be small at the beginning, and even in future the number of VPN+ services will be fewer than traditional VPNs because pre-existing VPN techniques are good enough to meet the needs of most existing VPN-type services.

In general, it is not required that the state in the network be maintained in a 1:1 relationship with the VPN+ services. It will usually be possible to aggregate a set or group of VPN+ services so that they share the same VTN and the same set of network resources (much in the same way that current VPNs are aggregated over transport tunnels) so that collections of VPN+ services that require the same behavior from the network in terms of resource reservation, latency bounds, resiliency, etc. can be grouped together. This is an important feature to assist with the scaling characteristics of VPN+ deployments.

[I-D.dong-teas-nrp-scalability] provides more details of scalability considerations for the network resource partitions used to instantiate VTNs, and Section 7 includes a greater discussion of scalability considerations.

5. Candidate Technologies

A VPN is a network created by applying a demultiplexing technique to the underlying network (the underlay) to distinguish the traffic of one VPN from that of another. A VPN path that travels by other than the shortest path through the underlay normally requires state in the underlay to specify that path. State is normally applied to the underlay through the use of the RSVP-TE signaling protocol, or directly through the use of an SDN controller, although other techniques may emerge as this problem is studied. This state gets harder to manage as the number of VPN paths increases. Furthermore, as we increase the coupling between the underlay and the overlay to support the VPN+ service, this state will increase further. Thus, a VPN+ solution needs tighter coupling with the underlay than is the case with existing VPN techniques. We cannot, for example, share the network resource between VPN+ services which require hard isolation.

In a VPN+ solution, different subsets of the underlay resources can be dedicated to different VPN+ services or different groups of VPN+ services through the use of VTNs.

5.1. Packet Forwarding Plane Technologies

Several candidate Layer 2 packet- or frame-based data plane solutions which provide the required isolation and guarantees are described in the following sections.

5.1.1. Flexible Ethernet

FlexE [FLEXE] provides the ability to multiplex channels over an Ethernet link to create point-to-point fixed- bandwidth connections in a way that provides hard isolation. FlexE also supports bonding links to create larger links out of multiple low capacity links.

However, FlexE is only a link level technology. When packets are received by the downstream node, they need to be processed in a way that preserves that isolation in the downstream node. This in turn requires a queuing and forwarding implementation that preserves the end-to-end isolation.

If different FlexE channels are used for different services, then no sharing is possible between the FlexE channels. This means that it may be difficult to dynamically redistribute unused bandwidth to lower priority services in another FlexE channel. If one FlexE channel is used by one customer, the customer can use some methods to manage the relative priority of their own traffic in the FlexE channel.

5.1.2. Dedicated Queues

DiffServ based queuing systems are described in [RFC2475] and [RFC4594]. This approach is not sufficient to provide isolation for VPN+ services because DiffServ does not provide enough markers to differentiate between traffic of a large number of VPN+ services. Nor does DiffServ offer the range of service classes that each VPN+ service needs to provide to its tenants. This problem is particularly acute with an MPLS underlay, because MPLS only provides eight traffic classes.

In addition, DiffServ, as currently implemented, mainly provides per-hop priority-based scheduling, and it is difficult to use it to achieve quantitative resource reservation for different VPN+ services.

To address these problems and to reduce the potential interference between VPN+ services, it would be necessary to steer traffic to dedicated input and output queues per VPN+ service or per group of VPN+ services: some routers have a large number of queues and sophisticated queuing systems which could support this, while some routers may struggle to provide the granularity and level of isolation required by the applications of VPN+.

5.1.3. Time Sensitive Networking

Time Sensitive Networking (TSN) [TSN] is an IEEE project to provide a method of carrying time sensitive information over Ethernet. It introduces the concept of packet scheduling where a packet stream may be given a time slot guaranteeing that it experiences no queuing delay or increase in latency beyond the very small scheduling delay. The mechanisms defined in TSN can be used to meet the requirements of time sensitive traffic flows of VPN+ service.

Ethernet can be emulated over a Layer 3 network using an IP or MPLS pseudowire. However, a TSN Ethernet payload would be opaque to the underlay and thus not treated specifically as time sensitive data. The preferred method of carrying TSN over a Layer 3 network is through the use of deterministic networking as explained in Section 5.2.1.

5.2. Layer Three Data Plane

This section considers the problem of VPN+ service differentiation and the representation of underlying network resources in the network layer. More specifically, it describes the possible data plane mechanisms to determine the network resources and the logical network topology or paths associated with a VTN.

5.2.1. Deterministic Networking

Deterministic Networking (DetNet) [RFC8655] is a technique being developed in the IETF to enhance the ability of Layer-3 networks to deliver packets more reliably and with greater control over the delay. The design cannot use re-transmission techniques such as TCP since that can exceed the delay tolerated by the applications. Even the delay improvements that are achieved with Stream Control Transmission Protocol Partial Reliability Extension (SCTP-PR) [RFC3758] may not meet the bounds set by application demands. DetNet pre-emptively sends copies of the packet over various paths to minimize the chance of all copies of a packet being lost. It also seeks to set an upper bound on latency, but the goal is not to minimize latency. Detnet can be realized over IP data plane [RFC8939] or MPLS data plane [RFC8964], and may be used to provide

Virtual Transport Path (VTP) for VPN+ services.

5.2.2. MPLS Traffic Engineering (MPLS-TE)

MPLS-TE [RFC2702][RFC3209] introduces the concept of reserving end-to-end bandwidth for a TE-LSP, which can be used to provide a point-to-point Virtual Transport Path (VTP) across the underlay network to support VPN services. VPN traffic can be carried over dedicated TE-LSPs to provide reserved bandwidth for each specific connection in a VPN, and VPNs with similar behavior requirements may be multiplexed onto the same TE-LSPs. Some network operators have concerns about the scalability and management overhead of MPLS-TE system, especially with regard to those systems that use an active control plane, and this has lead them to consider other solutions for traffic engineering in their networks.

5.2.3. Segment Routing

Segment Routing (SR) [RFC8402] is a method that prepends instructions to packets at the head-end of a path. These instructions are used to specify the nodes and links to be traversed, and allow the packets to be routed on paths other than the shortest path. By encoding the state in the packet, per-path state is transitioned out of the network.

An SR traffic engineered path operates with a granularity of a link. Hints about priority are provided using the Traffic Class (TC) or Differentiated Services Code Point (DSCP) field in the packet header. However, to achieve the performance and isolation characteristics that are sought by VPN+ customers, it will be necessary to steer packets through specific virtual links and/or queues on the same link and direct them to use specific resources. With SR, it is possible to introduce such fine-grained packet steering by specifying the queues and the associated resources through an SR instruction list.

Note that the concept of a queue is a useful abstraction for different types of underlay mechanism that may be used to provide enhanced isolation and performance support. How the queue satisfies the requirement is implementation specific and is transparent to the layer-3 data plane and control plane mechanisms used.

With Segment Routing, the SR instruction list could be used to build a P2P path, and a group of SR SIDs could also be used to represent an MP2MP network. Thus, the SR based mechanism could be used to provide both a Virtual Transport Path (VTP) and a Virtual Transport Network (VTN) for VPN+ services.

5.3. Non-Packet Data Plane

Non-packet underlay data plane technologies often have TE properties and behaviors, and meet many of the key requirements in particular for bandwidth guarantees, traffic isolation (with physical isolation often being an integral part of the technology), highly predictable latency and jitter characteristics, measurable loss characteristics, and ease of identification of flows. The cost is that the resources are allocated on a long-term and end-to-end basis. Such an arrangement means that the full cost of the resources has to be borne by the service that is allocated with the resources.

5.4. Control Plane

The control plane of VPN+ would likely be based on a hybrid control mechanism that takes advantage of a logically centralized controller for on-demand provisioning and global optimization, whilst still relying on a distributed control plane to provide scalability, high reliability, fast reaction, automatic failure recovery, etc. Extension to and optimization of the centralized and distributed control plane is needed to support the enhanced properties of VPN+.

As described in section 4, the VPN+ control plane needs to provide the following functions:

- * Collects information about the underlying network topology and network resources, and exports this to network nodes and/or a centralized controller as required.
- * Creates VTNs with the network resource and topology properties needed by the VPN+ services.
- * Distribute the attributes of VTNs to network nodes which participate in the VTNs and/or the centralized controller.
- * Compute and set up VTPs in each VTN.
- * Map VPN+ services to an appropriate VTN.

The collection of underlying network topology and resource information can be done using existing the IGP and BGP-LS based mechanisms. The creation of VTN and the distribution of VTN attributes may need further control protocol extensions. The computation of VTPs based on the attributes and constraints of the VTN can be performed either by the headend node of the path or a centralized Path Computation Element (PCE).

There are two candidate mechanisms for the setup of VTPs in the VTN: RSVP-TE and Segment Routing (SR).

- * RSVP-TE [RFC3209] provides the signaling mechanism for establishing a TE-LSP in an MPLS network with end-to-end resource reservation. This can be seen as an approach of providing a Virtual Transport Path (VTP) which could be used to bind the VPN to specific network resources allocated within the underlay, but there remain scalability concerns as mentioned in Section 5.2.2.
- * The SR control plane [RFC8665] [RFC8667] [RFC9085] does not have the capability of signaling resource reservations along the path. On the other hand, the SR approach provides a potential way of binding the underlay network resource and the VTNs without requiring per-path state to be maintained in the network. A centralized controller can perform resource planning and reservation for VTNs, and it needs to instruct the network nodes to ensure that resources are correctly allocated for the VTN. The controller could provision the SR paths based on the mechanism in [I-D.ietf-spring-segment-routing-policy] to the headend nodes of the paths.

According to the service requirements on connectivity, performance and isolation, one VPN+ service may be mapped a dedicated VTN, or a group of VPN+ services may be mapped to the same VTN. The mapping of VPN+ services to VTN can be achieved using existing control mechanisms with possible extensions, and it can be based on either the characteristics of the data packet or the attributes of the VPN service routes.

5.5. Management Plane

The management plane provides the interface between the VPN+ service provider and the customers for life-cycle management of the VPN+ service (i.e., creation, modification, assurance/monitoring, and decommissioning). It relies on a set of service data models for the description of the information and operations needed on the interface.

As an example, in the context of 5G end-to-end network slicing [TS28530], the management of VPN+ services is considered as the management of the transport network segment of the 5G end-to-end network slice. The 3GPP management system may provide the connectivity and performance related parameters as requirements to the management plane of the transport network. It may also require the transport network to expose the capabilities and status of the network slice. Thus, an interface between the VPN+ management plane and the 5G network slice management system, and relevant service data models are needed for the coordination of 5G end-to-end network slice management.

The management plane interface and data models for VPN+ services can be based on the service models described in Section 5.6.

It is important that the management life-cycle supports in-place modification of VPN+ services. That is, it should be possible to add and remove end points, as well as to change the requested characteristics of the service that is delivered. The management system needs to be able to assess the revised VPN+ requests and determine whether they can be provided by the existing VTNs or whether changes must be made, and it will additionally need to determine whether those changes to the VTN are possible. If not, then the customer's modification request may be rejected.

When the modification of a VPN+ service is possible, the management system should make every effort to make the changes in a non-disruptive way. That is, the modification of the VPN+ service or the underlying VTN should not perturbate traffic on the VPN+ service in a way that causes the service level to drop below the agreed levels. Furthermore, in the spirit of isolation, changes to one VPN+ service should not cause disruption to other VPN+ services.

The network operator for the underlay network (i.e., the provider of the VPN+ service) may delegate some operational aspects of the overlay VPN and the underlying VTN to the customer. In this way, the VPN+ is presented to the customer as a virtual network, and the customer can choose how to use that network. The customer cannot exceed the capabilities of the virtual links and nodes, but can decide how to load traffic onto the network, for example, by assigning different metrics to the virtual links so that the customer can control how traffic is routed through the virtual network. This approach requires a management system for the virtual network, but does not necessarily require any coordination between the management systems of the virtual network and the physical network, except that the virtual network management system might notice when the VTN is close to capacity or considerably under-used and automatically request changes in the service provided by the underlay network.

5.6. Applicability of Service Data Models to Enhanced VPN

This section describes the applicability of the existing and in-progress service data models to VPN+. [RFC8309] describes the the scope and purpose of service models and shows where a service model might fit into a SDN based network management architecture. New service models may also be introduced for some of the required management functions.

Service data models are used to represent, monitor, and manage the virtual networks and services enabled by VPN+. The VPN customer service models (e.g., the layer 3 VPN service model (L3SM) [RFC8299], the layer 2 VPN service model (L2SM) [RFC8466]), or the ACTN Virtual Network (VN) model [I-D.ietf-teas-actn-vn-yang]) are service models which can provide the customer's view of the VPN+ service. The layer 3 VPN network model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm], the layer 2 VPN network model (L2NM) [I-D.ietf-opsawg-l2nm] provide the operator's view of the managed infrastructure as a set of virtual networks and the associated resources. The NRP model [I-D.wd-teas-nrp-yang] further provides the management of the virtual underlay network topology and resources both in the controller and in the network devices to instantiate the VTNs needed for the VPN+ services.

The ACTN framework[RFC8453] supports operators in viewing and controlling different domains and presenting virtualized networks to their customers. [I-D.ietf-teas-applicability-actn-slicing] discusses the applicability of the ACTN approach in the context of network slicing. Since there is a strong correlation between network slices and enhanced VPNs, that document also give guidance on how ACTN can be applied to enhanced VPNs.

6. Applicability to Network Slice Realization

One of the typical use cases of enhanced VPN is to deliver IETF network slice service. This section describes the applicability of enhanced VPN to network slice realization.

In order to provide IETF network slices to customers, a technology-agnostic network slice service Northbound Interface (NBI) data model [I-D.ietf-teas-ietf-network-slice-nbi-yang] is needed for the customers to communicate the requirements of IETF network slices (end points, connectivity, SLOs, and SLEs). These requirements may be realized using technology specified in this document to instruct the network to instantiate a VPN+ service to meet the requirements of the IETF network slice customers.

6.1. VTN Planning

According to the network operators' network resource planning policy, or based on the requirement of one or a group of customers or services, a VTN may need to be created. One of the basic requirements for a VTN is to provide a set of dedicated network resources to avoid unexpected interference from other services in the same network. Other possible requirements may include the required topology and connectivity, bandwidth, latency, reliability, etc.

A centralized network controller can be responsible for calculating a subset of the underlay network topology (which is called a logical topology) to support the VTN requirement. And on the network nodes and links within the logical topology, the set of network resources to be allocated to the VTN can also be determined by the controller. Normally such calculation needs to take the underlay network connectivity information and the available network resource information of the underlay network into consideration. The network controller may also take the status of the existing VTNs into consideration in the planning and calculation of a new VTN.

6.2. VTN Instantiation

According to the result of the VTN planning, the network nodes and links involved in the logical topology of the VTN are instructed to allocate the required set of network resources for the VTN. One or multiple mechanisms as specified in section 5.1 can be used to partition the forwarding plane network resources and allocate different subsets of resources to different VTNs. In addition, the data plane identifiers which are used to identify the set of network resources allocated to the VTN are also provisioned on the network nodes. Depends on the data plane technologies used, the set of network resources of a VTN can be identified using either resource aware SR segments as specified in [I-D.ietf-spring-resource-aware-segments], or a dedicated VTN resource ID as specified in [I-D.dong-6man-enhanced-vpn-vtn-id] can be introduced. The network nodes involved in a VTN may distribute the logical topology information, the VTN specific network resource information and the VTN resource identifiers using the control plane. Such information could be used by the controller and the network nodes to compute the TE or shortest paths within the VTN, and install the VTN specific forwarding entries to network nodes.

6.3. VPN+ Service Provisioning

According to the connectivity requirements of an IETF network slice service, an overlay VPN can be created using the existing or future multi-tenancy overlay technologies as described in Section 3.6.

Then according to the SLOs and SLEs requirements of the network slice, the overlay VPN is mapped to an appropriate VTN as the virtual underlay. The integration of the overlay VPN and the underlay VTN together provide an enhanced VPN service which can meet the network slice service requirements.

6.4. Network Slice Traffic Steering and Forwarding

At the edge of the operator's network, traffic of IETF network slices can be classified based on the rules defined by operator's policy, so that the traffic is treated as a specific VPN+ service, which is further mapped to a underlay VTN. Packets belonging to the VPN+ service will be processed and forwarded by network nodes based the TE or shortest path forwarding entries and the set of network resources of the corresponding VTN.

7. Scalability Considerations

VPN+ provides performance guaranteed services in packet networks, but with the potential cost of introducing additional state into the network. There are at least three ways that this additional state might be brought into the network:

- * Introduce the complete state into the packet, as is done in SR. This allows the controller to specify the detailed series of forwarding and processing instructions for the packet as it transits the network. The cost of this is an increase in the packet header size. The cost is also that systems will have capabilities enabled in case they are called upon by a service. This is a type of latent state, and increases as the path and resources that need to be exclusively available to a VPN are specified more precisely.
- * Introduce the state to the network. This is normally done by creating a path using signaling such as RSVP-TE. This could be extended to include any element that needs to be specified along the path, for example explicitly specifying queuing policy. It is also possible to use other methods to introduce path state, such as via an SDN controller, or possibly by modifying a routing protocol. With this approach there is state per path: per-path characteristic that needs to be maintained over the life of the path. This is more network state than is needed using SR, but the packets are usually shorter.
- * Provide a hybrid approach. One example is based on using binding SIDs [RFC8402] to create path fragments, and bind them together with SR. Dynamic creation of a VPN service path using SR requires less state maintenance in the network core at the expense of

larger packet headers. The packet size can be lower if a form of loose source routing is used (using a few nodal SIDs), and it will be lower if no specific functions or resources on the routers are specified.

Reducing the state in the network is important to VPN+, as it requires the overlay to be more closely integrated with the underlay than with traditional VPNs. This tighter coupling would normally mean that more state needs to be created and maintained in the network, as the state about fine granularity processing would need to be loaded and maintained in the routers. However, an SR approach allows much of this state to be spread amongst the network ingress nodes, and transiently carried in the packets as SIDs.

Further discussion of the scalability considerations of the underlaying network resource partitions of VPN+ can be found in [I-D.dong-teas-nrp-scalability].

7.1. Maximum Stack Depth of SR

One of the challenges with SR is the stack depth that nodes are able to impose on packets [RFC8491]. This leads to a difficult balance between adding state to the network and minimizing stack depth, or minimizing state and increasing the stack depth.

7.2. RSVP-TE Scalability

The traditional method of creating a resource allocated path through an MPLS network is to use the RSVP-TE protocol. However, there have been concerns that this requires significant continuous state maintenance in the network. Work to improve the scalability of RSVP-TE LSPs in the control plane can be found in [RFC8370].

There is also concern at the scalability of the forwarder footprint of RSVP-TE as the number of paths through a label switching router (LSR) grows. [RFC8577] addresses this by employing SR within a tunnel established by RSVP-TE.

7.3. SDN Scaling

The centralized approach of SDN requires state to be stored in the network, but does not have the overhead of also requiring control plane state to be maintained. Each individual network node may need to maintain a communication channel with the SDN controller, but that compares favorably with the need for a control plane to maintain communication with all neighbors.

However, SDN may transfer some of the scalability concerns from the network to the centralized controller. In particular, there may be a heavy processing burden at the controller, and a heavy load in the network surrounding the controller. A centralized controller also presents a single point of failure within the network.

8. OAM Considerations

The design of OAM for VPN+ services needs to consider the following requirements:

- * Instrumentation of the underlay so that the network operator can be sure that the resources committed to a tenant are operating correctly and delivering the required performance.
- * Instrumentation of the overlay by the tenant. This is likely to be transparent to the network operator and to use existing methods. Particular consideration needs to be given to the need to verify the isolation and the various committed performance characteristics.
- * Instrumentation of the overlay by the network provider to proactively demonstrate that the committed performance is being delivered. This needs to be done in a non-intrusive manner, particularly when the tenant is deploying a performance sensitive application.
- * Verification of the conformity of the path to the service requirement. This may need to be done as part of a commissioning test.

A study of OAM in SR networks has been documented in [RFC8403].

9. Telemetry Considerations

Network visibility is essential for network operation. Network telemetry has been considered as an ideal means to gain sufficient network visibility with better flexibility, scalability, accuracy, coverage, and performance than conventional OAM technologies.

As defined in [I-D.ietf-opsawg-ntf], the objective of Network Telemetry is to acquire network data remotely for network monitoring and operation. It is a general term for a large set of network visibility techniques and protocols. Network telemetry addresses the current network operation issues and enables smooth evolution toward intent-driven autonomous networks. Telemetry can be applied on the forwarding plane, the control plane, and the management plane in a network.

How the telemetry mechanisms could be used or extended for the VPN+ service is out of the scope of this document.

10. Enhanced Resiliency

Each VPN+ service has a life cycle, and may need modification during deployment as the needs of its tenant change. This is discussed in Section 5.5. Additionally, as the network evolves, there may need to be garbage collection performed to consolidate resources into usable quanta.

Systems in which the path is imposed, such as SR or some form of explicit routing, tend to do well in these applications, because it is possible to perform an atomic transition from one path to another. That is, a single action by the head-end that changes the path without the need for coordinated action by the routers along the path. However, implementations and the monitoring protocols need to make sure that the new path is operational and meets the required SLA before traffic is transitioned to it. It is possible for deadlocks to arise as a result of the network becoming fragmented over time, such that it is impossible to create a new path or to modify an existing path without impacting the SLA of other paths. Resolution of this situation is as much a commercial issue as it is a technical issue and is outside the scope of this document.

There are, however, two manifestations of the latency problem that are for further study in any of these approaches:

- * The problem of packets overtaking one another if a path latency reduces during a transition.
- * The problem of transient variation in latency in either direction as a path migrates.

There is also the matter of what happens during failure in the underlay infrastructure. Fast reroute is one approach, but that still produces a transient loss with a normal goal of rectifying this within 50ms [RFC5654]. An alternative is some form of N+1 delivery such as has been used for many years to support protection from service disruption. This may be taken to a different level using the techniques of DetNet with multiple in-network replication and the culling of later packets [RFC8655].

In addition to the approach used to protect high priority packets, consideration should be given to the impact of best effort traffic on the high priority packets during a transition. Specifically, if a conventional re-convergence process is used there will inevitably be micro-loops and whilst some form of explicit routing will protect the

high priority traffic, lower priority traffic on best effort shortest paths will micro-loop without the use of a loop prevention technology. To provide the highest quality of service to high priority traffic, either this traffic must be shielded from the micro-loops, or micro-loops must be prevented completely.

11. Operational Considerations

It is likely that VPN+ services will be introduced in networks which already have traditional VPN services deployed. Depending on service requirements, the tenants or the operator may choose to use a traditional VPN or an enhanced VPN to fulfill a service requirement. The information and parameters to assist such a decision needs to be reflected on the management interface between the tenant and the operator.

12. Security Considerations

All types of virtual network require special consideration to be given to the isolation of traffic belonging to different tenants. That is, traffic belonging to one VPN must not be delivered to end points outside that VPN. In this regard VPN+ neither introduce, nor experience a greater security risks than other VPNs.

However, in a VPN+ service the additional service requirements need to be considered. For example, if a service requires a specific upper bound to latency then it can be damaged by simply delaying the packets through the activities of another tenant, i.e., by introducing bursts of traffic for other services. In some respects this makes the enhanced VPN more susceptible to attacks since the SLA may be broken. But another view is that the operator must, in any case, perform monitoring of the enhanced VPN to ensure that the SLA is met, and this means that the operator may be more likely to spot the early onset of a security attack and be able to take pre-emptive protective action.

The measures to address these dynamic security risks must be specified as part to the specific solution are form part of the isolation requirements of a service.

While a VPN+ service may be sold as offering encryption and other security features as part of the service, customers would be well advised to take responsibility for their own security requirements themselves possibly by encrypting traffic before handing it off to the service provider.

The privacy of VPN+ service customers must be preserved. It should not be possible for one customer to discover the existence of another customer, nor should the sites that are members of an VPN+ be externally visible.

13. IANA Considerations

There are no requested IANA actions.

14. Contributors

Daniel King
Email: daniel@olddog.co.uk

Adrian Farrel
Email: adrian@olddog.co.uk

Jeff Tansura
Email: jefftant.ietf@gmail.com

Zhenbin Li
Email: lizhenbin@huawei.com

Qin Wu
Email: bill.wu@huawei.com

Bo Wu
Email: lana.wubo@huawei.com

Daniele Ceccarelli
Email: daniele.ceccarelli@ericsson.com

Mohamed Boucadair
Email: mohamed.boucadair@orange.com

Sergio Belotti
Email: sergio.belotti@nokia.com

Haomian Zheng
Email: zhenghaomian@huawei.com

15. Acknowledgements

The authors would like to thank Charlie Perkins, James N Guichard, John E Drake, Shunsuke Homma, and Luis M. Contreras for their review and valuable comments.

This work was supported in part by the European Commission funded H2020-ICT-2016-2 METRO-HAUL project (G.A. 761727).

16. References

16.1. Normative References

- [RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, DOI 10.17487/RFC2764, February 2000, <<https://www.rfc-editor.org/info/rfc2764>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.

16.2. Informative References

- [DETNET] "Deterministic Networking", March , <<https://datatracker.ietf.org/wg/detnet/about/>>.
- [FLEXE] "Flex Ethernet Implementation Agreement", March 2016, <<http://www.oiforum.com/wp-content/uploads/OIF-FLEXE-01.0.pdf>>.
- [I-D.dong-6man-enhanced-vpn-vtn-id] Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra, "Carrying Virtual Transport Network (VTN) Identifier in IPv6 Extension Header", Work in Progress, Internet-Draft, draft-dong-6man-enhanced-vpn-vtn-id-06, 24 October 2021, <<https://www.ietf.org/archive/id/draft-dong-6man-enhanced-vpn-vtn-id-06.txt>>.
- [I-D.dong-teas-nrp-scalability] Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N., Mishra, G., Qin, F., Saad, T., and V. P. Beeram, "Scalability Considerations for Network Resource Partition", Work in Progress, Internet-Draft, draft-dong-teas-nrp-scalability-01, 7 February 2022, <<https://www.ietf.org/archive/id/draft-dong-teas-nrp-scalability-01.txt>>.

[I-D.ietf-opsawg-l2nm]

Barguil, S., Dios, O. G. D., Boucadair, M., and L. A. Munoz, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-12, 22 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-l2nm-12.txt>>.

[I-D.ietf-opsawg-l3sm-l3nm]

Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", Work in Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-18, 8 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-l3sm-l3nm-18.txt>>.

[I-D.ietf-opsawg-ntf]

Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", Work in Progress, Internet-Draft, draft-ietf-opsawg-ntf-13, 3 December 2021, <<https://www.ietf.org/archive/id/draft-ietf-opsawg-ntf-13.txt>>.

[I-D.ietf-spring-resource-aware-segments]

Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", Work in Progress, Internet-Draft, draft-ietf-spring-resource-aware-segments-03, 12 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-spring-resource-aware-segments-03.txt>>.

[I-D.ietf-spring-segment-routing-policy]

Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", Work in Progress, Internet-Draft, draft-ietf-spring-segment-routing-policy-18, 17 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-18.txt>>.

[I-D.ietf-teas-actn-vn-yang]

Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-13, 23 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-actn-vn-yang-13.txt>>.

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

Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B. Y., and S. Belotti, "Applicability of YANG models for Abstraction and

Control of Traffic Engineered Networks", Work in Progress, Internet-Draft, draft-ietf-teas-actn-yang-08, 8 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-actn-yang-08.txt>>.

[I-D.ietf-teas-applicability-actn-slicing]

King, D., Drake, J., Zheng, H., and A. Farrel, "Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Network Slicing", Work in Progress, Internet-Draft, draft-ietf-teas-applicability-actn-slicing-00, 21 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-teas-applicability-actn-slicing-00.txt>>.

[I-D.ietf-teas-ietf-network-slice-nbi-yang]

Wu, B., Dhody, D., Rokui, R., Saad, T., and L. Han, "IETF Network Slice Service YANG Model", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-nbi-yang-01, 4 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slice-nbi-yang-01.txt>>.

[I-D.ietf-teas-ietf-network-slices]

Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "Framework for IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-08, 6 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-ietf-network-slices-08.txt>>.

[I-D.wd-teas-nrp-yang]

Wu, B., Dhody, D., and Y. Cheng, "A YANG Data Model for Network Resource Partition (NRP)", Work in Progress, Internet-Draft, draft-wd-teas-nrp-yang-00, 30 January 2022, <<https://www.ietf.org/archive/id/draft-wd-teas-nrp-yang-00.txt>>.

[NGMN-NS-Concept]

hao , , "NGMN NS Concept", 2016, <https://www.ngmn.org/fileadmin/user_upload/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf>.

[RFC2475]

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <<https://www.rfc-editor.org/info/rfc2475>>.

- [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>>.
- [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <<https://www.rfc-editor.org/info/rfc3758>>.
- [RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <<https://www.rfc-editor.org/info/rfc3931>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <<https://www.rfc-editor.org/info/rfc4594>>.
- [RFC4719] Aggarwal, R., Ed., Townsley, M., Ed., and M. Dos Santos, Ed., "Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)", RFC 4719, DOI 10.17487/RFC4719, November 2006, <<https://www.rfc-editor.org/info/rfc4719>>.
- [RFC5151] Farrel, A., Ed., Ayyangar, A., and JP. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 5151, DOI 10.17487/RFC5151, February 2008, <<https://www.rfc-editor.org/info/rfc5151>>.

- [RFC5654] Niven-Jenkins, B., Ed., Brungard, D., Ed., Betts, M., Ed., Sprecher, N., and S. Ueno, "Requirements of an MPLS Transport Profile", RFC 5654, DOI 10.17487/RFC5654, September 2009, <<https://www.rfc-editor.org/info/rfc5654>>.
- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, DOI 10.17487/RFC7209, May 2014, <<https://www.rfc-editor.org/info/rfc7209>>.
- [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>>.
- [RFC8172] Morton, A., "Considerations for Benchmarking Virtual Network Functions and Their Infrastructure", RFC 8172, DOI 10.17487/RFC8172, July 2017, <<https://www.rfc-editor.org/info/rfc8172>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8370] Beeram, V., Ed., Minei, I., Shakir, R., Pacella, D., and T. Saad, "Techniques to Improve the Scalability of RSVP-TE Deployments", RFC 8370, DOI 10.17487/RFC8370, May 2018, <<https://www.rfc-editor.org/info/rfc8370>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

- [RFC8403] Geib, R., Ed., Filsfils, C., Pignataro, C., Ed., and N. Kumar, "A Scalable and Topology-Aware MPLS Data-Plane Monitoring System", RFC 8403, DOI 10.17487/RFC8403, July 2018, <<https://www.rfc-editor.org/info/rfc8403>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [RFC8568] Bernardos, CJ., Rahman, A., Zuniga, JC., Contreras, LM., Aranda, P., and P. Lynch, "Network Virtualization Research Challenges", RFC 8568, DOI 10.17487/RFC8568, April 2019, <<https://www.rfc-editor.org/info/rfc8568>>.
- [RFC8577] Sitaraman, H., Beeram, V., Parikh, T., and T. Saad, "Signaling RSVP-TE Tunnels on a Shared MPLS Forwarding Plane", RFC 8577, DOI 10.17487/RFC8577, April 2019, <<https://www.rfc-editor.org/info/rfc8577>>.
- [RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <<https://www.rfc-editor.org/info/rfc8578>>.
- [RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <<https://www.rfc-editor.org/info/rfc8655>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.

- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.
- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8964] Varga, B., Ed., Farkas, J., Berger, L., Malis, A., Bryant, S., and J. Korhonen, "Deterministic Networking (DetNet) Data Plane: MPLS", RFC 8964, DOI 10.17487/RFC8964, January 2021, <<https://www.rfc-editor.org/info/rfc8964>>.
- [RFC9085] Previdi, S., Talaulikar, K., Ed., Filsfils, C., Gredler, H., and M. Chen, "Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing", RFC 9085, DOI 10.17487/RFC9085, August 2021, <<https://www.rfc-editor.org/info/rfc9085>>.
- [SFC] "Service Function Chaining", March , <<https://datatracker.ietf.org/wg/sfc/about>>.
- [TS23501] "3GPP TS23.501", 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>>.
- [TS28530] "3GPP TS28.530", 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273>>.
- [TSN] "Time-Sensitive Networking", March , <<https://1.ieee802.org/tsn/>>.

Authors' Addresses

Jie Dong
Huawei
Email: jie.dong@huawei.com

Stewart Bryant
University of Surrey
Email: stewart.bryant@gmail.com

Zhenqiang Li
China Mobile
Email: lizhenqiang@chinamobile.com

Takuya Miyasaka
KDDI Corporation
Email: ta-miyasaka@kddi.com

Young Lee
Samsung
Email: younglee.tx@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
G. Fioccola
Q. Wu, Ed.
Huawei Technologies
D. Ceccarelli
Ericsson
J. Tantsura
Microsoft
7 March 2022

Traffic Engineering (TE) and Service Mapping YANG Model
draft-ietf-teas-te-service-mapping-yang-10

Abstract

This document provides a YANG data model to map customer service models (e.g., the L3VPN Service Model (L3SM)) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Virtual Network (VN) model). These models are referred to as TE Service Mapping Model and are applicable generically to the operator's need for seamless control and management of their VPN services with underlying TE support.

The models are principally used for 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

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 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Purpose of TE Service Mapping for Service Model	4
1.2. Purpose of TE Service Mapping for Network Model	5
1.3. Terminology	6
1.4. Tree diagram	6
1.5. Prefixes in Data Node Names	6
2. TE and Service Related Parameters	8
2.1. VN/Tunnel Selection Requirements	8
2.2. TE Policy	9
2.2.1. Availability Requirement	9
3. YANG Modeling Approach	9
3.1. Forward Compatibility	11
3.2. TE and Network Models	11
4. L3VPN Architecture in the ACTN Context	12
4.1. Service Mapping	16
4.2. Site Mapping	16
5. Applicability of TE-Service Mapping in Generic context	17
6. YANG Data Trees	17
6.1. Service Mapping Types	17
6.2. Service Models	18
6.2.1. L3SM	18
6.2.2. L2SM	19
6.2.3. L1CSM	21
6.3. Network Models	21
6.3.1. L3NM	22
6.3.2. L2NM	22
7. YANG Data Models	23
7.1. ietf-te-service-mapping-types	23
7.2. Service Models	32
7.2.1. ietf-l3sm-te-service-mapping	32
7.2.2. ietf-l2sm-te-service-mapping	35
7.2.3. ietf-l1csm-te-service-mapping	37

7.3. Network Models	39
7.3.1. ietf-l3nm-te-service-mapping	39
7.3.2. ietf-l2nm-te-service-mapping	41
8. Security Considerations	43
9. IANA Considerations	44
10. Acknowledgements	46
11. References	46
11.1. Normative References	46
11.2. Informative References	49
Appendix A. Examples	50
Appendix B. Out of Scope	52
Appendix C. Contributor Addresses	52
Authors' Addresses	53

1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [RFC7950] 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.

[I-D.ietf-teas-actn-vn-yang] defines a YANG model and describes how customers or end-to-end orchestrator can request and/or instantiate a generic virtual network service. [I-D.ietf-teas-actn-yang] 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.

[RFC8466] 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.

[I-D.ietf-ccamp-llcsm-yang] 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 and service related parameters that this document addresses as "new and advanced parameters" that are not included in the service models. Section 3 discusses YANG modeling approach.

1.1. Purpose of TE Service Mapping for Service Model

The TE service mapping for the LxSM supports:

- * A mapping of the LxSM with the underlying TE resources. The TE resources could be in a form of VN, set of TE tunnels, TE abstract topology etc. This mapping can be populated by the network at the time of realization of the service. It is also possible to configure the mapping provided one is aware of VN/tunnels. This mapping model is used only when there is an awareness of VN or TE by the consumer of the model. Otherwise this mapping information is internal and used for monitoring and diagnostics purpose such as telemetry, auto-scaling, closed-loop automation.
- * Possibility to request creation of a new VN/Tunnel to be binded to LxSM .
- * Indication to share the VN/Tunnel sharing (with or without modification) for the LxSM.
- * Support for configuration of underlying TE properties (as apposed to existing VN or tunnels).
- * Provide some additional service characteristics for the LxSM models

1.2. Purpose of TE Service Mapping for Network Model

Apart from the service model, the TE mapping is equally applicable to the Network Models (L3 VPN Service Network Model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm], L2 VPN Service Network Model (L2NM) [I-D.ietf-opsawg-l2nm] etc.). See Section 3.2 for details.

The TE service mapping for the LxNM supports:

- * A mapping of the LxNM with the underlying TE resources. The TE resources could be in a form of VN, set of TE tunnels, TE abstract topology etc. This mapping can be populated by the network or configured. This mapping is useful to understand the TE realization of the LxVPN as well for monitoring/diagnostic purpose.
- * Possibility to request creation of a new VN/Tunnel to be binded to LxNM .
- * Indication to share the VN/Tunnel sharing (with or without modification) for the LxNM.
- * Support for configuration of underlying TE properties (as apposed to existing VN or tunnels).

- * Provide some additional service characteristics for the LxNM models

1.3. Terminology

Refer to [RFC8453], [RFC7926], and [RFC8309] for the key terms used in this document.

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

1.4. 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].

1.5. 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
tsmt	ietf-te-service-mapping-types	[RFCXXXX]
l1csm	ietf-l1csm	[I-D.ietf-ccamp-l1csm-yang]
l2vpn-svc	ietf-l2vpn-svc	[RFC8466]
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	[I-D.ietf-teas-actn-vn-yang]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te	ietf-te	[I-D.ietf-teas-yang-te]
l2vpn-ntw	ietf-l2vpn-ntw	[I-D.ietf-opsawg-l2nm]
l3vpn-ntw	ietf-l3vpn-ntw	[I-D.ietf-opsawg-l3sm-l3nm]
rt	ietf-routing	[RFC8349]
sr-policy	ietf-sr-policy	[I-D.ietf-spring-sr-policy-yang]

Table 1: Prefixes and corresponding YANG modules

Note: The RFC Editor should replace XXXX with the number assigned to the RFC once this draft becomes an RFC.

2. TE and Service Related Parameters

While L1/L2/L3 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 these 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 VN/TE tunnels to be established. This may occur when there are no suitable existing VN/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 4 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 new TE tunnels which can be shared with other VPN services if need be.

- * 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).
- * TE Mapping Template - This mode allows a VPN service to use a mapping template containing constraints and optimization criteria. This allows mapping with the underlay TE characteristics without first creating a VN or tunnels to map. The VPN service could be mapped to a template first. Once the VN/Tunnels are actually created/selected for the VPN service, the mapping based on the actual TE resources is created.

2.2. TE Policy

The service models could be associated with various policies related to mapping the underlying TE resources. A color could be used to map to the underlying colored TE resources. The desired protection and availability requirements could be specified.

2.2.1. 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 document.

3. YANG Modeling Approach

This section provides how the TE and Service mapping parameters are supported using augmentation of the existing service models (i.e., [I-D.ietf-ccamp-l1csm-yang], [RFC8466], and [RFC8299]). Figure 1 shows the scope of the Augmented LxSM Model.

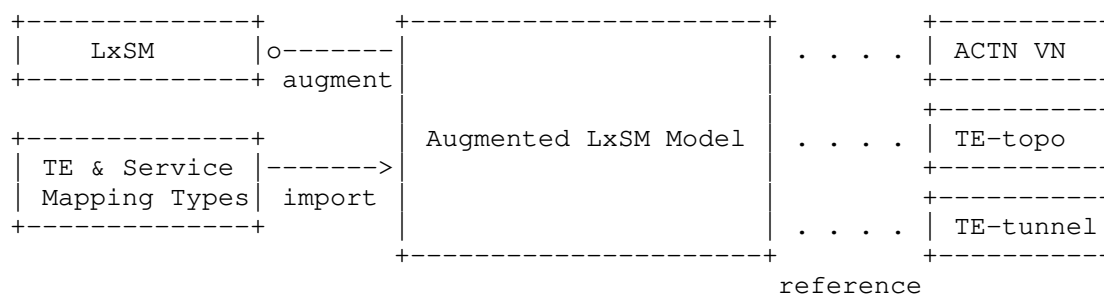


Figure 1: Augmented LxSM Model

The Augmented LxSM model (where $x=1,2,3$) augments the basic LxSM model while importing the common TE and Service related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The TE and 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 mapping-type common model.

The mapping could be made to any underlying TE resources such as VN, TE topology abstract node (and its connectivity matrix), set of TE tunnels etc. This flexibility from the modeling point of view allows for various use cases at both service and network model.

The role of the augmented LxSM 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. Note that this should be done only if the operator understands the VN/Tunnel resources and the MDSC is willing to share that information. 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. 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 [RFC8795], 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.

The YANG models defined in this document conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

3.1. Forward Compatibility

The YANG module defined in this document supports three existing service models via augmenting while sharing the common TE and 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.

Appendix B highlights some features that are deemed out of scope of this document.

3.2. TE and Network Models

The L2/L3 network models (L2NM, L3NM) are intended to describe a VPN Service in the Service Provider Network. It contains information of the Service Provider network and might include allocated resources. It can be used by network controllers to manage and control the VPN Service configuration in the Service Provider network.

Similar to service model, the existing network models (i.e., [I-D.ietf-opsawg-l3sm-l3nm], and [I-D.ietf-opsawg-l2nm]) are augmented to include the TE and Service mapping parameters. Figure 2 shows the scope of the Augmented LxNM Model.

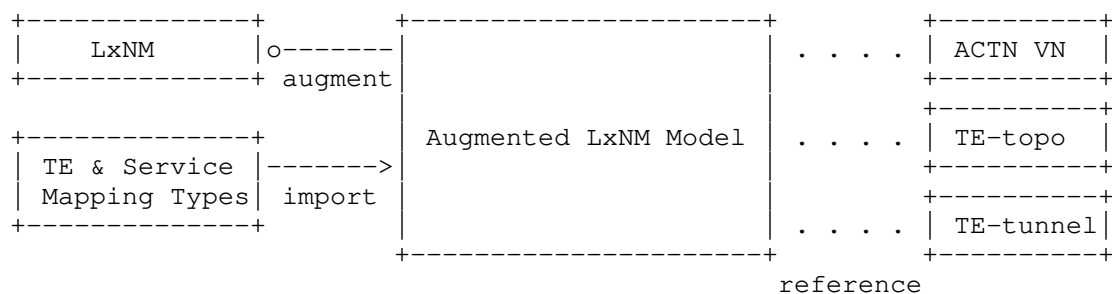


Figure 2: Augmented LxNM Model

The Augmented LxNM model (where x=2,3) augments the basic LxNM model while importing the common TE mapping related parameters (defined in Section 2) grouping information from TE and Service Mapping Types. The role of the augmented LxNM network model is to expose the mapping relationship between network models and TE models.

4. L3VPN Architecture in the ACTN Context

Figure 3 shows the architectural context of this document referencing the ACTN components and interfaces.

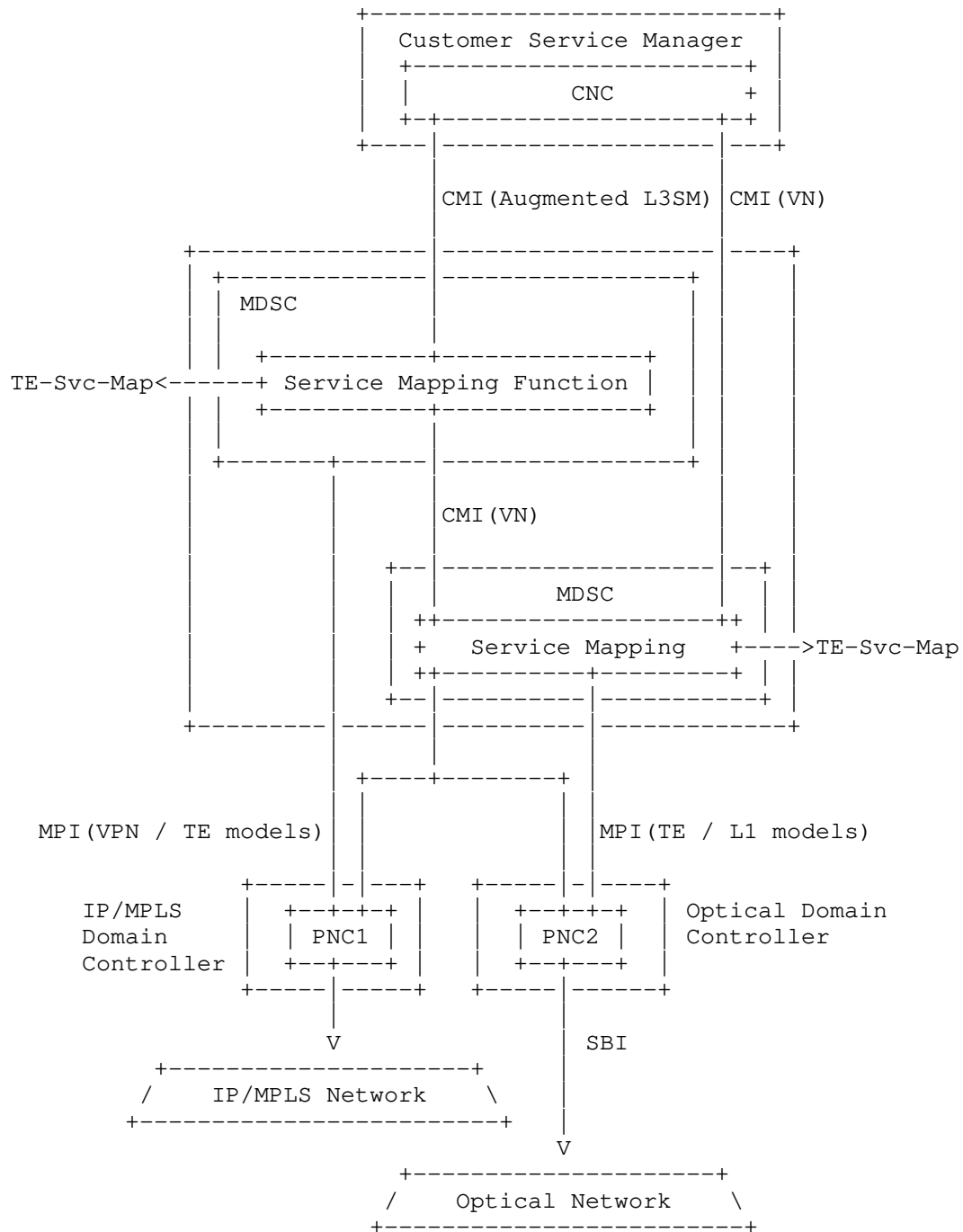


Figure 3: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 3.

- * 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 3 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.

The three main interfaces are shown in Figure 3 and listed below.

- * 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 3, 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 [I-D.ietf-teas-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 2: : 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

6.1. Service Mapping Types

```

module: ietf-te-service-mapping-types
  +--rw te-mapping-templates
    +--rw te-mapping-template* [id]
      +--rw id                te-mapping-template-id
      +--rw description?      string
      +--rw map-type?         identityref
      +--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 explicit-route-include-objects
      |   ...
      +--rw tiebreakers
      +--rw tiebreaker* [tiebreaker-type]
      ...
    +--:(objective-function)
      {path-optimization-objective-function}?
      +--rw objective-function
      +--rw objective-function-type?  identityref

```

6.2. Service Models

6.2.1. L3SM

```

module: ietf-l3sm-te-service-mapping

augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services
  /l3vpn-svc:vpn-service:
    +--rw te-service-mapping!
      +--rw te-mapping
        +--rw map-type?                                identityref
        +--rw te-policy
          +--rw color?                                uint32
          +--rw protection-type?                      identityref
          +--rw availability-type?                    identityref
        +--rw (te)?
          +--:(vn)
            +--rw vn*
              -> /vn:virtual-network/vn/vn-id
          +--:(te-topo)
            +--rw te-topology-identifier
              +--rw provider-id?    te-global-id
              +--rw client-id?      te-global-id
              +--rw topology-id?    te-topology-id
            +--rw abstract-node?
              -> /nw:networks/network/node/node-id

```

```

    +---:(te-tunnel)
    |   +---rw te-tunnel*           te:tunnel-ref
    |   +---rw sr-policy*
    |       [policy-color-ref policy-endpoint-ref]
    |       {sr-policy}?
    |   +---rw policy-color-ref     leafref
    |   +---rw policy-endpoint-ref  leafref
    +---rw te-mapping-template-ref?
    -> /tsmt:te-mapping-templates/te-mapping-template/id
    {template}?
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
    /l3vpn-svc:site-network-accesses
    /l3vpn-svc:site-network-access:
    +---rw (te)?
    |   +---:(vn)
    |   |   +---rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
    |   +---:(te)
    |   |   +---rw ltp?     te-types:te-tp-id
    |   +---rw ltp?     te-types:te-tp-id
    +---rw ltp?     te-types:te-tp-id
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
    /l3vpn-svc:service/l3vpn-svc:qos/l3vpn-svc:qos-profile
    /l3vpn-svc:qos-profile/l3vpn-svc:custom/l3vpn-svc:classes
    /l3vpn-svc:class:
    +---rw (te)?
    |   +---:(vn)
    |   |   +---rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
    |   +---:(te)
    |   |   +---rw ltp?     te-types:te-tp-id
    |   +---rw ltp?     te-types:te-tp-id
    +---rw ltp?     te-types:te-tp-id
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
    /l3vpn-svc:site-network-accesses
    /l3vpn-svc:site-network-access/l3vpn-svc:service
    /l3vpn-svc:qos/l3vpn-svc:qos-profile
    /l3vpn-svc:qos-profile/l3vpn-svc:custom/l3vpn-svc:classes
    /l3vpn-svc:class:
    +---rw (te)?
    |   +---:(vn)
    |   |   +---rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
    |   +---:(te)
    |   |   +---rw ltp?     te-types:te-tp-id
    |   +---rw ltp?     te-types:te-tp-id
    +---rw ltp?     te-types:te-tp-id

```

6.2.2. L2SM

module: ietf-l2sm-te-service-mapping

```

augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services
    /l2vpn-svc:vpn-service:
    +---rw te-service-mapping!
    +---rw te-mapping
    +---rw map-type?           identityref

```

```

+---rw te-policy
|   +---rw color?                uint32
|   +---rw protection-type?      identityref
|   +---rw availability-type?    identityref
+---rw (te)?
|   +---:(vn)
|   |   +---rw vn*
|   |   |   -> /vn:virtual-network/vn/vn-id
|   +---:(te-topo)
|   |   +---rw te-topology-identifier
|   |   |   +---rw provider-id?    te-global-id
|   |   |   +---rw client-id?     te-global-id
|   |   |   +---rw topology-id?   te-topology-id
|   |   +---rw abstract-node?
|   |   |   -> /nw:networks/network/node/node-id
|   +---:(te-tunnel)
|   |   +---rw te-tunnel*          te:tunnel-ref
|   |   +---rw sr-policy*
|   |   |   [policy-color-ref policy-endpoint-ref]
|   |   |   {sr-policy}?
|   |   |   +---rw policy-color-ref    leafref
|   |   |   +---rw policy-endpoint-ref leafref
+---rw te-mapping-template-ref?
|   -> /tsmt:te-mapping-templates/te-mapping-template/id
|   {template}?
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
|   /l2vpn-svc:site-network-accesses
|   /l2vpn-svc:site-network-access:
+---rw (te)?
|   +---:(vn)
|   |   +---rw vn-ap*    -> /vn:access-point/ap/vn-ap/vn-ap-id
|   +---:(te)
|   |   +---rw ltp?      te-types:te-tp-id
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
|   /l2vpn-svc:service/l2vpn-svc:qos/l2vpn-svc:qos-profile
|   /l2vpn-svc:qos-profile/l2vpn-svc:custom/l2vpn-svc:classes
|   /l2vpn-svc:class:
+---rw (te)?
|   +---:(vn)
|   |   +---rw vn-ap*    -> /vn:access-point/ap/vn-ap/vn-ap-id
|   +---:(te)
|   |   +---rw ltp?      te-types:te-tp-id
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
|   /l2vpn-svc:site-network-accesses
|   /l2vpn-svc:site-network-access/l2vpn-svc:service
|   /l2vpn-svc:qos/l2vpn-svc:qos-profile
|   /l2vpn-svc:qos-profile/l2vpn-svc:custom/l2vpn-svc:classes
|   /l2vpn-svc:class:

```

```

+--rw (te)?
+--:(vn)
|   +--rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
+--:(te)
    +--rw ltp?      te-types:te-tp-id

```

6.2.3. L1CSM

```
module: ietf-llcsm-te-service-mapping
```

```

augment /llcsm:ll-connectivity/llcsm:services/llcsm:service:
+--rw te-service-mapping!
+--rw te-mapping
+--rw map-type?                               identityref
+--rw te-policy
|   +--rw color?                               uint32
|   +--rw protection-type?                     identityref
|   +--rw availability-type?                   identityref
+--rw (te)?
+--:(vn)
|   +--rw vn*
|   |   -> /vn:virtual-network/vn/vn-id
+--:(te-topo)
|   +--rw te-topology-identifier
|   |   +--rw provider-id?                     te-global-id
|   |   +--rw client-id?                      te-global-id
|   |   +--rw topology-id?                    te-topology-id
|   +--rw abstract-node?
|   |   -> /nw:networks/network/node/node-id
+--:(te-tunnel)
|   +--rw te-tunnel*                           te:tunnel-ref
|   +--rw sr-policy*
|   |   [policy-color-ref policy-endpoint-ref]
|   |   {sr-policy}?
|   |   +--rw policy-color-ref                 leafref
|   |   +--rw policy-endpoint-ref              leafref
+--rw te-mapping-template-ref?
|   -> /tsmt:te-mapping-templates/te-mapping-template/id
|   {template}?
augment /llcsm:ll-connectivity/llcsm:access/llcsm:unis/llcsm:uni:
+--rw (te)?
+--:(vn)
|   +--rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
+--:(te)
    +--rw ltp?      te-types:te-tp-id

```

6.3. Network Models

6.3.1. L3NM

```

module: ietf-l3nm-te-service-mapping

augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
  /l3vpn-ntw:vpn-service:
    +--rw te-service-mapping!
      +--rw te-mapping
        +--rw map-type?                               identityref
        +--rw te-policy
          +--rw color?                                uint32
          +--rw protection-type?                       identityref
          +--rw availability-type?                     identityref
        +--rw (te)?
          +--:(vn)
            +--rw vn*
              -> /vn:virtual-network/vn/vn-id
          +--:(te-topo)
            +--rw te-topology-identifier
              +--rw provider-id?      te-global-id
              +--rw client-id?        te-global-id
              +--rw topology-id?      te-topology-id
            +--rw abstract-node?
              -> /nw:networks/network/node/node-id
          +--:(te-tunnel)
            +--rw te-tunnel*           te:tunnel-ref
            +--rw sr-policy*
              [policy-color-ref policy-endpoint-ref]
              {sr-policy}?
              +--rw policy-color-ref   leafref
              +--rw policy-endpoint-ref leafref
          +--rw te-mapping-template-ref?
            -> /tsmt:te-mapping-templates/te-mapping-template/id
            {template}?
augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
  /l3vpn-ntw:vpn-service/l3vpn-ntw:vpn-nodes
  /l3vpn-ntw:vpn-node/l3vpn-ntw:vpn-network-accesses
  /l3vpn-ntw:vpn-network-access:
    +--rw (te)?
      +--:(vn)
        | +--rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
      +--:(te)
        +--rw ltp?      te-types:te-tp-id

```

6.3.2. L2NM

```

module: ietf-l2nm-te-service-mapping

augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
  /l2vpn-ntw:vpn-service:
    +--rw te-service-mapping!
      +--rw te-mapping
        +--rw map-type?                               identityref
        +--rw te-policy
          +--rw color?                                uint32
          +--rw protection-type?                       identityref
          +--rw availability-type?                     identityref
        +--rw (te)?
          +--:(vn)
            +--rw vn*
              -> /vn:virtual-network/vn/vn-id
          +--:(te-topo)
            +--rw te-topology-identifier
              +--rw provider-id?                       te-global-id
              +--rw client-id?                         te-global-id
              +--rw topology-id?                      te-topology-id
            +--rw abstract-node?
              -> /nw:networks/network/node/node-id
          +--:(te-tunnel)
            +--rw te-tunnel*                           te:tunnel-ref
            +--rw sr-policy*
              [policy-color-ref policy-endpoint-ref]
              {sr-policy}?
              +--rw policy-color-ref                   leafref
              +--rw policy-endpoint-ref                 leafref
          +--rw te-mapping-template-ref?
            -> /tsmt:te-mapping-templates/te-mapping-template/id
            {template}?
augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
  /l2vpn-ntw:vpn-service/l2vpn-ntw:vpn-nodes
  /l2vpn-ntw:vpn-node/l2vpn-ntw:vpn-network-accesses
  /l2vpn-ntw:vpn-network-access:
    +--rw (te)?
      +--:(vn)
        +--rw vn-ap*   -> /vn:access-point/ap/vn-ap/vn-ap-id
      +--:(te)
        +--rw ltp?     te-types:te-tp-id

```

7. YANG Data Models

The YANG codes are as follows:

7.1. ietf-te-service-mapping-types


```
<CODE BEGINS> file "ietf-te-service-mapping-types@2022-03-07.yang"
module ietf-te-service-mapping-types {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";
  prefix tsmt;

  /* Import te-types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }

  /* Import network model */

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

  /* Import TE model */

  import ietf-te {
    prefix te;
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }

  /* Import VN model */

  import ietf-vn {
    prefix vn;
    reference
      "I-D.ietf-teas-actn-vn-yang: A Yang Data Model for VN Operation";
  }

  /* Import Routing */

  import ietf-routing {
    prefix rt;
    reference
      "RFC 8349: A YANG Data Model for Routing Management";
  }
}
```

```
/* Import SR Policy */

import ietf-sr-policy {
  prefix sr-policy;
  reference
    "I-D.ietf-spring-sr-policy-yang: YANG Data Model for Segment
    Routing Policy";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Young Lee
            <mailto:younglee.tx@gmail.com>
  Editor:   Dhruv Dhody
            <mailto:dhruv.ietf@gmail.com>
  Editor:   Qin Wu
            <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for TE & Service mapping
  parameters and policies as a common grouping applicable to
  various service models (e.g., L1CSM, L2SM, L3SM, etc.)

  Copyright (c) 2022 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; see the
  RFC itself for full legal notices.";

revision 2022-03-07 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
```

```
* Features
*/

feature template {
  description
    "Support TE mapping templates.";
}

feature sr-policy {
  description
    "Support SR Policy.";
}

/*
 * 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 hard-isolation {
  base new;
  description
    "Hard isolation.";
}

identity detnet-hard-isolation {
  base hard-isolation;
  description
    "Hard isolation with deterministic characteristics.";
}

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 none {
    base map-type;
    description
      "The VPN service is not mapped to any underlying TE";
  }

  /*
   * 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%";
    }

    /*
     * Typedef
     */

    typedef te-mapping-template-id {
        type string;
        description
            "Identifier for a TE mapping template.";
    }

    /*
     * Groupings
     */

    grouping te-ref {
        description
            "The reference to TE.";
        choice te {
            description
                "How the VPN is mapped to a VN, Topology, Tunnel, SR Policy
                etc.";
            case vn {
                leaf-list vn {
                    type leafref {
                        path "/vn:virtual-network/vn:vn/vn:vn-id";
                    }
                    description
                        "The reference to VN";
                    reference
                        "RFC 8453: Framework for Abstraction and Control of TE
                        Networks (ACTN)";
                }
            }
        }
        case te-topo {
            /*An identifier to the TE Topology Model where the abstract
            nodes and links of the Topology can be found for Type 2
            VNs as defined in RFC 8453*/
            uses te-types:te-topology-identifier;
            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";
    reference
      "RFC 8795: YANG Data Model for Traffic Engineering (TE)
       Topologies";
  }
}
case te-tunnel {
  leaf-list te-tunnel {
    type te:tunnel-ref;
    description
      "Reference to TE Tunnels";
    reference
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
       Engineering Tunnels and Interfaces";
  }
  list sr-policy {
    if-feature "sr-policy";
    /*Headend should also be there!*/
    key "policy-color-ref policy-endpoint-ref";
    description
      "SR Policy";
    leaf policy-color-ref {
      type leafref {
        path
          "/rt:routing/sr-policy:segment-routing"
          + "/sr-policy:traffic-engineering/sr-policy:policies"
          + "/sr-policy:policy/sr-policy:color";
      }
      description
        "Reference to sr-policy color";
    }
    leaf policy-endpoint-ref {
      type leafref {
        path
          "/rt:routing/sr-policy:segment-routing"
          + "/sr-policy:traffic-engineering/sr-policy:policies"
          + "/sr-policy:policy/sr-policy:endpoint";
      }
      description
        "Reference to sr-policy endpoint";
    }
  }
}
}
```

```
    leaf te-mapping-template-ref {
      if-feature "template";
      type leafref {
        path "/tsmt:te-mapping-templates/"
          + "tsmt:te-mapping-template/tsmt:id";
      }
      description
        "An identifier to the TE Mapping Template where the TE
        constraints and optimization criteria are specified.";
    }
  }

//grouping

grouping te-endpoint-ref {
  description
    "The reference to TE endpoints.";
  choice te {
    description
      "How the TE endpoint is defined by VN's AP or TE's LTP";
    case vn {
      leaf-list vn-ap {
        type leafref {
          path "/vn:access-point/vn:ap/vn:vn-ap/vn:vn-ap-id";
        }
        description
          "The reference to VNAP";
        reference
          "RFC 8453: Framework for Abstraction and Control of TE
          Networks (ACTN)";
      }
    }
    case te {
      leaf ltp {
        type te-types:te-tp-id;
        description
          "Reference LTP in the TE-topology";
        reference
          "RFC 8795: YANG Data Model for Traffic Engineering (TE)
          Topologies";
      }
    }
  }
}

//grouping

grouping te-policy {
```

```
    description
      "Various underlying TE policy requirements";
    leaf color {
      type uint32;
      description
        "Maps to the underlying colored TE resources";
    }
    leaf protection-type {
      type identityref {
        base te-types:lsp-protection-type;
      }
      description
        "Desired protection level for the underlying
         TE resources";
    }
    leaf availability-type {
      type identityref {
        base availability-type;
      }
      description
        "Availability Requirement for the Service";
    }
  }
}

//grouping

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";
    }
    container te-policy {
      uses te-policy;
      description
        "Desired Underlying TE Policy";
    }
    uses te-ref;
  }
}
```



```
//grouping

container te-mapping-templates {
  description
    "The TE constraints and optimization criteria";
  list te-mapping-template {
    key "id";
    leaf id {
      type te-mapping-template-id;
      description
        "Identification of the Template to be used.";
    }
    leaf description {
      type string;
      description
        "Description of the template.";
    }
    leaf map-type {
      type identityref {
        base map-type;
      }
      must "0 = derived-from-or-self(., 'none')" {
        error-message "The map-type must be other than "
          + "none";
      }
      description
        "Map type for the VN/Tunnel creation/
        selection.";
    }
    uses te-types:generic-path-constraints;
    uses te-types:generic-path-optimization;
    description
      "List for templates.";
  }
}
}
<CODE ENDS>
```

7.2. Service Models

7.2.1. ietf-l3sm-te-service-mapping

```
<CODE BEGINS> file "ietf-l3sm-te-service-mapping@2022-03-07.yang"
module ietf-l3sm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";
  prefix l3-tsm;
```

```
import ietf-te-service-mapping-types {
  prefix tsmt;
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}
import ietf-l3vpn-svc {
  prefix l3vpn-svc;
  reference
    "RFC 8299: YANG Data Model for L3VPN Service Delivery";
}

organization
  "IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Young Lee
            <mailto:younglee.tx@gmail.com>
  Editor:   Dhruv Dhody
            <mailto:dhruv.ietf@gmail.com>
  Editor:   Qin Wu
            <mailto: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.

  Copyright (c) 2022 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).
```

```
/*
 * Augmentation to L3SM
 */

augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services"
  + "/l3vpn-svc:vpn-service" {
  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";
    uses tsmt:te-mapping;
  }
}

//augment

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 of L3SM network-access
    to TE endpoints";
  uses tsmt:te-endpoint-ref;
}

//augment

augment
  "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
+ "/l3vpn-svc:service/l3vpn-svc:qos/l3vpn-svc:qos-profile"
+ "/l3vpn-svc:qos-profile/l3vpn-svc:custom"
+ "/l3vpn-svc:classes/l3vpn-svc:class" {
  description
    "This augment is for per-class in site for custom QoS profile";
  uses tsmt:te-endpoint-ref;
}

augment
  "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
+ "/l3vpn-svc:site-network-accesses"
+ "/l3vpn-svc:site-network-access"
+ "/l3vpn-svc:service/l3vpn-svc:qos/l3vpn-svc:qos-profile"
+ "/l3vpn-svc:qos-profile/l3vpn-svc:custom"
+ "/l3vpn-svc:classes/l3vpn-svc:class" {
  description
    "This augment is for per-class in site-network-access for custom
```

```
        QoS profile";
        uses tsmt:te-endpoint-ref;
    }
}
<CODE ENDS>
```

7.2.2. ietf-l2sm-te-service-mapping

```
<CODE BEGINS> file "ietf-l2sm-te-service-mapping@2022-03-07.yang"
module ietf-l2sm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";
  prefix l2-tsm;

  import ietf-te-service-mapping-types {
    prefix tsmt;
    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }
  import ietf-l2vpn-svc {
    prefix l2vpn-svc;
    reference
      "RFC 8466: A YANG Data Model for Layer 2 Virtual Private Network
      (L2VPN) Service Delivery";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
    WG List:  <mailto:teas@ietf.org>

    Editor:   Young Lee
              <mailto:younglee.tx@gmail.com>
    Editor:   Dhruv Dhody
              <mailto:dhruv.ietf@gmail.com>
    Editor:   Qin Wu
              <mailto:bill.wu@huawei.com>";
  description
    "This module contains a YANG module for the mapping of Layer 2
    Service Model (L2SM) to the TE and VN.

    Copyright (c) 2022 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; see the RFC itself for full legal notices.";

```
revision 2022-03-07 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L2SM
 */

augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services/"
  + "l2vpn-svc:vpn-service" {
  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";
    uses tsmt:te-mapping;
  }
}

//augment

augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
  + "l2vpn-svc:site-network-accesses"
  + "l2vpn-svc:site-network-access" {
  description
    "This augment the L2SM network-access with a reference
    to TE endpoints when underlying TE is used";
  uses tsmt:te-endpoint-ref;
}

//augment

augment
  "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
+ "/l2vpn-svc:service/l2vpn-svc:qos/l2vpn-svc:qos-profile"
```

```

+ "/l2vpn-svc:qos-profile/l2vpn-svc:custom"
+ "/l2vpn-svc:classes/l2vpn-svc:class" {
  when './l2vpn-svc:bandwidth/l2vpn-svc:end-to-end' {
    description
      "applicable only with end-to-end";
  }
  description
    "This augment is for per-class in site for custom QoS profile";
  uses tsmt:te-endpoint-ref;
}

augment
  "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
+ "/l2vpn-svc:site-network-accesses"
+ "/l2vpn-svc:site-network-access"
+ "/l2vpn-svc:service/l2vpn-svc:qos/l2vpn-svc:qos-profile"
+ "/l2vpn-svc:qos-profile/l2vpn-svc:custom"
+ "/l2vpn-svc:classes/l2vpn-svc:class" {
  description
    "This augment is for per-class in site-network-access for custom
    QoS profile";
  uses tsmt:te-endpoint-ref;
}
}
<CODE ENDS>

```

7.2.3. ietf-llcsm-te-service-mapping

```

<CODE BEGINS> file "ietf-llcsm-te-service-mapping@2022-03-07.yang"
module ietf-llcsm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-llcsm-te-service-mapping";
  prefix ll-tsm;

  import ietf-te-service-mapping-types {
    prefix tsmt;
    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }
  import ietf-llcsm {
    prefix llcsm;
    reference
      "I-D.ietf-ccamp-llcsm-yang: A YANG Data Model for L1 Connectivity
      Service Model (L1CSM)";
  }

  organization

```

```
"IETF Traffic Engineering Architecture and Signaling (TEAS)
  Working Group";
contact
  "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
  WG List:  <mailto:teas@ietf.org>

  Editor:   Young Lee
            <mailto:younglee.tx@gmail.com>
  Editor:   Dhruv Dhody
            <mailto:dhruv.ietf@gmail.com>
  Editor:   Qin Wu
            <mailto: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

  Copyright (c) 2022 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; see the
  RFC itself for full legal notices.";

revision 2022-03-07 {
  description
    "Initial revision.";
  reference
    "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L1CSM
 */

augment "/l1csm:l1-connectivity/l1csm:services/l1csm: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";
    uses tsmt:te-mapping;
  }
}
```

```
    }  
  }  
  
  //augment  
  
  augment "/l1csm:l1-connectivity/l1csm:access/l1csm:unis/"  
    + "l1csm:uni" {  
    description  
      "This augment the L1CSM UNI with a reference  
      to TE endpoints";  
    uses tsmt:te-endpoint-ref;  
  }  
  
  //augment  
}  
<CODE ENDS>
```

7.3. Network Models

7.3.1. ietf-l3nm-te-service-mapping

```
<CODE BEGINS> file "ietf-l3nm-te-service-mapping@2022-03-07.yang"  
module ietf-l3nm-te-service-mapping {  
  yang-version 1.1;  
  namespace  
    "urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping";  
  prefix l3nm-tsm;  
  
  import ietf-te-service-mapping-types {  
    prefix tsmt;  
    reference  
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";  
  }  
  import ietf-l3vpn-ntw {  
    prefix l3vpn-ntw;  
    reference  
      "I-D.ietf-opsawg-l3sm-l3nm: A Layer 3 VPN Network YANG Model";  
  }  
  
  organization  
    "IETF Traffic Engineering Architecture and Signaling (TEAS)  
    Working Group";  
  contact  
    "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>  
    WG List:  <mailto:teas@ietf.org>  
  
    Editor:    Young Lee  
              <mailto:younglee.tx@gmail.com>
```



```
Editor:  Dhruv Dhody
        <mailto:dhruv.ietf@gmail.com>
Editor:  Qin Wu
        <mailto:bill.wu@huawei.com>";
description
  "This module contains a YANG module for the mapping of Layer 3
  Network Model (L3NM) to the TE and VN.

  Copyright (c) 2022 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; see the
  RFC itself for full legal notices.";

revision 2022-03-07 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L3NM
 */

augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
  + "/l3vpn-ntw:vpn-service" {
  description
    "L3SM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "Indicates L3 network to TE mapping";
    description
      "Container to augment l3nm to TE parameters and mapping";
    uses tsmt:te-mapping;
  }
}

//augment

augment "/l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services"
  + "/l3vpn-ntw:vpn-service"
```

```
    + "/l3vpn-ntw:vpn-nodes/l3vpn-ntw:vpn-node"
    + "/l3vpn-ntw:vpn-network-accesses"
    + "/l3vpn-ntw:vpn-network-access" {
  description
    "This augment the L3NM network-access with a reference
    to TE endpoints when underlying TE is used";
  uses tsmt:te-endpoint-ref;
}

//augment
}
<CODE ENDS>
```

7.3.2. ietf-l2nm-te-service-mapping

```
<CODE BEGINS> file "ietf-l2nm-te-service-mapping@2022-03-07.yang"
module ietf-l2nm-te-service-mapping {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping";
  prefix l2nm-tsm;

  import ietf-te-service-mapping-types {
    prefix tsmt;
    reference
      "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
  }
  import ietf-l2vpn-ntw {
    prefix l2vpn-ntw;
    reference
      "I-D.ietf-opsawg-l2nm: A Layer 2 VPN Network YANG Model";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/about/>
    WG List:  <mailto:teas@ietf.org>

    Editor:   Young Lee
              <mailto:younglee.tx@gmail.com>
    Editor:   Dhruv Dhody
              <mailto:dhruv.ietf@gmail.com>
    Editor:   Qin Wu
              <mailto:bill.wu@huawei.com>";
  description
    "This module contains a YANG module for the mapping of Layer 2
```

Network Model (L2NM) to the TE and VN.

Copyright (c) 2022 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; see the RFC itself for full legal notices.";

```
revision 2022-03-07 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L2NM
 */

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
  + "/l2vpn-ntw:vpn-service" {
  description
    "L2SM augmented to include TE parameters and mapping";
  container te-service-mapping {
    presence "Indicates L2 network to TE mapping";
    description
      "Container to augment l2nm to TE parameters and mapping";
    uses tsmt:te-mapping;
  }
}

//augment

augment "/l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services"
  + "/l2vpn-ntw:vpn-service"
  + "/l2vpn-ntw:vpn-nodes/l2vpn-ntw:vpn-node"
  + "/l2vpn-ntw:vpn-network-accesses"
  + "/l2vpn-ntw:vpn-network-access" {
  description
    "This augment the L2NM network-access with a reference
    to TE endpoints when underlying TE is used";
```

```
    uses tsmt:te-endpoint-ref;
  }

  //augment
}
<CODE ENDS>
```

8. Security Considerations

The YANG modules defined in this document is designed to be accessed via network management protocol such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446]

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

There are a number of data nodes defined in the YANG modules 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. These are the subtrees and data nodes and their sensitivity/vulnerability:

- * /l3vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/
- can configure TE Service mapping.
- * /l3vpn-svc/sites/site/site-network-accesses/site-network-access/
te/ - can configure TE Endpoint mapping.
- * /l2vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/
- can configure TE Service mapping.
- * /l2vpn-svc/sites/site/site-network-accesses/site-network-access/
te/ - can configure TE Endpoint mapping.
- * /l1-connectivity/services/service/te-service-mapping/te-mapping/ -
can configure TE Service mapping.
- * /l1-connectivity/access/unis/uni/te/ - can configure TE Endpoint
mapping.

- * /l3vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/
- can configure TE Network mapping.
- * /l3vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-network-accesses/vpn-network-access/te/ - can configure TE Endpoint mapping.
- * /l2vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/
- can configure TE Network mapping.
- * /l2vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-network-accesses/vpn-network-access/te/ - can configure TE Endpoint mapping.

Unauthorized access to above list can adversely affect the VPN service.

Some of the readable data nodes in the 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. The TE related parameters attached to the VPN service can leak sensitive information about the network. This is applicable to all elements in the yang models defined in this document.

This document has no RPC defined.

9. IANA Considerations

This document request the IANA to register six URIs in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registrations are requested -

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-l3sm-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-l1lcsm-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-l3nm-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-l2nm-te-service-mapping
Registrant Contact: The IESG.

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

This document request the IANA to register six YANG modules in the
"YANG Module Names" registry [RFC6020], as follows -

Name: ietf-te-service-mapping-types
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types
Prefix: tsmt
Reference: [This.I-D]

Name: ietf-l3sm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping
Prefix: l3-tsm
Reference: [This.I-D]

Name: ietf-l2sm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
Prefix: l2-tsm
Reference: [This.I-D]

Name: ietf-l1csm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping
Prefix: l1-tsm
Reference: [This.I-D]

Name: ietf-l3nm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping
Prefix: l3nm-tsm
Reference: [This.I-D]

Name: ietf-l2nm-te-service-mapping
Namespace: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping
Prefix: l2nm-tsm
Reference: [This.I-D]

10. Acknowledgements

We thank Diego Caviglia, and Igor Bryskin for useful discussions and motivation for this work.

11. References

11.1. Normative References

[I-D.ietf-ccamp-l1csm-yang]
Lee, Y., Lee, K., Zheng, H., Dios, O. G. D., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", Work in Progress, Internet-Draft, draft-ietf-ccamp-l1csm-yang-16, 13 December 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-ccamp-l1csm-yang-16>>.

- [I-D.ietf-opsawg-l2nm]
Barguil, S., Dios, O. G. D., Boucadair, M., and L. A. Munoz, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-12, 22 November 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l2nm-12>>.
- [I-D.ietf-opsawg-l3sm-l3nm]
Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", Work in Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-18, 8 October 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-l3sm-l3nm-18>>.
- [I-D.ietf-spring-sr-policy-yang]
Raza, K., Sawaya, R., Shunwan, Z., Voyer, D., Durrani, M., Matsushima, S., and V. P. Beeram, "YANG Data Model for Segment Routing Policy", Work in Progress, Internet-Draft, draft-ietf-spring-sr-policy-yang-01, 7 April 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-policy-yang-01>>.
- [I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-14, 7 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-actn-vn-yang-14>>.
- [I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. D. Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-29, 7 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-yang-te-29>>.
- [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>>.

- [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>>.
- [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>>.
- [RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <<https://www.rfc-editor.org/info/rfc8299>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <<https://www.rfc-editor.org/info/rfc8466>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [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>>.

11.2. Informative References

- [I-D.dhody-teas-te-traffic-yang] Dhody, D., "Traffic Mapping YANG model for Traffic Engineering (TE)", Work in Progress, Internet-Draft, draft-dhody-teas-te-traffic-yang-00, 24 October 2021, <<https://datatracker.ietf.org/doc/html/draft-dhody-teas-te-traffic-yang-00>>.
- [I-D.ietf-teas-actn-yang] Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B. Y., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", Work in Progress, Internet-Draft, draft-ietf-teas-actn-yang-08, 8 September 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-actn-yang-08>>.
- [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>>.
- [RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <<https://www.rfc-editor.org/info/rfc8199>>.

- [RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <<https://www.rfc-editor.org/info/rfc8309>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.

Appendix A. Examples

This section details a few examples on how the TE-service mapping is used in various scenarios.

Example 1: An L3VPN service with an optimization criteria for the underlying TE as delay can be set in the mapping template and then augmented to the L3SM service.

```
{
  "te-mapping-template":[
    {
      "id": "delay",
      "map-type": "select",
      "optimizations":
      {
        "algorithm":{
          "optimization-metric": [
            {
              "metric-type":"path-metric-delay-average"
            }
          ]
        }
      }
    }
  ]
}
```

The L3SM service can map it to the existing least delay TE resources in form of a VN or TE-tunnels.

Example 2: An L2VPN service with a bandwidth constraint and a hop-limit criteria for the underlying TE can be set in the mapping template and then augmented to the L2SM service.

```

{
  "te-mapping-template":[
    {
      "id": "bw-hop",
      "map-type": "new",
      "path-constraints":{
        "te-bandwidth":{
          "generic":10000
        },
        "path-metric-bounds":{
          "path-metric-bound":[
            {
              "metric-type":"path-metric-hop",
              "upper-bound":10
            }
          ]
        }
      }
    }
  ]
}

```

The L2SM service can map it to a new TE resources in form of a VN or TE-tunnels.

Example 3: A VN (VN1) could be created before hand and then explicitly mapped to the L2VPN service as shown below.

```

<?xml version="1.0"?>
<l2vpn-svc xmlns="urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc">
  <vpn-services>
    <vpn-service>
      <vpn-id>VPN1</vpn-id>
      <te-service-mapping>
        <te-mapping>
          <map-type>select</map-type>
          <te>
            <vn>VN1</vn>
          </te>
        </te-mapping>
      </te-service-mapping>
    </vpn-service>
  </vpn-services>
</l2vpn-svc>

```

Example 4: A VPN service may want different optimization criteria for some of its sites. The template does not allow for such a case but it can be achieved by creating the TE resources separately and then mapping them to the service.

Appendix B. Out of Scope

Scheduling is currently out of scope, although an operator could use their own scheduling mechanism on top of this YANG model. In future augmentations to this model might also be designed to integrate scheduling and calendaring.

Note that the mechanism to map traffic (for example the enterprise customer can tell, the traffic from source X on port Y should go on a path with delay less than Z) can be via local configuration or through a YANG model developed in the future (See one such attempt at [I-D.dhody-teas-te-traffic-yang]).

Appendix C. Contributor Addresses

Adrian Farrel
Old Dog Consulting

EMail: adrian@olddog.co.uk

Italo Busi
Huawei Technologies

EMail: Italo.Busi@huawei.com

Haomian Zheng
Huawei Technologies

EMail: zhenghaomian@huawei.com

Anton Snitser
Sedonasys

EMail: antons@sedonasys.com

SAMIER BARGUIL GIRALDO
Telefonica

EMail: samier.barguilgiraldo.ext@telefonica.com

Oscar Gonzalez de Dios
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

Carlo Perocchio
Ericsson

EMail: carlo.perocchio@ericsson.com

Kenichi Ogaki
KDDI
Email: ke-oogaki@kddi.com

Authors' Addresses

Young Lee (editor)
Samsung Electronics
Email: younglee.tx@gmail.com

Dhruv Dhody (editor)
Huawei Technologies
Divyashree Techno Park, Whitefield
Bangalore 560066
Karnataka
India
Email: dhruv.ietf@gmail.com

Giuseppe Fioccola
Huawei Technologies
Email: giuseppe.fioccola@huawei.com

Qin Wu (editor)
Huawei Technologies
Email: bill.wu@huawei.com

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

Jeff Tantsura
Microsoft
Email: jefftant.ietf@gmail.com

Network Working Group
Internet-Draft
Updates: 8776 (if approved)
Intended status: Standards Track
Expires: April 27, 2022

X. Liu
Volta Networks
I. Bryskin
Individual
V. Beeram
T. Saad
Juniper Networks
H. Shah
Ciena
O. Gonzalez de Dios
Telefonica
October 24, 2021

YANG Data Model for Layer 3 TE Topologies
draft-ietf-teas-yang-l3-te-topo-12

Abstract

This document defines a YANG data model for layer 3 traffic engineering topologies.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 27, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Tree Diagrams	3
2. Modeling Considerations for L3 TE Topologies	4
2.1. Relationship Between Layer 3 Unicast Topology and TE topology	5
2.2. Relationship Modeling	6
2.2.1. Topology Referencing	6
2.2.2. Node Referencing	7
2.2.3. Link Termination Point Referencing	8
2.2.4. Link Referencing	8
2.3. Topology Type Modeling	8
3. Packet TE YANG Types	8
4. Packet Switching Technology Extensions to TE Topologies	10
4.1. Technology Specific Link Attributes	10
4.2. Performance Metric	10
5. Complete Model Tree Structure	11
5.1. Layer 3 TE Topology Module	11
5.2. Packet Switching TE Topology Module	12
5.2.1. Network Types	12
5.2.2. Node Connectivity Matrix Attributes	12
5.2.3. Node Information Source	16
5.2.4. Node Local Link Connectivity	21
5.2.5. Link Template for Performance Metrics	25
5.2.6. Link for Performance Metrics	28
5.2.7. Link Information Source for Performance Metrics	30
5.2.8. Link Template for Packet-specific Attributes	32
5.2.9. Link for Packet-specific Attributes	32
5.2.10. Link Information Source for Packet-specific Attributes	33
5.2.11. TE Bandwidth for Packet-specific Technologies	33
6. YANG Modules	36
6.1. Layer 3 TE Topology Module	36
6.2. Packet TE YANG Types Module	40
6.3. Packet Switching TE Topology Module	54
7. IANA Considerations	69
8. Security Considerations	70
9. References	73
9.1. Normative References	73
9.2. Informative References	76

Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations	77
A.1. Layer 3 TE Topology State Module	77
A.2. Packet Switching TE Topology State Module	80
Appendix B. Data Tree Example	87
Authors' Addresses	97

1. Introduction

This document defines a YANG [RFC7950] data model for describing the relationship between a layer 3 unicast topology [RFC8346] and a Traffic Engineering (TE) topology [RFC8795].

When traffic engineering is enabled on a layer 3 unicast topology, there will be a corresponding TE topology. The TE topology may or may not be congruent with the layer 3 unicast topology. When such a congruent TE topology exists, there will be a one-to-one association between the one modeling element in the layer 3 unicast topology and another element in the TE topology. When such a congruent TE topology does not exist, the association will not be one-to-one. This YANG data model supports both cases.

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

- o augment
- o data model
- o data node

1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

2. Modeling Considerations for L3 TE Topologies

A layer 3 TE topology is a layer 3 unicast topology with additional TE capabilities enabled. [RFC8346] defines a YANG data model for layer 3 unicast topologies, consisting of two modules: `ietf-l3-unicast-topology` and `ietf-l3-unicast-topology-state`. The YANG data model defined in this document augments the YANG data model defined in [RFC8346]. This document specifies two YANG modules `ietf-l3-te-topology` and `ietf-l3-te-topology-state`, augmenting `ietf-l3-unicast-topology` and `ietf-l3-unicast-topology-state` respectively, to add additional TE capabilities. Such an augmentation relationship is shown in Figure 1 below.

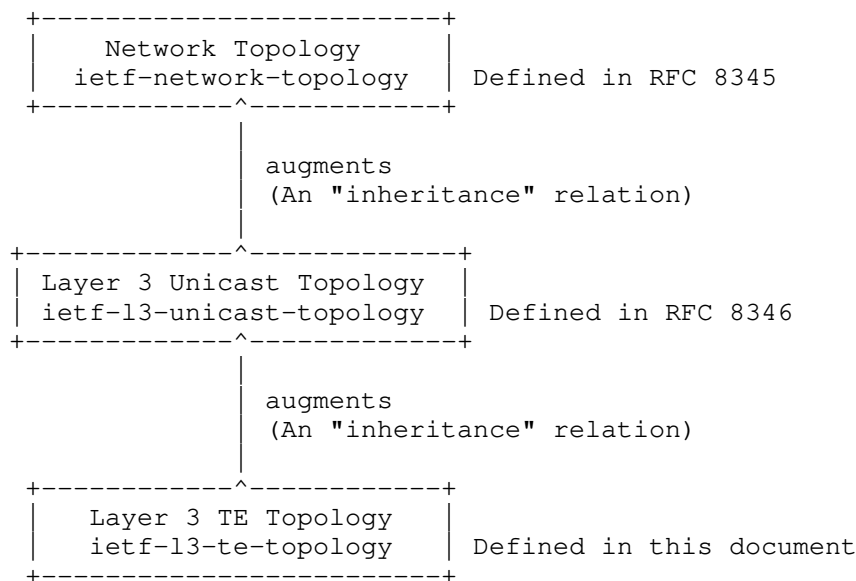


Figure 1: L3 TE Model Augmentation

Additionally, this document specifies two more YANG modules `ietf-te-topology-packet` and `ietf-te-topology-packet-state`, augmenting `ietf-te-topology` and `ietf-te-topology-state` respectively, to add additional attributes for TE packet data networks, as shown in Figure 2. Section 3 of this document describes these additional attributes in further details.

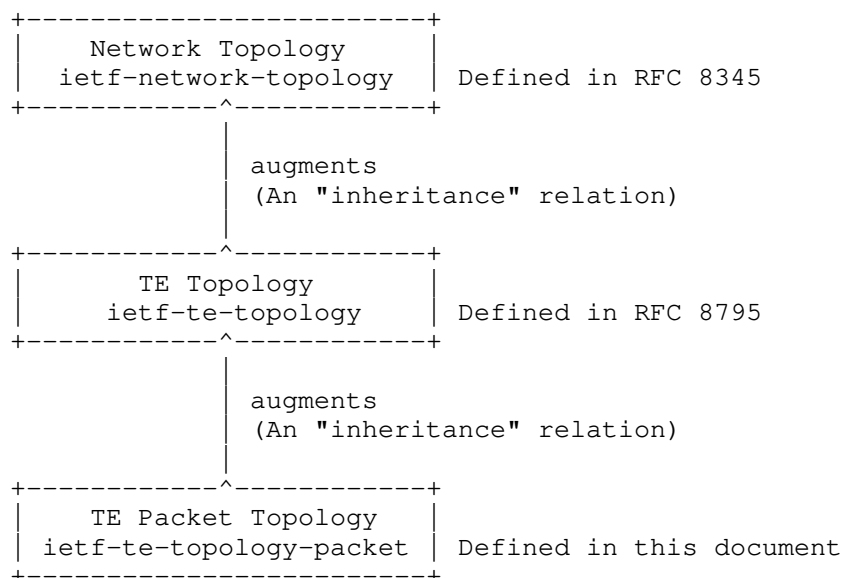


Figure 2: TE Packet Model Augmentation

2.1. Relationship Between Layer 3 Unicast Topology and TE topology

In general, the layer 3 unicast topology model specified in [RFC8346] and the TE topology model specified in [RFC8795] can be used independently. This document provides a method to use both together.

When traffic engineering is enabled on a layer 3 unicast topology, there will be a resulting layer 3 TE topology, which is modeled by the YANG modules defined in this document. A layer 3 TE topology augments a layer 3 unicast topology, so it inherits all the objects and properties of the base layer 3 unicast topology. In addition, in such a layer 3 TE topology, the objects that are inherited from the base layer 3 unicast topology to this layer 3 TE topology may be associated with the objects in one corresponding TE topology. Such associations are supported by the module `ietf-l3-te-topology` defined in this document.

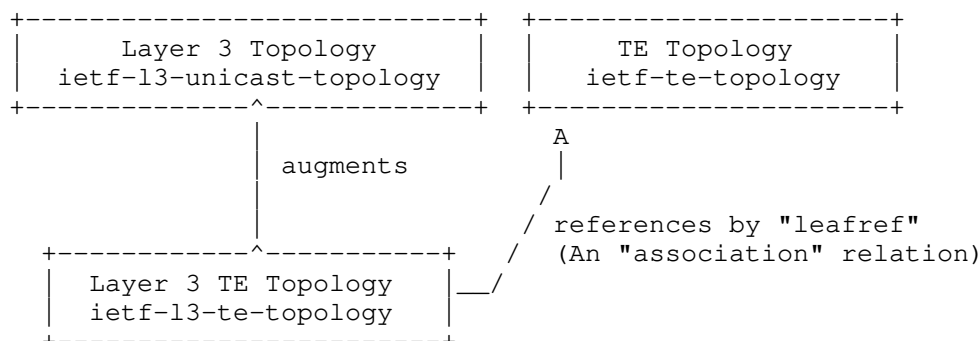


Figure 3: Model References

The properties of the relations between the objects in a layer 3 TE topology and the objects in the corresponding TE topology are:

- o The associations are between objects of the same class, i.e. node to node or link to link.
- o The multiplicity of such an association is: 0..1 to 0..1. An object in a layer 3 TE topology modeled by ietf-l3-te-topology may be associated with zero or one object in the corresponding TE topology.

2.2. Relationship Modeling

YANG data type leafref is used to model the association relationship between a layer 3 TE topology and a TE topology. YANG "must" statements are used to enforce that the referenced objects are in a topology of the proper type.

2.2.1. Topology Referencing

When TE is enabled on a layer 3 unicast topology, if the TE topology is not congruent with the layer 3 unicast topology, the layer 3 TE topology will have a reference to the corresponding TE topology. Such a reference is modeled as follows:

```

augment /nw:networks/nw:network/l3t:l3-topology-attributes:
  +--rw l3-te-topology-attributes
    +--rw network-ref?   -> /nw:networks/network/network-id
  
```

The above network-ref is a YANG data node of type leafref, used to indicate the network-id of the corresponding TE topology. The

leafref relationship is illustrated by the simplified data instances in the following diagram.

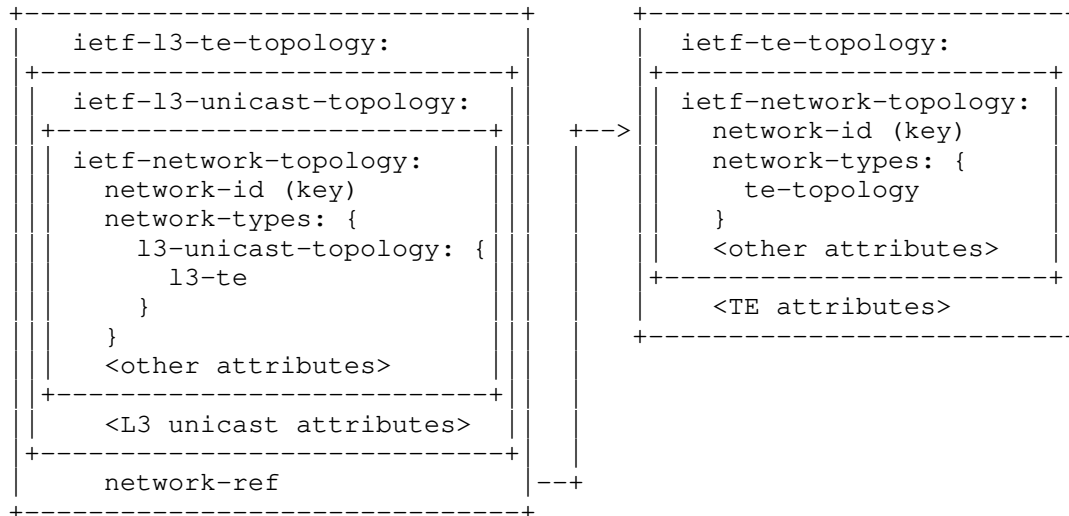


Figure 4: Topology Referencing

network-id defined in in [RFC8345] uniquely identifies a network topology instance of any type. As shown in the above diagram, the value of network-ref in the ietf-l3-te-topology instance matches the value of network-id in the ietf-te-topology instance. To ensure that the topology instance identified by this network-id is of type "te-topology", a "must" statement is defined in module ietf-l3-te-topology. Such "must" statements are also used in all the following leafref nodes in this section.

If the TE topology is congruent with the layer 3 unicast topology, the above reference can still be used to specified TE parameters defined in the TE topology model.

2.2.2. Node Referencing

When TE is enabled on a layer 3 unicast topology, if the TE topology is not congruent with the layer 3 unicast topology, a node in the layer 3 TE topology may have a reference to the corresponding node in the TE Topology. Such a reference is modeled as follows:

```
augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
  +--rw l3-te-node-attributes
    +--rw node-ref?      leafref
    +--rw network-ref?   -> /nw:networks/network/network-id
```

2.2.3. Link Termination Point Referencing

When TE is enabled on a layer 3 unicast topology, if the TE topology is not congruent with the layer 3 unicast topology, a link termination point in the layer 3 TE topology may have a reference to the corresponding link termination point in the TE Topology. Such a reference is modeled as follows:

```
augment /nw:networks/nw:network/nw:node/nt:termination-point
  /l3t:l3-termination-point-attributes:
  +--rw l3-te-tp-attributes
    +--rw tp-ref?      leafref
    +--rw node-ref?    leafref
    +--rw network-ref? -> /nw:networks/network/network-id
```

2.2.4. Link Referencing

When TE is enabled on a layer 3 unicast topology, if the TE topology is not congruent with the layer 3 unicast topology, a link in the layer 3 topology may have a reference to the corresponding link in the TE Topology. Such a reference is modeled as follows:

```
augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
  +--rw l3-te-link-attributes
    +--rw link-ref?    leafref
    +--rw network-ref? -> /nw:networks/network/network-id
```

2.3. Topology Type Modeling

A new topology type is defined in this document, to indicate a topology that is a layer 3 TE topology, which both inherits l3 unicast topology properties and is capable of TE.

```
augment /nw:networks/nw:network/nw:network-types
  /l3t:l3-unicast-topology:
  +--rw l3-te!
```

3. Packet TE YANG Types

This document updates [RFC8776] with a new revision of the module ietf-te-packet-types.

The module `ietf-te-packet-types` has been updated to add the following YANG identities, types and groupings which can be reused by MPLS-TE and other packet technologies YANG models:

`bandwidth-profile-type` This identity defines various bandwidth profiles specified by IETF and other organizations that may be used to limit bandwidth utilization of MPLS-TE LSPs.

`bandwidth-scientific-notation` This types represents the bandwidth in bit-per-second, using the scientific notation (e.g., 10e3).

`te-packet-path-bandwidth` This grouping defines the path bandwidth information and could be used in any Packet TE topology model (e.g., MPLS-TE) for the path bandwidth representation (e.g., the bandwidth of an MPLS-TE LSP). All the path and LSP bandwidth related sections in generic module, [RFC8776], need to be augmented with this grouping for the usage of Packet TE technologies. This grouping is also applicable to set up the MPLS-TE tunnel.

The Packet TE path bandwidth can be represented by a bandwidth profile as follow:

```
+--:(packet)
  +--rw bandwidth-profile-name?  string
  +--rw bandwidth-profile-type?  identityref
  +--rw cir?                     uint64
  +--rw eir?                     uint64
  +--rw cbs?                     uint64
  +--rw ebs?                     uint64
```

Other formats for the MPLS-TE path bandwidth are defined in [I-D.ietf-teas-yang-te-mpls] and they could be added in a future update of this document.

`te-packet-link-bandwidth` This grouping defines the link bandwidth information and could be used in any Packet TE topology model (e.g., MPLS-TE) for link bandwidth representation. All the link bandwidth related sections in generic module, [RFC8776], need to be augmented with this grouping for the usage of Packet TE technologies.

The Packet TE link bandwidth can be represented by a bandwidth expressed in scientific notation as follow:

```
+--:(packet)
  +--rw packet-bandwidth?  bandwidth-scientific-notation
```


4. Packet Switching Technology Extensions to TE Topologies

The technology agnostic TE Topology model is defined in [RFC8795], which is extended by this document to cover the Packet Switch Capable (PSC) technology [RFC3471] [RFC7074].

4.1. Technology Specific Link Attributes

The technology agnostic TE Topology model is augmented with packet switching specific link attributes:

```
augment /nw:networks/tet:te/tet:templates/tet:link-template
  /tet:te-link-attributes
  /tet:interface-switching-capability:
    +--rw packet-switch-capable
      +--rw minimum-lsp-bandwidth?  rt-types:bandwidth-ieee-float32
      +--rw interface-mtu?          uint16
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes
  /tet:interface-switching-capability:
    +--rw packet-switch-capable
      +--rw minimum-lsp-bandwidth?  rt-types:bandwidth-ieee-float32
      +--rw interface-mtu?          uint16
augment /nw:networks/nw:network/nt:link/tet:te
  /tet:information-source-entry
  /tet:interface-switching-capability:
    +--ro packet-switch-capable
      +--ro minimum-lsp-bandwidth?  rt-types:bandwidth-ieee-float32
      +--ro interface-mtu?          uint16
```

4.2. Performance Metric

[RFC7471], [RFC8570] and [RFC7823] specify TE performance metric parameters and their usage. The packet switching augmentations specified in this document support such a capability, which can be conditional enabled by a YANG feature "te-performance-metric".

```
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices:
    +--rw performance-metric
      +--rw measurement
      |      .....
      +--rw normality
      |      .....
      +--rw throttle
      |      .....
      +--rw .....
```

Such an augmentation has been applied to:

- o Connectivity matrices container
- o Connectivity matrix entry
- o Local link connectivities container
- o Local link connectivity entry
- o TE link attributes container in a TE link template
- o TE link attributes container in a TE link
- o Information source entry in a TE link

5. Complete Model Tree Structure

5.1. Layer 3 TE Topology Module

The model tree structure of the layer 3 TE topology module is as shown below:

```
module: ietf-l3-te-topology
  augment /nw:networks/nw:network/nw:network-types
    /l3t:l3-unicast-topology:
      +--rw l3-te!
  augment /nw:networks/nw:network/l3t:l3-topology-attributes:
    +--rw l3-te-topology-attributes
      +--rw network-ref? -> /nw:networks/network/network-id
  augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
    +--rw l3-te-node-attributes
      +--rw node-ref? leafref
      +--rw network-ref? -> /nw:networks/network/network-id
  augment /nw:networks/nw:network/nw:node/nt:termination-point
    /l3t:l3-termination-point-attributes:
      +--rw l3-te-tp-attributes
        +--rw tp-ref? leafref
        +--rw node-ref? leafref
        +--rw network-ref? -> /nw:networks/network/network-id
  augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
    +--rw l3-te-link-attributes
      +--rw link-ref? leafref
      +--rw network-ref? -> /nw:networks/network/network-id
```

5.2. Packet Switching TE Topology Module

This is an augmentation to base TE topology model.

5.2.1. Network Types

This augments the network types with a new network type for TE packet topologies.

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

5.2.2. Node Connectivity Matrix Attributes

This augments the node connectivity matrix attributes with configuration and states for performance metrics when the network type is packet.

```
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices:
    +---ro performance-metrics-one-way {te-performance-metric}?
    |   +---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 one-way-min-delay?                             uint32
    |   +---ro one-way-min-delay-normality?
    |       |
    |       te-types:performance-metrics-normality
    |   +---ro one-way-max-delay?                             uint32
    |   +---ro one-way-max-delay-normality?
    |       |
    |       te-types:performance-metrics-normality
    |   +---ro one-way-delay-variation?                       uint32
```

```

|   +---ro one-way-delay-variation-normality?
|   |       te-types:performance-metrics-normality
|   +---ro one-way-packet-loss?                decimal64
|   +---ro one-way-packet-loss-normality?
|   |       te-types:performance-metrics-normality
+---ro performance-metrics-two-way {te-performance-metric}?
|   +---ro two-way-delay?                        uint32
|   +---ro two-way-delay-normality?
|   |       te-types:performance-metrics-normality
|   +---ro two-way-min-delay?                    uint32
|   +---ro two-way-min-delay-normality?
|   |       te-types:performance-metrics-normality
|   +---ro two-way-max-delay?                    uint32
|   +---ro two-way-max-delay-normality?
|   |       te-types:performance-metrics-normality
|   +---ro two-way-delay-variation?              uint32
|   +---ro two-way-delay-variation-normality?
|   |       te-types:performance-metrics-normality
|   +---ro two-way-packet-loss?                decimal64
|   +---ro two-way-packet-loss-normality?
|   |       te-types:performance-metrics-normality
+---rw throttle {te-performance-metric}?
|   +---rw one-way-delay-offset?                uint32
|   +---rw measure-interval?                    uint32
|   +---rw advertisement-interval?              uint32
|   +---rw suppression-interval?                uint32
|   +---rw threshold-out
|   |   +---rw one-way-delay?                    uint32
|   |   +---rw one-way-residual-bandwidth?
|   |   |       rt-types:bandwidth-ieee-float32
|   |   +---rw one-way-available-bandwidth?
|   |   |       rt-types:bandwidth-ieee-float32
|   |   +---rw one-way-utilized-bandwidth?
|   |   |       rt-types:bandwidth-ieee-float32
|   |   +---rw two-way-delay?                    uint32
|   |   +---rw one-way-min-delay?                uint32
|   |   +---rw one-way-max-delay?                uint32
|   |   +---rw one-way-delay-variation?          uint32
|   |   +---rw one-way-packet-loss?              decimal64
|   |   +---rw two-way-min-delay?                uint32
|   |   +---rw two-way-max-delay?                uint32
|   |   +---rw two-way-delay-variation?          uint32
|   |   +---rw two-way-packet-loss?              decimal64
|   +---rw threshold-in
|   |   +---rw one-way-delay?                    uint32
|   |   +---rw one-way-residual-bandwidth?
|   |   |       rt-types:bandwidth-ieee-float32
|   |   +---rw one-way-available-bandwidth?

```

```

    |         rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
    |         rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64
+---rw threshold-accelerated-advertisement
    +---rw one-way-delay?              uint32
    +---rw one-way-residual-bandwidth?
        |         rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
    |         rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
    |         rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix:
+---ro performance-metrics-one-way {te-performance-metric}?
    +---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 one-way-min-delay?                               uint32
+---ro one-way-min-delay-normality?
|   te-types:performance-metrics-normality
+---ro one-way-max-delay?                               uint32
+---ro one-way-max-delay-normality?
|   te-types:performance-metrics-normality
+---ro one-way-delay-variation?                         uint32
+---ro one-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+---ro one-way-packet-loss?                             decimal64
+---ro one-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+---ro performance-metrics-two-way {te-performance-metric}?
+---ro two-way-delay?                                   uint32
+---ro two-way-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-min-delay?                               uint32
+---ro two-way-min-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-max-delay?                               uint32
+---ro two-way-max-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-delay-variation?                         uint32
+---ro two-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+---ro two-way-packet-loss?                             decimal64
+---ro two-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+---rw throttle {te-performance-metric}?
+---rw one-way-delay-offset?                             uint32
+---rw measure-interval?                               uint32
+---rw advertisement-interval?                         uint32
+---rw suppression-interval?                           uint32
+---rw threshold-out
|   +---rw one-way-delay?                               uint32
|   +---rw one-way-residual-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-available-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-utilized-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw two-way-delay?                               uint32
|   +---rw one-way-min-delay?                           uint32
|   +---rw one-way-max-delay?                           uint32
|   +---rw one-way-delay-variation?                     uint32
|   +---rw one-way-packet-loss?                         decimal64
|   +---rw two-way-min-delay?                           uint32
|   +---rw two-way-max-delay?                           uint32

```

```

|   +---rw two-way-delay-variation?          uint32
|   +---rw two-way-packet-loss?             decimal64
+---rw threshold-in
|   +---rw one-way-delay?                    uint32
|   +---rw one-way-residual-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw one-way-available-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw one-way-utilized-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw two-way-delay?                    uint32
|   +---rw one-way-min-delay?                uint32
|   +---rw one-way-max-delay?                uint32
|   +---rw one-way-delay-variation?          uint32
|   +---rw one-way-packet-loss?             decimal64
|   +---rw two-way-min-delay?                uint32
|   +---rw two-way-max-delay?                uint32
|   +---rw two-way-delay-variation?          uint32
|   +---rw two-way-packet-loss?             decimal64
+---rw threshold-accelerated-advertisement
|   +---rw one-way-delay?                    uint32
|   +---rw one-way-residual-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw one-way-available-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw one-way-utilized-bandwidth?
|       | rt-types:bandwidth-ieee-float32
|   +---rw two-way-delay?                    uint32
|   +---rw one-way-min-delay?                uint32
|   +---rw one-way-max-delay?                uint32
|   +---rw one-way-delay-variation?          uint32
|   +---rw one-way-packet-loss?             decimal64
|   +---rw two-way-min-delay?                uint32
|   +---rw two-way-max-delay?                uint32
|   +---rw two-way-delay-variation?          uint32
|   +---rw two-way-packet-loss?             decimal64

```

5.2.3. Node Information Source

This augments the node information source with states for performance metrics when the network type is packet.

```

augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices:
    +---ro performance-metrics-one-way {te-performance-metric}?
    |   +---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 one-way-min-delay?                               uint32
+---ro one-way-min-delay-normality?
|   te-types:performance-metrics-normality
+---ro one-way-max-delay?                               uint32
+---ro one-way-max-delay-normality?
|   te-types:performance-metrics-normality
+---ro one-way-delay-variation?                         uint32
+---ro one-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+---ro one-way-packet-loss?                             decimal64
+---ro one-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+---ro performance-metrics-two-way {te-performance-metric}?
+---ro two-way-delay?                                   uint32
+---ro two-way-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-min-delay?                               uint32
+---ro two-way-min-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-max-delay?                               uint32
+---ro two-way-max-delay-normality?
|   te-types:performance-metrics-normality
+---ro two-way-delay-variation?                         uint32
+---ro two-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+---ro two-way-packet-loss?                             decimal64
+---ro two-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+---ro throttle {te-performance-metric}?
+---ro one-way-delay-offset?                             uint32
+---ro measure-interval?                                uint32
+---ro advertisement-interval?                          uint32
+---ro suppression-interval?                            uint32
+---ro threshold-out

```



```

+---ro one-way-delay?                               uint32
+---ro one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro two-way-delay?                               uint32
+---ro one-way-min-delay?                           uint32
+---ro one-way-max-delay?                           uint32
+---ro one-way-delay-variation?                     uint32
+---ro one-way-packet-loss?                         decimal64
+---ro two-way-min-delay?                           uint32
+---ro two-way-max-delay?                           uint32
+---ro two-way-delay-variation?                     uint32
+---ro two-way-packet-loss?                         decimal64
+---ro threshold-in
+---ro one-way-delay?                               uint32
+---ro one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro two-way-delay?                               uint32
+---ro one-way-min-delay?                           uint32
+---ro one-way-max-delay?                           uint32
+---ro one-way-delay-variation?                     uint32
+---ro one-way-packet-loss?                         decimal64
+---ro two-way-min-delay?                           uint32
+---ro two-way-max-delay?                           uint32
+---ro two-way-delay-variation?                     uint32
+---ro two-way-packet-loss?                         decimal64
+---ro threshold-accelerated-advertisement
+---ro one-way-delay?                               uint32
+---ro one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---ro two-way-delay?                               uint32
+---ro one-way-min-delay?                           uint32
+---ro one-way-max-delay?                           uint32
+---ro one-way-delay-variation?                     uint32
+---ro one-way-packet-loss?                         decimal64
+---ro two-way-min-delay?                           uint32
+---ro two-way-max-delay?                           uint32

```

```

        +---ro two-way-delay-variation?          uint32
        +---ro two-way-packet-loss?             decimal64
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:connectivity-matrix:
+---ro performance-metrics-one-way {te-performance-metric}?
|   +---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 one-way-min-delay?                  uint32
|   +---ro one-way-min-delay-normality?
|       |   te-types:performance-metrics-normality
|   +---ro one-way-max-delay?                  uint32
|   +---ro one-way-max-delay-normality?
|       |   te-types:performance-metrics-normality
|   +---ro one-way-delay-variation?            uint32
|   +---ro one-way-delay-variation-normality?
|       |   te-types:performance-metrics-normality
|   +---ro one-way-packet-loss?                decimal64
|   +---ro one-way-packet-loss-normality?
|       |   te-types:performance-metrics-normality
+---ro performance-metrics-two-way {te-performance-metric}?
|   +---ro two-way-delay?                      uint32
|   +---ro two-way-delay-normality?
|       |   te-types:performance-metrics-normality
|   +---ro two-way-min-delay?                  uint32
|   +---ro two-way-min-delay-normality?
|       |   te-types:performance-metrics-normality
|   +---ro two-way-max-delay?                  uint32
|   +---ro two-way-max-delay-normality?
|       |   te-types:performance-metrics-normality
|   +---ro two-way-delay-variation?            uint32
|   +---ro two-way-delay-variation-normality?
|       |   te-types:performance-metrics-normality
|   +---ro two-way-packet-loss?                decimal64
|   +---ro two-way-packet-loss-normality?

```

```

|           te-types:performance-metrics-normality
+---ro throttle {te-performance-metric}?
|   +---ro one-way-delay-offset?                uint32
|   +---ro measure-interval?                    uint32
|   +---ro advertisement-interval?              uint32
|   +---ro suppression-interval?                uint32
+---ro threshold-out
|   +---ro one-way-delay?                        uint32
|   +---ro one-way-residual-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro two-way-delay?                            uint32
+---ro one-way-min-delay?                        uint32
+---ro one-way-max-delay?                        uint32
+---ro one-way-delay-variation?                 uint32
+---ro one-way-packet-loss?                     decimal64
+---ro two-way-min-delay?                        uint32
+---ro two-way-max-delay?                        uint32
+---ro two-way-delay-variation?                 uint32
+---ro two-way-packet-loss?                     decimal64
+---ro threshold-in
|   +---ro one-way-delay?                        uint32
|   +---ro one-way-residual-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro two-way-delay?                            uint32
+---ro one-way-min-delay?                        uint32
+---ro one-way-max-delay?                        uint32
+---ro one-way-delay-variation?                 uint32
+---ro one-way-packet-loss?                     decimal64
+---ro two-way-min-delay?                        uint32
+---ro two-way-max-delay?                        uint32
+---ro two-way-delay-variation?                 uint32
+---ro two-way-packet-loss?                     decimal64
+---ro threshold-accelerated-advertisement
|   +---ro one-way-delay?                        uint32
|   +---ro one-way-residual-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-available-bandwidth?
|       |           rt-types:bandwidth-ieee-float32
+---ro one-way-utilized-bandwidth?
|       |           rt-types:bandwidth-ieee-float32

```

```

+---ro two-way-delay?                uint32
+---ro one-way-min-delay?            uint32
+---ro one-way-max-delay?            uint32
+---ro one-way-delay-variation?      uint32
+---ro one-way-packet-loss?          decimal64
+---ro two-way-min-delay?            uint32
+---ro two-way-max-delay?            uint32
+---ro two-way-delay-variation?      uint32
+---ro two-way-packet-loss?          decimal64

```

5.2.4. Node Local Link Connectivity

This augments the node local link connectivity attributes with configuration and states for performance metrics when the network type is packet.

```

augment /nw:networks/nw:network/nw:node/tet:te
  /tet:tunnel-termination-point
  /tet:local-link-connectivities:
+---ro performance-metrics-one-way {te-performance-metric}?
|
|   +---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 one-way-min-delay?                uint32
|   +---ro one-way-min-delay-normality?
|   |   te-types:performance-metrics-normality
|   +---ro one-way-max-delay?                uint32
|   +---ro one-way-max-delay-normality?
|   |   te-types:performance-metrics-normality
|   +---ro one-way-delay-variation?          uint32
|   +---ro one-way-delay-variation-normality?
|   |   te-types:performance-metrics-normality
|   +---ro one-way-packet-loss?              decimal64
|   +---ro one-way-packet-loss-normality?

```

```

|         te-types:performance-metrics-normality
+--ro performance-metrics-two-way {te-performance-metric}?
|   +--ro two-way-delay?                               uint32
|   +--ro two-way-delay-normality?
|       |
|       |         te-types:performance-metrics-normality
|       +--ro two-way-min-delay?                       uint32
|       +--ro two-way-min-delay-normality?
|           |
|           |         te-types:performance-metrics-normality
|           +--ro two-way-max-delay?                   uint32
|           +--ro two-way-max-delay-normality?
|               |
|               |         te-types:performance-metrics-normality
|               +--ro two-way-delay-variation?         uint32
|               +--ro two-way-delay-variation-normality?
|                   |
|                   |         te-types:performance-metrics-normality
|                   +--ro two-way-packet-loss?         decimal64
|                   +--ro two-way-packet-loss-normality?
|                       |
|                       |         te-types:performance-metrics-normality
+--rw throttle {te-performance-metric}?
|   +--rw one-way-delay-offset?                         uint32
|   +--rw measure-interval?                             uint32
|   +--rw advertisement-interval?                       uint32
|   +--rw suppression-interval?                         uint32
|   +--rw threshold-out
|       |
|       |   +--rw one-way-delay?                       uint32
|       |   +--rw one-way-residual-bandwidth?
|       |       |
|       |       |         rt-types:bandwidth-ieee-float32
|       |       +--rw one-way-available-bandwidth?
|       |           |
|       |           |         rt-types:bandwidth-ieee-float32
|       |       +--rw one-way-utilized-bandwidth?
|       |           |
|       |           |         rt-types:bandwidth-ieee-float32
|       |       +--rw two-way-delay?                   uint32
|       |       +--rw one-way-min-delay?               uint32
|       |       +--rw one-way-max-delay?               uint32
|       |       +--rw one-way-delay-variation?         uint32
|       |       +--rw one-way-packet-loss?             decimal64
|       |       +--rw two-way-min-delay?               uint32
|       |       +--rw two-way-max-delay?               uint32
|       |       +--rw two-way-delay-variation?         uint32
|       |       +--rw two-way-packet-loss?             decimal64
|   +--rw threshold-in
|       |
|       |   +--rw one-way-delay?                       uint32
|       |   +--rw one-way-residual-bandwidth?
|       |       |
|       |       |         rt-types:bandwidth-ieee-float32
|       |       +--rw one-way-available-bandwidth?
|       |           |
|       |           |         rt-types:bandwidth-ieee-float32
|       |       +--rw one-way-utilized-bandwidth?
|       |           |
|       |           |         rt-types:bandwidth-ieee-float32
|       |       +--rw two-way-delay?                   uint32

```

```

+---rw one-way-min-delay?                uint32
+---rw one-way-max-delay?                uint32
+---rw one-way-delay-variation?          uint32
+---rw one-way-packet-loss?              decimal64
+---rw two-way-min-delay?                uint32
+---rw two-way-max-delay?                uint32
+---rw two-way-delay-variation?          uint32
+---rw two-way-packet-loss?              decimal64
+---rw threshold-accelerated-advertisement
+---rw one-way-delay?                    uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                    uint32
+---rw one-way-min-delay?                uint32
+---rw one-way-max-delay?                uint32
+---rw one-way-delay-variation?          uint32
+---rw one-way-packet-loss?              decimal64
+---rw two-way-min-delay?                uint32
+---rw two-way-max-delay?                uint32
+---rw two-way-delay-variation?          uint32
+---rw two-way-packet-loss?              decimal64
augment /nw:networks/nw:network/nw:node/tet:te
|   /tet:tunnel-termination-point
|   /tet:local-link-connectivities
|   /tet:local-link-connectivity:
+---ro performance-metrics-one-way {te-performance-metric}?
|   +---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 one-way-min-delay?            uint32
|   +---ro one-way-min-delay-normality?
|   |   te-types:performance-metrics-normality

```

```

+--ro one-way-max-delay?                               uint32
+--ro one-way-max-delay-normality?
|   te-types:performance-metrics-normality
+--ro one-way-delay-variation?                           uint32
+--ro one-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+--ro one-way-packet-loss?                               decimal64
+--ro one-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+--ro performance-metrics-two-way {te-performance-metric}?
+--ro two-way-delay?                                     uint32
+--ro two-way-delay-normality?
|   te-types:performance-metrics-normality
+--ro two-way-min-delay?                                 uint32
+--ro two-way-min-delay-normality?
|   te-types:performance-metrics-normality
+--ro two-way-max-delay?                                 uint32
+--ro two-way-max-delay-normality?
|   te-types:performance-metrics-normality
+--ro two-way-delay-variation?                           uint32
+--ro two-way-delay-variation-normality?
|   te-types:performance-metrics-normality
+--ro two-way-packet-loss?                               decimal64
+--ro two-way-packet-loss-normality?
|   te-types:performance-metrics-normality
+--rw throttle {te-performance-metric}?
+--rw one-way-delay-offset?                             uint32
+--rw measure-interval?                                 uint32
+--rw advertisement-interval?                           uint32
+--rw suppression-interval?                             uint32
+--rw threshold-out
|   +--rw one-way-delay?                                 uint32
|   +--rw one-way-residual-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +--rw one-way-available-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +--rw one-way-utilized-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +--rw two-way-delay?                                 uint32
|   +--rw one-way-min-delay?                             uint32
|   +--rw one-way-max-delay?                             uint32
|   +--rw one-way-delay-variation?                       uint32
|   +--rw one-way-packet-loss?                           decimal64
|   +--rw two-way-min-delay?                             uint32
|   +--rw two-way-max-delay?                             uint32
|   +--rw two-way-delay-variation?                       uint32
|   +--rw two-way-packet-loss?                           decimal64
+--rw threshold-in

```

```

+---rw one-way-delay?                               uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                               uint32
+---rw one-way-min-delay?                           uint32
+---rw one-way-max-delay?                           uint32
+---rw one-way-delay-variation?                     uint32
+---rw one-way-packet-loss?                         decimal64
+---rw two-way-min-delay?                           uint32
+---rw two-way-max-delay?                           uint32
+---rw two-way-delay-variation?                     uint32
+---rw two-way-packet-loss?                         decimal64
+---rw threshold-accelerated-advertisement
+---rw one-way-delay?                               uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                               uint32
+---rw one-way-min-delay?                           uint32
+---rw one-way-max-delay?                           uint32
+---rw one-way-delay-variation?                     uint32
+---rw one-way-packet-loss?                         decimal64
+---rw two-way-min-delay?                           uint32
+---rw two-way-max-delay?                           uint32
+---rw two-way-delay-variation?                     uint32
+---rw two-way-packet-loss?                         decimal64

```

5.2.5. Link Template for Performance Metrics

This augments the link template attributes with configuration and states for performance metrics when the network type is packet.

```

augment /nw:networks/tet:te/tet:templates/tet:link-template
  /tet:te-link-attributes:
+---ro performance-metrics-one-way {te-performance-metric}?
|   +---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 one-way-min-delay?                               uint32
+--ro one-way-min-delay-normality?
|         te-types:performance-metrics-normality
+--ro one-way-max-delay?                               uint32
+--ro one-way-max-delay-normality?
|         te-types:performance-metrics-normality
+--ro one-way-delay-variation?                         uint32
+--ro one-way-delay-variation-normality?
|         te-types:performance-metrics-normality
+--ro one-way-packet-loss?                             decimal64
+--ro one-way-packet-loss-normality?
|         te-types:performance-metrics-normality
+--ro performance-metrics-two-way {te-performance-metric}?
|         +--ro two-way-delay?                         uint32
|         +--ro two-way-delay-normality?
|         |         te-types:performance-metrics-normality
|         +--ro two-way-min-delay?                     uint32
|         +--ro two-way-min-delay-normality?
|         |         te-types:performance-metrics-normality
|         +--ro two-way-max-delay?                     uint32
|         +--ro two-way-max-delay-normality?
|         |         te-types:performance-metrics-normality
|         +--ro two-way-delay-variation?               uint32
|         +--ro two-way-delay-variation-normality?
|         |         te-types:performance-metrics-normality
|         +--ro two-way-packet-loss?                   decimal64
|         +--ro two-way-packet-loss-normality?
|         |         te-types:performance-metrics-normality
+--rw throttle {te-performance-metric}?
|         +--rw one-way-delay-offset?                  uint32
|         +--rw measure-interval?                     uint32
|         +--rw advertisement-interval?                uint32
|         +--rw suppression-interval?                  uint32
|         +--rw threshold-out
|         |         +--rw one-way-delay?                uint32
|         |         +--rw one-way-residual-bandwidth?
|         |         |         rt-types:bandwidth-ieee-float32

```

```

+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64
+---rw threshold-in
+---rw one-way-delay?                uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64
+---rw threshold-accelerated-advertisement
+---rw one-way-delay?                uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64

```

5.2.6. Link for Performance Metrics

This augments the link attributes with configuration and states for performance metrics when the network type is packet.

```

augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes:
    +--ro performance-metrics-one-way {te-performance-metric}?
      +--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 one-way-min-delay?                             uint32
      +--ro one-way-min-delay-normality?
        | te-types:performance-metrics-normality
      +--ro one-way-max-delay?                             uint32
      +--ro one-way-max-delay-normality?
        | te-types:performance-metrics-normality
      +--ro one-way-delay-variation?                       uint32
      +--ro one-way-delay-variation-normality?
        | te-types:performance-metrics-normality
      +--ro one-way-packet-loss?                           decimal64
      +--ro one-way-packet-loss-normality?
        | te-types:performance-metrics-normality
    +--ro performance-metrics-two-way {te-performance-metric}?
      +--ro two-way-delay?                                 uint32
      +--ro two-way-delay-normality?
        | te-types:performance-metrics-normality
      +--ro two-way-min-delay?                             uint32
      +--ro two-way-min-delay-normality?
        | te-types:performance-metrics-normality
      +--ro two-way-max-delay?                             uint32
      +--ro two-way-max-delay-normality?
        | te-types:performance-metrics-normality
      +--ro two-way-delay-variation?                       uint32
      +--ro two-way-delay-variation-normality?

```

```

|         te-types:performance-metrics-normality
+---ro two-way-packet-loss?                decimal64
|         te-types:performance-metrics-normality
+---rw throttle {te-performance-metric}?
+---rw one-way-delay-offset?                uint32
+---rw measure-interval?                   uint32
+---rw advertisement-interval?              uint32
+---rw suppression-interval?                uint32
+---rw threshold-out
|   +---rw one-way-delay?                   uint32
|   +---rw one-way-residual-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-available-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-utilized-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw two-way-delay?                   uint32
|   +---rw one-way-min-delay?                uint32
|   +---rw one-way-max-delay?                uint32
|   +---rw one-way-delay-variation?          uint32
|   +---rw one-way-packet-loss?              decimal64
|   +---rw two-way-min-delay?                uint32
|   +---rw two-way-max-delay?                uint32
|   +---rw two-way-delay-variation?          uint32
|   +---rw two-way-packet-loss?              decimal64
+---rw threshold-in
|   +---rw one-way-delay?                   uint32
|   +---rw one-way-residual-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-available-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw one-way-utilized-bandwidth?
|   |   rt-types:bandwidth-ieee-float32
|   +---rw two-way-delay?                   uint32
|   +---rw one-way-min-delay?                uint32
|   +---rw one-way-max-delay?                uint32
|   +---rw one-way-delay-variation?          uint32
|   +---rw one-way-packet-loss?              decimal64
|   +---rw two-way-min-delay?                uint32
|   +---rw two-way-max-delay?                uint32
|   +---rw two-way-delay-variation?          uint32
|   +---rw two-way-packet-loss?              decimal64
+---rw threshold-accelerated-advertisement
+---rw one-way-delay?                       uint32
+---rw one-way-residual-bandwidth?
|   rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?

```

```

|         rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
|         rt-types:bandwidth-ieee-float32
+---rw two-way-delay?                uint32
+---rw one-way-min-delay?            uint32
+---rw one-way-max-delay?            uint32
+---rw one-way-delay-variation?      uint32
+---rw one-way-packet-loss?          decimal64
+---rw two-way-min-delay?            uint32
+---rw two-way-max-delay?            uint32
+---rw two-way-delay-variation?      uint32
+---rw two-way-packet-loss?          decimal64

```

5.2.7. Link Information Source for Performance Metrics

This augments the link information source with states for performance metrics when the network type is packet.

```

augment /nw:networks/nw:network/nt:link/tet:te
  /tet:information-source-entry:
+--ro performance-metrics-one-way {te-performance-metric}?
|   +--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 one-way-min-delay?            uint32
|   +--ro one-way-min-delay-normality?
|   |   te-types:performance-metrics-normality
|   +--ro one-way-max-delay?            uint32
|   +--ro one-way-max-delay-normality?
|   |   te-types:performance-metrics-normality
|   +--ro one-way-delay-variation?      uint32
|   +--ro one-way-delay-variation-normality?
|   |   te-types:performance-metrics-normality
|   +--ro one-way-packet-loss?          decimal64

```

```

    +---ro one-way-packet-loss-normality?
        |
        |   te-types:performance-metrics-normality
+---ro performance-metrics-two-way {te-performance-metric}?
    +---ro two-way-delay?                               uint32
    +---ro two-way-delay-normality?
        |
        |   te-types:performance-metrics-normality
    +---ro two-way-min-delay?                             uint32
    +---ro two-way-min-delay-normality?
        |
        |   te-types:performance-metrics-normality
    +---ro two-way-max-delay?                             uint32
    +---ro two-way-max-delay-normality?
        |
        |   te-types:performance-metrics-normality
    +---ro two-way-delay-variation?                       uint32
    +---ro two-way-delay-variation-normality?
        |
        |   te-types:performance-metrics-normality
    +---ro two-way-packet-loss?                           decimal64
    +---ro two-way-packet-loss-normality?
        |
        |   te-types:performance-metrics-normality
+---ro throttle {te-performance-metric}?
    +---ro one-way-delay-offset?                         uint32
    +---ro measure-interval?                             uint32
    +---ro advertisement-interval?                       uint32
    +---ro suppression-interval?                         uint32
    +---ro threshold-out
        |
        |   +---ro one-way-delay?                         uint32
        |   +---ro one-way-residual-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32
        |   +---ro one-way-available-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32
        |   +---ro one-way-utilized-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32
        |   +---ro two-way-delay?                         uint32
        |   +---ro one-way-min-delay?                     uint32
        |   +---ro one-way-max-delay?                     uint32
        |   +---ro one-way-delay-variation?               uint32
        |   +---ro one-way-packet-loss?                   decimal64
        |   +---ro two-way-min-delay?                     uint32
        |   +---ro two-way-max-delay?                     uint32
        |   +---ro two-way-delay-variation?               uint32
        |   +---ro two-way-packet-loss?                   decimal64
    +---ro threshold-in
        |
        |   +---ro one-way-delay?                         uint32
        |   +---ro one-way-residual-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32
        |   +---ro one-way-available-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32
        |   +---ro one-way-utilized-bandwidth?
        |       |
        |       |   rt-types:bandwidth-ieee-float32

```

```

| +--ro two-way-delay?                uint32
| +--ro one-way-min-delay?            uint32
| +--ro one-way-max-delay?            uint32
| +--ro one-way-delay-variation?      uint32
| +--ro one-way-packet-loss?          decimal64
| +--ro two-way-min-delay?            uint32
| +--ro two-way-max-delay?            uint32
| +--ro two-way-delay-variation?      uint32
| +--ro two-way-packet-loss?          decimal64
+--ro threshold-accelerated-advertisement
  +--ro one-way-delay?                uint32
  +--ro one-way-residual-bandwidth?
    | rt-types:bandwidth-ieee-float32
  +--ro one-way-available-bandwidth?
    | rt-types:bandwidth-ieee-float32
  +--ro one-way-utilized-bandwidth?
    | rt-types:bandwidth-ieee-float32
  +--ro two-way-delay?                uint32
  +--ro one-way-min-delay?            uint32
  +--ro one-way-max-delay?            uint32
  +--ro one-way-delay-variation?      uint32
  +--ro one-way-packet-loss?          decimal64
  +--ro two-way-min-delay?            uint32
  +--ro two-way-max-delay?            uint32
  +--ro two-way-delay-variation?      uint32
  +--ro two-way-packet-loss?          decimal64

```

5.2.8. Link Template for Packet-specific Attributes

This augments the link template attributes with configuration and states for packet-specific attributes when the network type is packet.

```

augment /nw:networks/tet:te/tet:templates/tet:link-template
  /tet:te-link-attributes
    /tet:interface-switching-capability:
      +--rw packet-switch-capable
      +--rw minimum-lsp-bandwidth?   rt-types:bandwidth-ieee-float32
      +--rw interface-mtu?           uint16

```

5.2.9. Link for Packet-specific Attributes

This augments the link attributes with configuration and states for packet-specific attributes when the network type is packet.

```

augment /nw:networks/nw:network/nt:link/tet:te
  /tet:te-link-attributes
  /tet:interface-switching-capability:
  +--rw packet-switch-capable
  +--rw minimum-lsp-bandwidth?   rt-types:bandwidth-ieee-float32
  +--rw interface-mtu?           uint16

```

5.2.10. Link Information Source for Packet-specific Attributes

This augments the link information source with states for packet-specific attributes when the network type is packet.

```

augment /nw:networks/nw:network/nt:link/tet:te
  /tet:information-source-entry
  /tet:interface-switching-capability:
  +--ro packet-switch-capable
  +--ro minimum-lsp-bandwidth?   rt-types:bandwidth-ieee-float32
  +--ro interface-mtu?           uint16

```

5.2.11. TE Bandwidth for Packet-specific Technologies

These augmentations specify TE bandwidth for packet-specific technologies.

```

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:
  +--:(packet)
  +--rw bandwidth-profile-name?   string
  +--rw bandwidth-profile-type?   identityref
  +--rw cir?                      uint64
  +--rw eir?                      uint64
  +--rw cbs?                      uint64
  +--rw ebs?                      uint64
augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:path-constraints/tet:te-bandwidth/tet:technology:
  +--:(packet)
  +--rw packet-bandwidth?         bandwidth-scientific-notation
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:
  +--:(packet)

```



```

    +---rw packet-bandwidth?    bandwidth-scientific-notation
augment /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices
    /tet:path-constraints/tet:te-bandwidth/tet:technology:
+---:(packet)
    +---ro packet-bandwidth?    bandwidth-scientific-notation
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:
+---:(packet)
    +---ro packet-bandwidth?    bandwidth-scientific-notation
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:
+---:(packet)
    +---rw packet-bandwidth?    bandwidth-scientific-notation
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:
+---:(packet)
    +---rw packet-bandwidth?    bandwidth-scientific-notation
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:
+---:(packet)
    +---rw packet-bandwidth?    bandwidth-scientific-notation
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:
+---:(packet)
    +---rw bandwidth-profile-name?    string
    +---rw bandwidth-profile-type?    identityref
    +---rw cir?                        uint64
    +---rw eir?                        uint64
    +---rw cbs?                        uint64
    +---rw ebs?                        uint64
augment /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes/tet:max-link-bandwidth
    /tet:te-bandwidth/tet:technology:
+---:(packet)
    +---rw packet-bandwidth?    bandwidth-scientific-notation
augment /nw:networks/nw:network/nt:link/tet:te

```

```

        /tet:te-link-attributes/tet:max-resv-link-bandwidth
        /tet:te-bandwidth/tet:technology:
    +---: (packet)
        +---rw packet-bandwidth?    bandwidth-scientific-notation
    augment /nw:networks/nw:network/nt:link/tet:te
        /tet:te-link-attributes/tet:unreserved-bandwidth
        /tet:te-bandwidth/tet:technology:
    +---: (packet)
        +---rw packet-bandwidth?    bandwidth-scientific-notation
    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:
    +---: (packet)
        +---ro bandwidth-profile-name?    string
        +---ro bandwidth-profile-type?    identityref
        +---ro cir?                        uint64
        +---ro eir?                        uint64
        +---ro cbs?                        uint64
        +---ro ebs?                        uint64
    augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:max-link-bandwidth
        /tet:te-bandwidth/tet:technology:
    +---: (packet)
        +---ro packet-bandwidth?    bandwidth-scientific-notation
    augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:max-resv-link-bandwidth
        /tet:te-bandwidth/tet:technology:
    +---: (packet)
        +---ro packet-bandwidth?    bandwidth-scientific-notation
    augment /nw:networks/nw:network/nt:link/tet:te
        /tet:information-source-entry/tet:unreserved-bandwidth
        /tet:te-bandwidth/tet:technology:
    +---: (packet)
        +---ro packet-bandwidth?    bandwidth-scientific-notation
    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:
    +---: (packet)
        +---rw bandwidth-profile-name?    string
        +---rw bandwidth-profile-type?    identityref
        +---rw cir?                        uint64
        +---rw eir?                        uint64
        +---rw cbs?                        uint64
        +---rw ebs?                        uint64
    augment /nw:networks/tet:te/tet:templates/tet:link-template
        /tet:te-link-attributes/tet:max-link-bandwidth

```

```

        /tet:te-bandwidth/tet:technology:
    +--:(packet)
      +--rw packet-bandwidth?    bandwidth-scientific-notation
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:max-resv-link-bandwidth
    /tet:te-bandwidth/tet:technology:
    +--:(packet)
      +--rw packet-bandwidth?    bandwidth-scientific-notation
augment /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes/tet:unreserved-bandwidth
    /tet:te-bandwidth/tet:technology:
    +--:(packet)
      +--rw packet-bandwidth?    bandwidth-scientific-notation

```

6. YANG Modules

6.1. Layer 3 TE Topology Module

This module references [RFC8345], [RFC8346], and [RFC8795].

```

<CODE BEGINS> file "ietf-l3-te-topology@2020-05-03.yang"
module ietf-l3-te-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3-te-topology";
  prefix "l3tet";

  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";
  }
  import ietf-l3-unicast-topology {
    prefix "l3t";
    reference "RFC 8346: A YANG Data Model for Layer 3 Topologies";
  }
  import ietf-te-topology {
    prefix "tet";
    reference
      "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
       Engineering (TE) 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@cisco.com>>

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

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

description

"YANG data model for representing and manipulating Layer 3 TE
Topologies.

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>).

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";

revision 2020-05-03 {

description

"Initial revision";

reference "RFC XXXX: YANG Data Model for Layer 3 TE Topologies";

}

```
grouping l3-te-topology-type {
  description
    "Identifies the L3 TE topology type.";
  container l3-te {
    presence "Indicates L3 TE Topology";
    description
      "Its presence identifies the L3 TE topology type.";
  }
}

augment "/nw:networks/nw:network/nw:network-types/"
+ "l3t:l3-unicast-topology" {
  description
    "Defines the L3 TE topology type.";
  uses l3-te-topology-type;
}

augment "/nw:networks/nw:network/l3t:l3-topology-attributes" {
  when "../nw:network-types/l3t:l3-unicast-topology/l3tet:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment topology configuration";
  uses l3-te-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node/l3t:l3-node-attributes" {
  when "../..nw:network-types/l3t:l3-unicast-topology/"
+ "l3tet:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment node configuration";
  uses l3-te-node-attributes;
}

augment "/nw:networks/nw:network/nw:node/nt:termination-point/"
+ "l3t:l3-termination-point-attributes" {
  when "../..nw:network-types/l3t:l3-unicast-topology/"
+ "l3tet:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment termination point configuration";
  uses l3-te-tp-attributes;
}
```

```
}

augment "/nw:networks/nw:network/nt:link/l3t:l3-link-attributes" {
  when "../..nw:network-types/l3t:l3-unicast-topology/"
    + "l3tet:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment link configuration";
  uses l3-te-link-attributes;
}

grouping l3-te-topology-attributes {
  description
    "L3 TE topology scope attributes";
  container l3-te-topology-attributes {
    must "/nw:networks/nw:network"
      + "[nw:network-id = current()/network-ref]/nw:network-types/"
      + "tet:te-topology" {
      error-message
        "The referenced network must be a TE topology.";
      description
        "The referenced network must be a TE topology.";
    }
    description
      "Containing TE topology references";
    uses nw:network-ref;
  } // l3-te-topology-attributes
} // l3-te-topology-attributes

grouping l3-te-node-attributes {
  description
    "L3 TE node scope attributes";
  container l3-te-node-attributes {
    must "/nw:networks/nw:network"
      + "[nw:network-id = current()/network-ref]/nw:network-types/"
      + "tet:te-topology" {
      error-message
        "The referenced network must be a TE topology.";
      description
        "The referenced network must be a TE topology.";
    }
    description
      "Containing TE node references";
    uses nw:node-ref;
  } // l3-te
} // l3-te-node-attributes
```

```
grouping l3-te-tp-attributes {
  description
    "L3 TE termination point scope attributes";
  container l3-te-tp-attributes {
    must "/nw:networks/nw:network"
      + "[nw:network-id = current()/network-ref]/nw:network-types/"
      + "tet:te-topology" {
      error-message
        "The referenced network must be a TE topology.";
      description
        "The referenced network must be a TE topology.";
    }
    description
      "Containing TE termination point references";
    uses nt:tp-ref;
  } // l3-te
} // l3-te-tp-attributes

grouping l3-te-link-attributes {
  description
    "L3 TE link scope attributes";
  container l3-te-link-attributes {
    must "/nw:networks/nw:network"
      + "[nw:network-id = current()/network-ref]/nw:network-types/"
      + "tet:te-topology" {
      error-message
        "The referenced network must be a TE topology.";
      description
        "The referenced network must be a TE topology.";
    }
    description
      "Containing TE link references";
    uses nt:link-ref;
  }
} // l3-te-link-attributes
}
<CODE ENDS>
```

6.2. Packet TE YANG Types Module

This module references [RFC2697], [RFC2698], [RFC4115], [RFC4124], [RFC4125], [RFC4126], [RFC4127], [RFC5481], [RFC7471], [RFC7823], [RFC8570], [RFC8776], [IEEE754], and [ISO-IEC-C99]

<CODE BEGINS> file "ietf-te-packet-types@2020-06-10.yang"

```
module ietf-te-packet-types {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-packet-types";
  prefix te-packet-types;

  /* Import TE generic types */

  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

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

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

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

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

    Editor:   Igor Bryskin
              <mailto:i_bryskin@yahoo.com>";
  description
    "This YANG module contains a collection of generally useful YANG
    data type definitions specific to MPLS TE.  The model fully
    conforms to the Network Management Datastore Architecture
    (NMDA).

    Copyright (c) 2021 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 8776; see the RFC itself for full legal notices.";

```
revision 2021-10-24 {
  description
    "Added common TE packet identities:
     - bandwidth-profile-type.

     Added common TE packet types:
     - bandwidth-scientific-notation.

     Added common TE packet groupings:
     - te-packet-path-bandwidth;
     - te-packet-link-bandwidth.";
  reference
    "draft-busizheng-teas-yang-te-mpls-topology";
}

revision 2020-06-10 {
  description
    "Latest revision of TE MPLS types.";
  reference
    "RFC 8776: Common YANG Data Types for Traffic Engineering";
}

/*
 * Identities
 */

identity bandwidth-profile-type {
  description
    "Bandwidth Profile Types";
}

identity mef-10-bwp {
  base bandwidth-profile-type;
  description
    "MEF 10 Bandwidth Profile";
}

identity rfc-2697-bwp {
  base bandwidth-profile-type;
  description
    "RFC 2697 Bandwidth Profile";
}
```

```
identity rfc-2698-bwp {
  base bandwidth-profile-type;
  description
    "RFC 2698 Bandwidth Profile";
}

identity rfc-4115-bwp {
  base bandwidth-profile-type;
  description
    "RFC 4115 Bandwidth Profile";
}

/*
 * Typedefs
 */

typedef te-bandwidth-requested-type {
  type enumeration {
    enum specified {
      description
        "Bandwidth is explicitly specified.";
    }
    enum auto {
      description
        "Bandwidth is automatically computed.";
    }
  }
  description
    "Enumerated type for specifying whether bandwidth is
    explicitly specified or automatically computed.";
}

typedef te-class-type {
  type uint8;
  description
    "Diffserv-TE Class-Type. Defines a set of Traffic Trunks
    crossing a link that is governed by a specific set of
    bandwidth constraints. Class-Type is used for the purposes
    of link bandwidth allocation, constraint-based routing, and
    admission control.";
  reference
    "RFC 4124: Protocol Extensions for Support of Diffserv-aware
    MPLS Traffic Engineering";
}

typedef bc-type {
  type uint8 {
    range "0..7";
  }
}
```

```
    }
    description
      "Diffserv-TE bandwidth constraints as defined in RFC 4124.";
    reference
      "RFC 4124: Protocol Extensions for Support of Diffserv-aware
        MPLS Traffic Engineering";
  }

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

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

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

  typedef bandwidth-scientific-notation {
    type string {
      pattern
        '0(\.0?)?([eE](\+)?0?)?|'
        + '[1-9](\.[0-9]{0,6})?[eE](\+)?(9[0-6]|[1-8][0-9]|0?[0-9])?';
    }
    units "bps";
    description
      "Bandwidth values, expressed using the scientific notation
        in bits per second.
        The encoding format is the external decimal-significant
        character sequences specified in IEEE 754 and ISO/IEC C99
        for 32-bit decimal floating-point numbers:
        (-1)**(S) * 10**(Exponent) * (Significant),
        where Significant uses 7 digits.
        An implementation for this representation may use decimal32
        or binary32. The range of the Exponent is from -95 to +96
        for decimal32, and from -38 to +38 for binary32.
        As a bandwidth value, the format is restricted to be
```

```
normalized, non-negative, and non-fraction:
n.dddddde{+}dd, N.DDDDDDE{+}DD, 0e0 or 0E0,
where 'd' and 'D' are decimal digits; 'n' and 'N' are
non-zero decimal digits; 'e' and 'E' indicate a power of ten.
Some examples are 0e0, 1e10, and 9.953e9.";
reference
  "IEEE Std 754-2008: IEEE Standard for Floating-Point
  Arithmetic.
  ISO/IEC C99: Information technology - Programming
  Languages - C.";
}

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

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

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

identity bc-model-type {
  description
    "Base identity for the Diffserv-TE Bandwidth Constraints
    Model type.";
  reference
    "RFC 4124: Protocol Extensions for Support of Diffserv-aware
    MPLS Traffic Engineering";
}

identity bc-model-rdm {
  base bc-model-type;
  description
    "Russian Dolls Bandwidth Constraints Model type.";
  reference
    "RFC 4127: Russian Dolls Bandwidth Constraints Model for
    Diffserv-aware MPLS Traffic Engineering";
}

identity bc-model-mam {
```

```
    base bc-model-type;
    description
        "Maximum Allocation Bandwidth Constraints Model type.";
    reference
        "RFC 4125: Maximum Allocation Bandwidth Constraints Model for
        Diffserv-aware MPLS Traffic Engineering";
}

identity bc-model-mar {
    base bc-model-type;
    description
        "Maximum Allocation with Reservation Bandwidth Constraints
        Model type.";
    reference
        "RFC 4126: Max Allocation with Reservation Bandwidth
        Constraints Model for Diffserv-aware MPLS Traffic Engineering
        & Performance Comparisons";
}

/*
 * Groupings
 */

grouping performance-metrics-attributes-packet {
    description
        "Contains PM attributes.";
    uses te-types:performance-metrics-attributes {
        augment "performance-metrics-one-way" {
            leaf one-way-min-delay {
                type uint32 {
                    range "0..16777215";
                }
                description
                    "One-way minimum delay or latency in microseconds.";
            }
            leaf one-way-min-delay-normality {
                type te-types:performance-metrics-normality;
                default "normal";
                description
                    "One-way minimum delay or latency normality.";
            }
            leaf one-way-max-delay {
                type uint32 {
                    range "0..16777215";
                }
                description
                    "One-way maximum delay or latency in microseconds.";
            }
        }
    }
}
```

```
leaf one-way-max-delay-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "One-way maximum delay or latency normality.";
}
leaf one-way-delay-variation {
  type uint32 {
    range "0..16777215";
  }
  description
    "One-way delay variation in microseconds.";
  reference
    "RFC 5481: Packet Delay Variation Applicability
    Statement, Section 4.2";
}
leaf one-way-delay-variation-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "One-way delay variation normality.";
  reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric
    Extensions
    RFC 7823: Performance-Based Path Selection for
    Explicitly Routed Label Switched Paths (LSPs) Using
    TE Metric Extensions
    RFC 8570: IS-IS Traffic Engineering (TE) Metric
    Extensions";
}
leaf one-way-packet-loss {
  type decimal64 {
    fraction-digits 6;
    range "0..50.331642";
  }
  description
    "One-way packet loss as a percentage of the total traffic
    sent over a configurable interval. The finest precision
    is 0.000003%, where the maximum is 50.331642%.";
  reference
    "RFC 8570: IS-IS Traffic Engineering (TE) Metric
    Extensions, Section 4.4";
}
leaf one-way-packet-loss-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "Packet loss normality.";
```

```
        reference
          "RFC 7471: OSPF Traffic Engineering (TE) Metric
           Extensions
          RFC 7823: Performance-Based Path Selection for
           Explicitly Routed Label Switched Paths (LSPs) Using
           TE Metric Extensions
          RFC 8570: IS-IS Traffic Engineering (TE) Metric
           Extensions";
    }
    description
      "PM one-way packet-specific augmentation for a generic PM
       grouping.";
  }
  augment "performance-metrics-two-way" {
    leaf two-way-min-delay {
      type uint32 {
        range "0..16777215";
      }
      default "0";
      description
        "Two-way minimum delay or latency in microseconds.";
    }
    leaf two-way-min-delay-normality {
      type te-types:performance-metrics-normality;
      default "normal";
      description
        "Two-way minimum delay or latency normality.";
      reference
        "RFC 7471: OSPF Traffic Engineering (TE) Metric
         Extensions
        RFC 7823: Performance-Based Path Selection for
         Explicitly Routed Label Switched Paths (LSPs) Using
         TE Metric Extensions
        RFC 8570: IS-IS Traffic Engineering (TE) Metric
         Extensions";
    }
    leaf two-way-max-delay {
      type uint32 {
        range "0..16777215";
      }
      default "0";
      description
        "Two-way maximum delay or latency in microseconds.";
    }
    leaf two-way-max-delay-normality {
      type te-types:performance-metrics-normality;
      default "normal";
      description
```

```
        "Two-way maximum delay or latency normality.";
    reference
        "RFC 7471: OSPF Traffic Engineering (TE) Metric
        Extensions
        RFC 7823: Performance-Based Path Selection for
        Explicitly Routed Label Switched Paths (LSPs) Using
        TE Metric Extensions
        RFC 8570: IS-IS Traffic Engineering (TE) Metric
        Extensions";
}
leaf two-way-delay-variation {
    type uint32 {
        range "0..16777215";
    }
    default "0";
    description
        "Two-way delay variation in microseconds.";
    reference
        "RFC 5481: Packet Delay Variation Applicability
        Statement, Section 4.2";
}
leaf two-way-delay-variation-normality {
    type te-types:performance-metrics-normality;
    default "normal";
    description
        "Two-way delay variation normality.";
    reference
        "RFC 7471: OSPF Traffic Engineering (TE) Metric
        Extensions
        RFC 7823: Performance-Based Path Selection for
        Explicitly Routed Label Switched Paths (LSPs) Using
        TE Metric Extensions
        RFC 8570: IS-IS Traffic Engineering (TE) Metric
        Extensions";
}
leaf two-way-packet-loss {
    type decimal64 {
        fraction-digits 6;
        range "0..50.331642";
    }
    default "0";
    description
        "Two-way packet loss as a percentage of the total traffic
        sent over a configurable interval. The finest precision
        is 0.000003%.";
}
leaf two-way-packet-loss-normality {
    type te-types:performance-metrics-normality;
```



```
        default "normal";
        description
            "Two-way packet loss normality.";
    }
    description
        "PM two-way packet-specific augmentation for a generic PM
        grouping.";
    reference
        "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
        RFC 7823: Performance-Based Path Selection for
        Explicitly Routed Label Switched Paths (LSPs) Using
        TE Metric Extensions
        RFC 8570: IS-IS Traffic Engineering (TE) Metric
        Extensions";
    }
}

grouping one-way-performance-metrics-packet {
    description
        "One-way packet PM throttle grouping.";
    leaf one-way-min-delay {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "One-way minimum delay or latency in microseconds.";
    }
    leaf one-way-max-delay {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "One-way maximum delay or latency in microseconds.";
    }
    leaf one-way-delay-variation {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "One-way delay variation in microseconds.";
    }
    leaf one-way-packet-loss {
        type decimal64 {
            fraction-digits 6;
        }
    }
}
```

```
        range "0..50.331642";
    }
    default "0";
    description
        "One-way packet loss as a percentage of the total traffic
        sent over a configurable interval. The finest precision is
        0.000003%.";
}
}

grouping two-way-performance-metrics-packet {
    description
        "Two-way packet PM throttle grouping.";
    leaf two-way-min-delay {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "Two-way minimum delay or latency in microseconds.";
    }
    leaf two-way-max-delay {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "Two-way maximum delay or latency in microseconds.";
    }
    leaf two-way-delay-variation {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description
            "Two-way delay variation in microseconds.";
    }
    leaf two-way-packet-loss {
        type decimal64 {
            fraction-digits 6;
            range "0..50.331642";
        }
        default "0";
        description
            "Two-way packet loss as a percentage of the total traffic
            sent over a configurable interval. The finest precision is
            0.000003%.";
    }
}
```

```
}

grouping performance-metrics-throttle-container-packet {
  description
    "Packet PM threshold grouping.";
  uses te-types:performance-metrics-throttle-container {
    augment "throttle/threshold-out" {
      uses one-way-performance-metrics-packet;
      uses two-way-performance-metrics-packet;
      description
        "PM threshold-out packet augmentation for a
        generic grouping.";
    }
    augment "throttle/threshold-in" {
      uses one-way-performance-metrics-packet;
      uses two-way-performance-metrics-packet;
      description
        "PM threshold-in packet augmentation for a
        generic grouping.";
    }
    augment "throttle/threshold-accelerated-advertisement" {
      uses one-way-performance-metrics-packet;
      uses two-way-performance-metrics-packet;
      description
        "PM accelerated advertisement packet augmentation for a
        generic grouping.";
    }
  }
}

grouping te-packet-path-bandwidth {
  description
    "Path bandwidth for Packet. ";
  leaf bandwidth-profile-name {
    type string;
    description "Name of Bandwidth Profile.";
  }
  leaf bandwidth-profile-type {
    type identityref {
      base bandwidth-profile-type;
    }
    description "Type of Bandwidth Profile.";
  }

  leaf cir {
    type uint64;
    description
      "Committed Information Rate in Kbps";
  }
}
```

```
    }

    leaf eir {
      type uint64;
      /*
       * Need to indicate that EIR is not supported by RFC 2697
       */
      must
        ' ../bw-profile-type = "etht-types:mef-10-bwp" or ' +
        ' ../bw-profile-type = "etht-types:rfc-2698-bwp" or ' +
        ' ../bw-profile-type = "etht-types:rfc-4115-bwp"

      must
        ' ../bw-profile-type != "etht-types:rfc-2697-bwp"
    */
    description
      "Excess Information Rate in Kbps
       In case of RFC 2698, PIR = CIR + EIR";
    }

    leaf cbs {
      type uint64;
      description
        "Committed Burst Size in in KBytes";
    }

    leaf ebs {
      type uint64;
      description
        "Excess Burst Size in KBytes.
       In case of RFC 2698, PBS = CBS + EBS";
    }
  }
}

grouping te-packet-link-bandwidth {
  description
    "Link Bandwidth for Packet. ";
  leaf packet-bandwidth {
    type bandwidth-scientific-notation;
    description
      "Available bandwith value expressed in kilobits per
       second";
  }
}
}
}
<CODE ENDS>
```

6.3. Packet Switching TE Topology Module

This module references [RFC7471], [RFC7823], [RFC8294], [RFC8345], [RFC8346], [RFC8570], [RFC8776], and [RFC8795].

```
<CODE BEGINS> file "ietf-te-topology-packet@2021-07-11.yang"
module ietf-te-topology-packet {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-topology-packet";

  prefix "tet-pkt";

  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";
  }

  import ietf-routing-types {
    prefix "rt-types";
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }

  import ietf-te-topology {
    prefix "tet";
    reference
      "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
      Engineering (TE) Topologies";
  }

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

  import ietf-te-packet-types {
    prefix "te-packet-types";
    reference
```

```
"I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
  Types";
}

organization
  "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@cisco.com>

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

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

description
  "YANG data model for representing and manipulating PSC (Packet
  Switching) TE Topologies.

  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).

  This version of this YANG module is part of RFC XXXX; see the
  RFC itself for full legal notices.";

revision 2021-07-11 {
```

```
    description
      "Initial revision";
    reference
      "RFC XXXX: YANG Data Model for Layer 3 TE Topologies";
  }

/*
 * Features
 */

feature te-performance-metric {
  description
    "This feature indicates that the system supports
     TE performance metric.";
  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";
}

/*
 * Groupings
 */
grouping packet-switch-capable-container {
  description
    "The container of packet switch capable attributes.";
  container packet-switch-capable {
    description
      "Interface has packet-switching capabilities.";
    leaf minimum-lsp-bandwidth {
      type rt-types:bandwidth-ieee-float32;
      description
        "Minimum LSP Bandwidth. Units in bytes per second";
    }
    leaf interface-mtu {
      type uint16;
      description
        "Interface MTU.";
    }
  }
}

/*
 * Augmentations
 */
/* Augmentations to network-types */
```

```
augment "/nw:networks/nw:network/nw:network-types/"
+ "tet:te-topology" {
  description
    "Defines the packet TE topology type.";
  container packet {
    presence "Indicates packet TE topology.";
    description
      "Its presence identifies the packet TE topology type.";
  }
}

/* Augmentations to connectivity-matrix */
augment "/nw:networks/nw:network/nw:node/tet:te/"
+ "tet:te-node-attributes/tet:connectivity-matrices" {
  when "../..../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature te-performance-metric;
    refine performance-metrics-one-way {
      config false;
    }
    refine performance-metrics-two-way {
      config false;
    }
  }
}
uses
  te-packet-types:performance-metrics-throttle-container-packet {
    if-feature te-performance-metric;
  }
}

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/"
  + "tet-pkt:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature te-performance-metric;
  }
}
```



```
    refine performance-metrics-one-way {
        config false;
    }
    refine performance-metrics-two-way {
        config false;
    }
}
uses
    te-packet-types:performance-metrics-throttle-container-packet {
        if-feature te-performance-metric;
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices" {
    when "../.../nw:network-types/tet:te-topology/"
        + "tet-pkt:packet" {
        description
            "Augment only for packet TE topology";
    }
    description
        "Parameters for PSC TE topology.";
    uses te-packet-types:performance-metrics-attributes-packet {
        if-feature te-performance-metric;
    }
    uses
        te-packet-types:performance-metrics-throttle-container-packet {
            if-feature te-performance-metric;
        }
    }
}

augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:information-source-entry/tet:connectivity-matrices/"
    + "tet:connectivity-matrix" {
    when "../.../nw:network-types/tet:te-topology/"
        + "tet-pkt:packet" {
        description
            "Augment only for packet TE topology";
    }
    description
        "Parameters for PSC TE topology.";
    uses te-packet-types:performance-metrics-attributes-packet {
        if-feature te-performance-metric;
    }
    uses
        te-packet-types:performance-metrics-throttle-container-packet {
            if-feature te-performance-metric;
        }
    }
}
```

```
}

/* Augmentations to tunnel-termination-point */
augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities" {
  when "../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature te-performance-metric;
    refine performance-metrics-one-way {
      config false;
    }
    refine performance-metrics-two-way {
      config false;
    }
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature te-performance-metric;
    }
  }

augment "/nw:networks/nw:network/nw:node/tet:te/"
  + "tet:tunnel-termination-point/"
  + "tet:local-link-connectivities/"
  + "tet:local-link-connectivity" {
  when "../../../nw:network-types/tet:te-topology/"
  + "tet-pkt:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature te-performance-metric;
    refine performance-metrics-one-way {
      config false;
    }
    refine performance-metrics-two-way {
      config false;
    }
  }
  }
}
```

```
    uses
      te-packet-types:performance-metrics-throttle-container-packet {
        if-feature te-performance-metric;
      }
  }

  /* Augmentations to te-link-attributes */
  augment "/nw:networks/tet:te/tet:templates/"
    + "tet:link-template/tet:te-link-attributes" {
    when "tet:interface-switching-capability "
      + "[tet:switching-capability = 'te-types:switching-pscl']" {
      description
        "Valid only for PSC";
    }
    description
      "Parameters for PSC TE topology.";
    uses te-packet-types:performance-metrics-attributes-packet {
      if-feature te-performance-metric;
      refine performance-metrics-one-way {
        config false;
      }
      refine performance-metrics-two-way {
        config false;
      }
    }
    uses
      te-packet-types:performance-metrics-throttle-container-packet {
        if-feature te-performance-metric;
      }
  }

  augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:te-link-attributes" {
    when "(../../nw:network-types/tet:te-topology/tet-pkt:packet) "
      + "and (tet:interface-switching-capability "
      + "[tet:switching-capability = 'te-types:switching-pscl'])" {
      description
        "Valid only for PSC";
    }
    description
      "Parameters for PSC TE topology.";
    uses te-packet-types:performance-metrics-attributes-packet {
      if-feature te-performance-metric;
      refine performance-metrics-one-way {
        config false;
      }
      refine performance-metrics-two-way {
        config false;
      }
    }
  }
```

```

    }
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature te-performance-metric;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:information-source-entry" {
    when "(../../../../../nw:network-types/tet:te-topology/tet-pkt:packet) "
      + " and (tet:interface-switching-capability "
      + "[tet:switching-capability = 'te-types:switching-pscl'])" {
      description
        "Valid only for PSC";
    }
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature te-performance-metric;
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature te-performance-metric;
    }
}

/* Augmentations to interface-switching-capability */
augment "/nw:networks/tet:te/tet:templates/"
  + "tet:link-template/tet:te-link-attributes/"
  + "tet:interface-switching-capability" {
    when "tet:switching-capability = 'te-types:switching-pscl' " {
      description
        "Valid only for PSC";
    }
  }
  description
    "Parameters for PSC TE topology.";
  uses packet-switch-capable-container;
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
  + "tet:te-link-attributes/"
  + "tet:interface-switching-capability" {
    when "(../../../../../nw:network-types/tet:te-topology/"
      + "tet-pkt:packet) and "
      + "(tet:switching-capability = 'te-types:switching-pscl')" {
      description
        "Valid only for PSC";
    }
  }

```

```

    }
    description
      "Parameters for PSC TE topology.";
    uses packet-switch-capable-container;
  }

  augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:information-source-entry/"
    + "tet:interface-switching-capability" {
    when "(../../../../../nw:network-types/tet:te-topology/"
      + "tet-pkt:packet) and "
      + "(tet:switching-capability = 'te-types:switching-pscl')" {
      description
        "Valid only for PSC";
    }
    description
      "Parameters for PSC TE topology.";
    uses packet-switch-capable-container;
  }

  /*
  * 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/"
      + "tet-pkt:packet" {
      description
        "Augmentation parameters apply only for networks with
        Packet TE topology type.";
    }
    description
      "Augment maximum LSP TE bandwidth for the link termination
      point (LTP).";
    case packet {
      uses te-packet-types:te-packet-path-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/"
      + "tet-pkt:packet" {
      description

```

```
        "Augmentation parameters apply only for networks with
        Packet TE topology type.";
    }
    description
        "Augment TE bandwidth path constraints of the TE node
        connectivity matrices.";
    case packet {
        uses te-packet-types:te-packet-link-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/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment TE bandwidth path constraints of the
        connectivity matrix entry.";
    case packet {
        uses te-packet-types:te-packet-link-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/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment TE bandwidth path constraints of the TE node
        connectivity matrices information source.";
    case packet {
        uses te-packet-types:te-packet-link-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/"
+ "tet-pkt:packet" {
description
    "Augmentation parameters apply only for networks with
      Packet TE topology type.";
}
description
    "Augment TE bandwidth path constraints of the
      connectivity matrix entry information source";
case packet {
    uses te-packet-types:te-packet-link-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/"
+ "tet-pkt:packet" {
description
    "Augmentation parameters apply only for networks with
      Packet TE topology type.";
}
description
    "Augment client TE bandwidth of the tunnel termination point
      (TTP)";
case packet {
    uses te-packet-types:te-packet-link-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/"
+ "tet-pkt:packet" {
description
    "Augmentation parameters apply only for networks with
      Packet TE topology type.";
}
description
    "Augment TE bandwidth path constraints for the TTP
      Local Link Connectivities.";
case packet {
```

```

    uses te-packet-types:te-packet-link-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/"
    + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment TE bandwidth path constraints for the TTP
    Local Link Connectivity entry.";
  case packet {
    uses te-packet-types:te-packet-link-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/"
    + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with
      Packet TE topology type.";
  }
  description
    "Augment maximum LSP TE bandwidth for the TE link.";
  case packet {
    uses te-packet-types:te-packet-path-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/"
    + "tet-pkt:packet" {
    description
      "Augmentation parameters apply only for networks with

```



```
        Packet TE topology type.";
    }
    description
        "Augment maximum TE bandwidth for the TE link";
    case packet {
        uses te-packet-types:te-packet-link-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/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment maximum reservable TE bandwidth for the TE link";
    case packet {
        uses te-packet-types:te-packet-link-bandwidth;
    }
}

augment "/nw:networks/nw:network/nt:link/tet:te/"
    + "tet:te-link-attributes/"
    + "tet:unreserved-bandwidth/"
    + "tet:te-bandwidth/tet:technology" {
    when "../../../nw:network-types/tet:te-topology/"
        + "tet-pkt:packet" {
        description
            "Augmentation parameters apply only for networks with
            Packet TE topology type.";
    }
    description
        "Augment unreserved TE bandwidth for the TE Link";
    case packet {
        uses te-packet-types:te-packet-link-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/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum LSP TE bandwidth for the TE link
  information source";
case packet {
  uses te-packet-types:te-packet-path-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/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum TE bandwidth for the TE link
  information source";
case packet {
  uses te-packet-types:te-packet-link-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/"
  + "tet-pkt:packet" {
  description
    "Augmentation parameters apply only for networks with
    Packet TE topology type.";
}
description
  "Augment maximum reservable TE bandwidth for the TE link
  information source";
case packet {
  uses te-packet-types:te-packet-link-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/"
      + "tet-pkt:packet" {
      description
        "Augmentation parameters apply only for networks with
        Packet TE topology type.";
    }
    description
      "Augment unreserved TE bandwidth of the TE link
      information source";
    case packet {
      uses te-packet-types:te-packet-link-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" {
    description
      "Augment maximum LSP TE bandwidth of the TE link
      template";
    case packet {
      uses te-packet-types:te-packet-path-bandwidth;
    }
  }

  augment "/nw:networks/tet:te/tet:templates/"
    + "tet:link-template/tet:te-link-attributes/"
    + "tet:max-link-bandwidth/"
    + "tet:te-bandwidth/tet:technology" {
    description
      "Augment maximum TE bandwidth the TE link template";
    case packet {
      uses te-packet-types:te-packet-link-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" {

```

```
    description
      "Augment maximum reservable TE bandwidth for the TE link
       template.";
    case packet {
      uses te-packet-types:te-packet-link-bandwidth;
    }
  }

  augment "/nw:networks/tet:te/tet:templates/"
    + "tet:link-template/tet:te-link-attributes/"
    + "tet:unreserved-bandwidth/"
    + "tet:te-bandwidth/tet:technology" {
    description
      "Augment unreserved TE bandwidth the TE link template";
    case packet {
      uses te-packet-types:te-packet-link-bandwidth;
    }
  }
}
<CODE ENDS>
```

7. 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-l3-te-topology
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

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

URI: urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-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 [RFC6020]:

name: ietf-l3-te-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology
prefix: l3tet
reference: RFC XXXX

name: ietf-l3-te-topology-state
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state
prefix: l3tet-s
reference: RFC XXXX

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

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

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/l3t:l3-unicast-topology/
l3-te

This subtree specifies the layer 3 TE topology type. Modifying the configurations can make layer 3 TE topology type invalid and cause interruption to all layer 3 TE networks.

/nw:networks/nw:network/l3t:l3-topology-attributes/l3-te-topology-
attributes

This subtree specifies the topology-wide configurations, including the reference to a TE topology from a layer 3 TE topology. Modifying the configurations here can cause traffic disabled or rerouted in this topology and the connected topologies.

/nw:networks/nw:network/nw:node/l3t:l3-node-attributes/l3-te-node-
attributes

This subtree specifies the configurations of layer 3 TE nodes. Modifying the configurations in this subtree can change the relationship between a TE node and a node in a layer 3 TE topology, causing traffic disabled or rerouted in the specified nodes and the related layer 3 topologies.

/nw:networks/nw:network/nw:node/nt:termination-point//l3t:l3-
termination-point-attributes/l3-te-tp-attributes

This subtree specifies the configurations of layer 3 TE link termination points. Modifying the configurations in this subtree can change the relationship between a TE link termination point and a link termination point in a layer 3 TE topology, causing traffic disabled or rerouted on the related layer 3 links and the related layer 3 topologies.

/nw:networks/nw:network/nt:link/l3t:l3-link-attributes/l3-te-link-
attributes

This subtree specifies the configurations of layer 3 TE links. Modifying the configurations in this subtree can change the relationship between a TE link and a link in a layer 3 TE

topology, causing traffic disabled or rerouted on the specified layer 3 link and the related layer 3 topologies.

performance-metric containers

The container "performance-metric" is augmented to multiple locations of the base TE topology model, as specified in Section 3.2. Modifying the configuration in such a container can change the behaviors of performance metric monitoring, causing traffic disabled or rerouted on the related layer 3 links, nodes, or 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/l3t:l3-unicast-topology/
l3-te

Unauthorized access to this subtree can disclose the layer 3 TE topology type.

/nw:networks/nw:network/l3t:l3-topology-attributes/l3-te-topology-
attributes

Unauthorized access to this subtree can disclose the topology-wide configurations, including the reference to a TE topology from a layer 3 topology.

/nw:networks/nw:network/nw:node/l3t:l3-node-attributes/l3-te-node-
attributes

Unauthorized access to this subtree can disclose the operational state information of layer 3 TE nodes.

/nw:networks/nw:network/nw:node/nt:termination-point//l3t:l3-
termination-point-attributes/l3-te-tp-attributes

Unauthorized access to this subtree can disclose the operational state information of layer 3 TE link termination points.

/nw:networks/nw:network/nt:link/l3t:l3-link-attributes/l3-te-link-
attributes

Unauthorized access to this subtree can disclose the operational state information of layer 3 TE links.

performance-metric containers

The container "performance-metric" is augmented to multiple locations of the base TE topology model, as specified in Section 3.2. Unauthorized access to this subtree can disclose the operational state information of performance metric monitoring.

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>>.
- [RFC2697] Heinanen, J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, DOI 10.17487/RFC2697, September 1999, <<https://www.rfc-editor.org/info/rfc2697>>.
- [RFC2698] Heinanen, J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, DOI 10.17487/RFC2698, September 1999, <<https://www.rfc-editor.org/info/rfc2698>>.
- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC4115] Aboul-Magd, O. and S. Rabie, "A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic", RFC 4115, DOI 10.17487/RFC4115, July 2005, <<https://www.rfc-editor.org/info/rfc4115>>.
- [RFC4124] Le Faucheur, F., Ed., "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering", RFC 4124, DOI 10.17487/RFC4124, June 2005, <<https://www.rfc-editor.org/info/rfc4124>>.
- [RFC4125] Le Faucheur, F. and W. Lai, "Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4125, DOI 10.17487/RFC4125, June 2005, <<https://www.rfc-editor.org/info/rfc4125>>.
- [RFC4126] Ash, J., "Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering & Performance Comparisons", RFC 4126, DOI 10.17487/RFC4126, June 2005, <<https://www.rfc-editor.org/info/rfc4126>>.

- [RFC4127] Le Faucheur, F., Ed., "Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4127, DOI 10.17487/RFC4127, June 2005, <<https://www.rfc-editor.org/info/rfc4127>>.
- [RFC5481] Morton, A. and B. Claise, "Packet Delay Variation Applicability Statement", RFC 5481, DOI 10.17487/RFC5481, March 2009, <<https://www.rfc-editor.org/info/rfc5481>>.
- [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>>.
- [RFC7074] Berger, L. and J. Meuric, "Revised Definition of the GMPLS Switching Capability and Type Fields", RFC 7074, DOI 10.17487/RFC7074, November 2013, <<https://www.rfc-editor.org/info/rfc7074>>.
- [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>>.

- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [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>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [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>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [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>>.

9.2. Informative References

- [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, DOI 10.17487/RFC7823, May 2016, <<https://www.rfc-editor.org/info/rfc7823>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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-teas-yang-te-mpls]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for MPLS Traffic Engineering Tunnels", draft-ietf-teas-yang-te-mpls-03 (work in progress), March 2020.
- [IEEE754] IEEE, "IEEE Standard for Floating-Point Arithmetic", IEEE Std 754-2008 DOI 10.1109/IEEESTD.2008.4610935, August 2008.
- [ISO-IEC-C99]
ISO/IEC, "Programming languages - C", ISO/IEC 9899:1999 /Cor.1:2001(E), September 2001.

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

The YANG modules `ietf-l3-te-topology` and `ietf-te-topology-packet` defined in this document are 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 modules, `ietf-l3-te-topology-state` and `ietf-te-topology-packet-state`, are defined as state models, which mirror the modules `ietf-l3-te-topology` and `ietf-te-topology-packet` 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 modules, `ietf-l3-te-topology-state` and `ietf-te-topology-packet-state`, are redundant and SHOULD NOT be supported by implementations that support NMDA.

As the structure of the companion modules mirrors that of the cooresponding NMDA models, the YANG trees of the companion modules are not depicted separately.

A.1. Layer 3 TE Topology State Module

This module references [RFC8345], and [RFC8346].

```
<CODE BEGINS> file "ietf-l3-te-topology-state@2020-05-03.yang"
module ietf-l3-te-topology-state {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state";
  prefix "l3tet-s";

  import ietf-l3-te-topology {
    prefix "l3tet";
  }
  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";
  }
  import ietf-l3-unicast-topology-state {
    prefix "l3t-s";
    reference "RFC 8346: A YANG Data Model for Layer 3 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@cisco.com>>

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

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

description

"YANG data model for representing operational state information
of Layer 3 TE Topologies, when NMDA is not supported.

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>).

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";

revision 2020-05-03 {

description

"Initial revision";

reference "RFC XXXX: YANG Data Model for Layer 3 TE Topologies";

}

```
augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
+ "l3t-s:l3-unicast-topology" {
  description
    "Defines the L3 TE topology type.";
  uses l3tet:l3-te-topology-type;
}

augment "/nw-s:networks/nw-s:network/"
+ "l3t-s:l3-topology-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
  + "l3tet-s:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment topology configuration";
  uses l3tet:l3-te-topology-attributes;
}

augment "/nw-s:networks/nw-s:network/nw-s:node/"
+ "l3t-s:l3-node-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
  + "l3tet-s:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment node configuration";
  uses l3tet:l3-te-node-attributes;
}

augment "/nw-s:networks/nw-s:network/nw-s:node/"
+ "nt-s:termination-point/"
+ "l3t-s:l3-termination-point-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
  + "l3tet-s:l3-te" {
    description
      "Augment only for L3 TE topology";
  }
  description
    "Augment termination point configuration";
  uses l3tet:l3-te-tp-attributes;
}

augment "/nw-s:networks/nw-s:network/nt-s:link/"
+ "l3t-s:l3-link-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
  + "l3tet-s:l3-te" {
```

```
        description
          "Augment only for L3 TE topology";
      }
      description
        "Augment link configuration";
      uses l3tet:l3-te-link-attributes;
    }
  }
<CODE ENDS>
```

A.2. Packet Switching TE Topology State Module

```
<CODE BEGINS> file "ietf-te-topology-packet-state@2020-07-03.yang"
module iETF-te-topology-packet-state {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-te-topology-packet-state";

  prefix "tet-pkt-s";

  import iETF-te-topology-packet {
    prefix "tet-pkt";
  }

  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";
  }

  import iETF-te-topology-state {
    prefix "tet-s";
    reference
      "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
      Engineering (TE) Topologies";
  }

  import iETF-te-types {
    prefix "te-types";
```

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

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

organization
  "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@cisco.com>

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

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

description
  "YANG data model for representing operational state information
  of PSC (Packet Switching) TE Topologies, when NMDA is not
  supported.

  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>).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";

```
revision 2020-07-03 {
  description
    "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Layer 3 TE Topologies";
}

/*
 * Augmentations
 */
/* Augmentations to network-types */
augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
+ "tet-s:te-topology" {
  description
    "Defines the packet TE topology type.";
  container packet {
    presence "Indicates packet TE topology.";
    description
      "Its presence identifies the packet TE topology type.";
  }
}

/* Augmentations to connectivity-matrix */
augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
+ "tet-s:te-node-attributes/tet-s:connectivity-matrices" {
  when "../..../nw-s:network-types/tet-s:te-topology/"
+ "tet-pkt-s:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC (Packet Switching) TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature tet-pkt:te-performance-metric;
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature tet-pkt:te-performance-metric;
    }
}
```

```
augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
  + "tet-s:te-node-attributes/tet-s:connectivity-matrices/"
  + "tet-s:connectivity-matrix" {
when "../.../nw-s:network-types/tet-s:te-topology/"
  + "tet-pkt-s:packet" {
  description
    "Augment only for packet TE topology";
}
description
  "Parameters for PSC TE topology.";
uses te-packet-types:performance-metrics-attributes-packet {
  if-feature tet-pkt:te-performance-metric;
}
uses
  te-packet-types:performance-metrics-throttle-container-packet {
    if-feature tet-pkt:te-performance-metric;
  }
}

augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
  + "tet-s:information-source-entry/"
  + "tet-s:connectivity-matrices" {
when "../.../nw-s:network-types/tet-s:te-topology/"
  + "tet-pkt-s:packet" {
  description
    "Augment only for packet TE topology";
}
description
  "Parameters for PSC TE topology.";
uses te-packet-types:performance-metrics-attributes-packet {
  if-feature tet-pkt:te-performance-metric;
}
uses
  te-packet-types:performance-metrics-throttle-container-packet {
    if-feature tet-pkt:te-performance-metric;
  }
}

augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
  + "tet-s:information-source-entry/"
  + "tet-s:connectivity-matrices/"
  + "tet-s:connectivity-matrix" {
when "../.../nw-s:network-types/tet-s:te-topology/"
  + "tet-pkt-s:packet" {
  description
    "Augment only for packet TE topology";
}
description
```

```
    "Parameters for PSC TE topology.";
    uses te-packet-types:performance-metrics-attributes-packet {
      if-feature tet-pkt:te-performance-metric;
    }
    uses
      te-packet-types:performance-metrics-throttle-container-packet {
        if-feature tet-pkt:te-performance-metric;
      }
  }

/* Augmentations to tunnel-termination-point */
augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
  + "tet-s:tunnel-termination-point/"
  + "tet-s:local-link-connectivities" {
  when "../.../nw-s:network-types/tet-s:te-topology/"
    + "tet-pkt-s:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature tet-pkt:te-performance-metric;
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature tet-pkt:te-performance-metric;
    }
}

augment "/nw-s:networks/nw-s:network/nw-s:node/tet-s:te/"
  + "tet-s:tunnel-termination-point/"
  + "tet-s:local-link-connectivities/"
  + "tet-s:local-link-connectivity" {
  when "../.../nw-s:network-types/tet-s:te-topology/"
    + "tet-pkt-s:packet" {
    description
      "Augment only for packet TE topology";
  }
  description
    "Parameters for PSC TE topology.";
  uses te-packet-types:performance-metrics-attributes-packet {
    if-feature tet-pkt:te-performance-metric;
  }
  uses
    te-packet-types:performance-metrics-throttle-container-packet {
      if-feature tet-pkt:te-performance-metric;
    }
}
```

```

    }

    /* Augmentations to te-link-attributes */
    augment "/nw-s:networks/tet-s:te/tet-s:templates/"
      + "tet-s:link-template/tet-s:te-link-attributes" {
      when "tet-s:interface-switching-capability "
        + "[tet-s:switching-capability = 'te-types:switching-pscl']" {
        description
          "Valid only for PSC";
      }
      description
        "Parameters for PSC TE topology.";
      uses te-packet-types:performance-metrics-attributes-packet {
        if-feature tet-pkt:te-performance-metric;
      }
      uses
        te-packet-types:performance-metrics-throttle-container-packet {
          if-feature tet-pkt:te-performance-metric;
        }
    }

    augment "/nw-s:networks/nw-s:network/nt-s:link/tet-s:te/"
      + "tet-s:te-link-attributes" {
      when "(../../nw-s:network-types/tet-s:te-topology/"
        + "tet-pkt-s:packet) and "
        + "(tet-s:interface-switching-capability "
        + "[tet-s:switching-capability = 'te-types:switching-pscl'])" {
        description "Valid only for PSC";
      }
      description
        "Parameters for PSC TE topology.";
      uses te-packet-types:performance-metrics-attributes-packet {
        if-feature tet-pkt:te-performance-metric;
      }
      uses
        te-packet-types:performance-metrics-throttle-container-packet {
          if-feature tet-pkt:te-performance-metric;
        }
    }

    augment "/nw-s:networks/nw-s:network/nt-s:link/tet-s:te/"
      + "tet-s:information-source-entry" {
      when "(../../nw-s:network-types/tet-s:te-topology/"
        + "tet-pkt-s:packet) and "
        + "(tet-s:interface-switching-capability "
        + "[tet-s:switching-capability = 'te-types:switching-pscl'])" {
        description "Valid only for PSC";
      }
    }

```

```
description
  "Parameters for PSC TE topology.";
uses te-packet-types:performance-metrics-attributes-packet {
  if-feature tet-pkt:te-performance-metric;
}
uses
  te-packet-types:performance-metrics-throttle-container-packet {
    if-feature tet-pkt:te-performance-metric;
  }
}

/* Augmentations to interface-switching-capability */
augment "/nw-s:networks/tet-s:te/tet-s:templates/"
  + "tet-s:link-template/tet-s:te-link-attributes/"
  + "tet-s:interface-switching-capability" {
  when "tet-s:switching-capability = 'te-types:switching-pscl'" {
    description "Valid only for PSC";
  }
  description
    "Parameters for PSC TE topology.";
  uses tet-pkt:packet-switch-capable-container;
}

augment "/nw-s:networks/nw-s:network/nt-s:link/tet-s:te/"
  + "tet-s:te-link-attributes/"
  + "tet-s:interface-switching-capability" {
  when "(../../../../../nw-s:network-types/tet-s:te-topology/"
    + "tet-pkt-s:packet) and "
    + "(tet-s:switching-capability = 'te-types:switching-pscl')" {
    description "Valid only for PSC";
  }
  description
    "Parameters for PSC TE topology.";
  uses tet-pkt:packet-switch-capable-container;
}

augment "/nw-s:networks/nw-s:network/nt-s:link/tet-s:te/"
  + "tet-s:information-source-entry/"
  + "tet-s:interface-switching-capability" {
  when "(../../../../../nw-s:network-types/tet-s:te-topology/"
    + "tet-pkt-s:packet) and "
    + "(tet-s:switching-capability = 'te-types:switching-pscl')" {
    description
      "Valid only for PSC";
  }
  description
    "Parameters for PSC TE topology.";
  uses tet-pkt:packet-switch-capable-container;
```

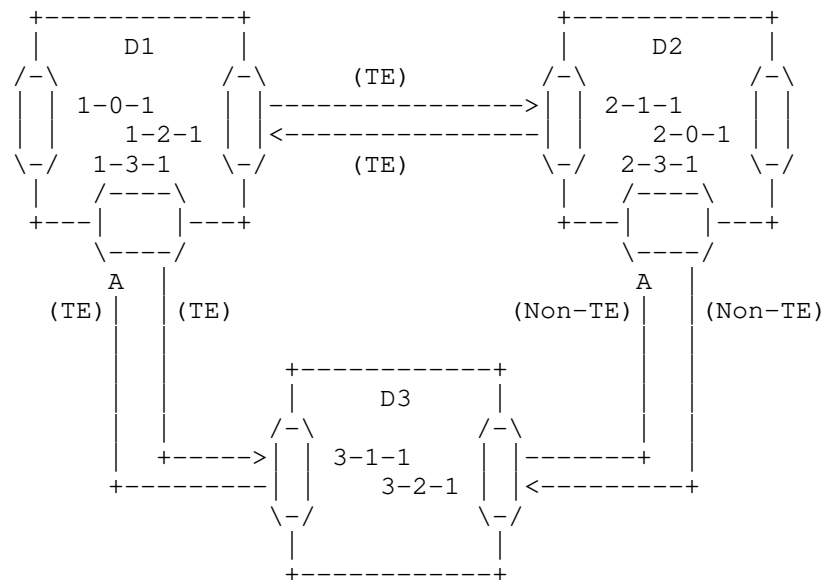
```

    }
  }
<CODE ENDS>

```

Appendix B. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-l3-te-topology" for the topology that is depicted in the following diagram.



The corresponding instance data tree is depicted below. Note that some lines have been wrapped to adhere to the 72-character line limitation of RFCs.

```

{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-topo-te",
        "network-types": {
          "ietf-te-topology:te-topology": {
          }
        }
      },
    ],
  },
}

```

```
"ietf-te-topology:te-topology-identifier": {
  "provider-id":200,
  "client-id":300,
  "topology-id":"example-topo-te"
},
"ietf-te-topology:te": {
},
"node": [
  {
    "node-id":"D1",
    "ietf-te-topology:te-node-id":"2.0.1.1",
    "ietf-te-topology:te": {
      "te-node-attributes": {
      }
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id":"1-2-1",
        "ietf-te-topology:te-tp-id":10201,
        "ietf-te-topology:te": {
          "interface-switching-capability": [
            {
              "switching-capability":
                "ietf-te-types:switching-pscl",
              "encoding":
                "ietf-te-types:lsp-encoding-ethernet"
            }
          ]
        }
      }
    ],
    "ietf-te-topology:te-tp-id":10301,
    "ietf-te-topology:te": {
      "interface-switching-capability": [
        {
          "switching-capability":
            "ietf-te-types:switching-pscl",
          "encoding":
            "ietf-te-types:lsp-encoding-ethernet"
        }
      ]
    }
  }
],
{
  "node-id":"D2",
```

```
"ietf-te-topology:te-node-id":"2.0.2.1",
"ietf-te-topology:te": {
  "te-node-attributes": {
  }
},
"ietf-network-topology:termination-point": [
{
  "tp-id":"2-1-1",
  "ietf-te-topology:te-tp-id":20101,
  "ietf-te-topology:te": {
    "interface-switching-capability": [
    {
      "switching-capability":
        "ietf-te-types:switching-pscl",
      "encoding":
        "ietf-te-types:lsp-encoding-ethernet"
    }
  ]
}
]
},
{
  "node-id":"D3",
  "ietf-te-topology:te-node-id":"2.0.3.1",
  "ietf-te-topology:te": {
    "te-node-attributes": {
    }
  },
  "ietf-network-topology:termination-point": [
  {
    "tp-id":"3-1-1",
    "ietf-te-topology:te-tp-id":30101,
    "ietf-te-topology:te": {
      "interface-switching-capability": [
      {
        "switching-capability":
          "ietf-te-types:switching-pscl",
        "encoding":
          "ietf-te-types:lsp-encoding-ethernet"
      }
    ]
  }
  ]
}
]
},
"ietf-network-topology:link": [
```



```

{
  "link-id": "D1,1-2-1,D2,2-1-1",
  "source": {
    "source-node": "D1",
    "source-tp": "1-2-1"
  },
  "destination": {
    "dest-node": "D2",
    "dest-tp": "2-1-1"
  },
  "ietf-te-topology:te": {
    "te-link-attributes": {
      "interface-switching-capability": [
        {
          "switching-capability":
            "ietf-te-types:switching-pscl",
          "encoding": "ietf-te-types:lsp-encoding-ethernet"
        }
      ],
      "max-link-bandwidth": {
        "te-bandwidth": {
          "generic": "0x1p+18"
        }
      },
      "te-default-metric": 100
    }
  },
{
  "link-id": "D2,2-1-1,D1,1-2-1",
  "source": {
    "source-node": "D2",
    "source-tp": "2-1-1"
  },
  "destination": {
    "dest-node": "D1",
    "dest-tp": "1-2-1"
  },
  "ietf-te-topology:te": {
    "te-link-attributes": {
      "interface-switching-capability": [
        {
          "switching-capability":
            "ietf-te-types:switching-pscl",
          "encoding": "ietf-te-types:lsp-encoding-ethernet"
        }
      ],
      "max-link-bandwidth": {

```

```

        "te-bandwidth": {
            "generic": "0x1p+18"
        }
    },
    "te-default-metric": 100
}
},
{
    "link-id": "D1,1-3-1,D3,3-1-1",
    "source": {
        "source-node": "D1",
        "source-tp": "1-3-1"
    },
    "destination": {
        "dest-node": "D3",
        "dest-tp": "3-1-1"
    },
    "ietf-te-topology:te": {
        "te-link-attributes": {
            "interface-switching-capability": [
                {
                    "switching-capability":
                        "ietf-te-types:switching-pscl",
                    "encoding": "ietf-te-types:lsp-encoding-ethernet"
                }
            ],
            "max-link-bandwidth": {
                "te-bandwidth": {
                    "generic": "0x1p+18"
                }
            },
            "te-default-metric": 100
        }
    }
},
{
    "link-id": "D3,3-1-1,D1,1-3-1",
    "source": {
        "source-node": "D3",
        "source-tp": "3-1-1"
    },
    "destination": {
        "dest-node": "D1",
        "dest-tp": "1-3-1"
    },
    "ietf-te-topology:te": {
        "te-link-attributes": {

```

```

        "interface-switching-capability": [
            {
                "switching-capability":
                    "ietf-te-types:switching-pscl",
                "encoding": "ietf-te-types:lsp-encoding-ethernet"
            }
        ],
        "max-link-bandwidth": {
            "te-bandwidth": {
                "generic": "0x1p+18"
            }
        },
        "te-default-metric": 100
    }
}
]
},
{
    "network-id": "example-topo-l3-te",
    "network-types": {
        "ietf-l3-unicast-topology:l3-unicast-topology": {
            "ietf-l3-te-topology:l3-te": {
            }
        }
    },
    "ietf-l3-unicast-topology:l3-topology-attributes": {
        "ietf-l3-te-topology:l3-te-topology-attributes": {
            "network-ref": "example-topo-te"
        }
    },
    "node": [
        {
            "node-id": "D1",
            "ietf-l3-unicast-topology:l3-node-attributes": {
                "router-id": [
                    "203.0.113.1"
                ],
                "prefix": [
                    {
                        "prefix": "203.0.113.1/32"
                    }
                ],
                "ietf-l3-te-topology:l3-te-node-attributes": {
                    "node-ref": "D1",
                    "network-ref": "example-topo-te"
                }
            }
        }
    ],
}

```

```
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "1-0-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
          "unnumbered-id": 101
        }
      },
      {
        "tp-id": "1-2-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
          "unnumbered-id": 121,
          "ietf-l3-te-topology:l3-te-tp-attributes": {
            "network-ref": "example-topo-te",
            "tp-ref": "1-2-1"
          }
        }
      }
    ],
    {
      "tp-id": "1-3-1",
      "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 131,
        "ietf-l3-te-topology:l3-te-tp-attributes": {
          "network-ref": "example-topo-te",
          "tp-ref": "1-3-1"
        }
      }
    }
  ]
},
{
  "node-id": "D2",
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "router-id": [
      "203.0.113.2"
    ],
    "prefix": [
      {
        "prefix": "203.0.113.2/32"
      }
    ],
    "ietf-l3-te-topology:l3-te-node-attributes": {
      "node-ref": "D2",
      "network-ref": "example-topo-te"
    }
  },
  "ietf-network-topology:termination-point": [
    {
      "tp-id": "2-0-1",
```

```

    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id":201
    }
},
{
    "tp-id":"2-1-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id":211,
        "ietf-l3-te-topology:l3-te-tp-attributes": {
            "tp-ref":"2-1-1",
            "network-ref":"example-topo-te"
        }
    }
},
{
    "tp-id":"2-3-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id":231
    }
}
]
},
{
    "node-id":"D3",
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "router-id": [
            "203.0.113.3"
        ],
        "prefix": [
            {
                "prefix":"203.0.113.3/32"
            }
        ],
        "ietf-l3-te-topology:l3-te-node-attributes": {
            "node-ref":"D3",
            "network-ref":"example-topo-te"
        }
    },
    "ietf-network-topology:termination-point": [
        {
            "tp-id":"3-0-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id":301
            }
        },
        {
            "tp-id":"3-1-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {

```

```

        "unnumbered-id":311,
        "ietf-l3-te-topology:l3-te-tp-attributes": {
            "tp-ref":"3-1-1",
            "network-ref":"example-topo-te"
        }
    },
    {
        "tp-id":"3-2-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
            "unnumbered-id":321
        }
    }
]
},
"ietf-network-topology:link": [
    {
        "link-id":"D1,1-2-1,D2,2-1-1",
        "source": {
            "source-node":"D1",
            "source-tp":"1-2-1"
        },
        "destination": {
            "dest-node":"D2",
            "dest-tp":"2-1-1"
        },
        "ietf-l3-unicast-topology:l3-link-attributes": {
            "metric1":"100",
            "ietf-l3-te-topology:l3-te-link-attributes": {
                "link-ref":"D1,1-2-1,D2,2-1-1",
                "network-ref":"example-topo-te"
            }
        }
    },
    {
        "link-id":"D2,2-1-1,D1,1-2-1",
        "source": {
            "source-node":"D2",
            "source-tp":"2-1-1"
        },
        "destination": {
            "dest-node":"D1",
            "dest-tp":"1-2-1"
        },
        "ietf-l3-unicast-topology:l3-link-attributes": {
            "metric1":"100",
            "ietf-l3-te-topology:l3-te-link-attributes": {

```

```

        "link-ref": "D2,2-1-1,D1,1-2-1",
        "network-ref": "example-topo-te"
    }
}
},
{
    "link-id": "D1,1-3-1,D3,3-1-1",
    "source": {
        "source-node": "D1",
        "source-tp": "1-3-1"
    },
    "destination": {
        "dest-node": "D3",
        "dest-tp": "3-1-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-l3-te-topology:l3-te-link-attributes": {
            "link-ref": "D1,1-3-1,D3,3-1-1",
            "network-ref": "example-topo-te"
        }
    }
}
},
{
    "link-id": "D3,3-1-1,D1,1-3-1",
    "source": {
        "source-node": "D3",
        "source-tp": "3-1-1"
    },
    "destination": {
        "dest-node": "D1",
        "dest-tp": "1-3-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-l3-te-topology:l3-te-link-attributes": {
            "link-ref": "D3,3-1-1,D1,1-3-1",
            "network-ref": "example-topo-te"
        }
    }
}
},
{
    "link-id": "D2,2-3-1,D3,3-2-1",
    "source": {
        "source-node": "D2",
        "source-tp": "2-3-1"
    },
    "destination": {

```

```
        "dest-node": "D3",
        "dest-tp": "3-2-1"
      },
      "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100"
      }
    },
    {
      "link-id": "D3,3-2-1,D2,2-3-1",
      "source": {
        "source-node": "D3",
        "source-tp": "3-2-1"
      },
      "destination": {
        "dest-node": "D2",
        "dest-tp": "2-3-1"
      },
      "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100"
      }
    }
  ]
}
}
```

Authors' Addresses

Xufeng Liu
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Igor Bryskin
Individual

EMail: i_bryskin@yahoo.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: Standards Track
Expires: 22 September 2022

I. Busi, Ed.
Huawei Technologies
S. Belotti, Ed.
Nokia
O. Gonzalez de Dios
Telefonica
A. Sharma
Google
D. Ceccarelli
Ericsson
21 March 2022

A YANG Data Model for requesting path computation
draft-ietf-teas-yang-path-computation-18

Abstract

There are scenarios, typically in a hierarchical Software-Defined Networking (SDN) context, where the topology information provided by a Traffic Engineering (TE) network provider may be insufficient for its client to perform multi-domain path computation. In these cases the client would need to request the TE network provider to compute some intra-domain paths.

This document defines a YANG data model which contains Remote Procedure Calls (RPCs) to request path computation. This model complements the solution, defined in RFC YYYY, to configure a TE tunnel path in "compute-only" mode.

[RFC EDITOR NOTE: Please replace RFC YYYY with the RFC number of draft-ietf-teas-yang-te once it has been published.]

Moreover, this document describes some use cases where the path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

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 22 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Terminology	4
1.2. Tree Diagram	5
1.3. Prefixes in Data Node Names	5
2. Use Cases	6
2.1. Packet/Optical Integration	6
2.2. Multi-domain TE networks	10
2.3. Data Center Interconnections	12
2.4. Backward Recursive Path Computation scenario	14
2.5. Hierarchical PCE scenario	15
3. Motivations	17
3.1. Motivation for a YANG Model	17
3.1.1. Benefits of common data models	17
3.1.2. Benefits of a single interface	18
3.1.3. Extensibility	18
3.2. Interactions with TE topology	19
3.2.1. TE topology aggregation	20
3.2.2. TE topology abstraction	23
3.2.3. Complementary use of the TE topology	24
3.3. Path Computation RPC	26
3.3.1. Temporary reporting of the computed path state	28
4. Path computation and optimization for multiple paths	30
5. YANG data model for requesting Path Computation	31
5.1. Synchronization of multiple path computation requests	32
5.2. Returned metric values	34

5.3. Multiple Paths Requests for the same TE tunnel	35
5.3.1. Tunnel attributes specified by value	38
5.3.2. Tunnel attributes specified by reference	38
5.4. Multi-Layer Path Computation	41
6. YANG data model for TE path computation	42
6.1. Tree diagram	42
6.2. YANG module	53
7. Security Considerations	72
8. IANA Considerations	73
9. References	74
9.1. Normative References	74
9.2. Informative References	75
Appendix A. Examples	77
A.1. Basic Path Computation	77
A.2. Path Computation with transient state	77
A.3. Path Computation with Global Path Constraint	78
A.4. Path Computation with Per-tunnel Path Constraint	79
A.5. Path Computation result	80
Acknowledgments	81
Contributors	82
Authors' Addresses	83

1. Introduction

There are scenarios, typically in a hierarchical Software-Defined Networking (SDN) context, where the topology information provided by a Traffic Engineering (TE) network provider may be insufficient for its client to perform multi-domain path computation. In these cases the client would need to request the TE network provider to compute some intra-domain paths that could be used together with its topology information to compute the multi-domain path.

These types of scenarios can be applied to different interfaces in different reference architectures:

- * Application-Based Network Operations (ABNO) control interface [RFC7491], in which an Application Service Coordinator can request the ABNO controller to take in charge path calculation (see Figure 1 in [RFC7491]).
- * Abstraction and Control of TE Networks (ACTN) [RFC8453], where a controller hierarchy is defined. In the ACTN context, path computation is needed on the interface between Customer Network Controller (CNC) and Multi-Domain Service Coordinator (MDSC), called CNC-MDSC Interface (CMI), and on the interface between MDSC and Provisioning Network Controller (PNC), called MDSC-PNC Interface (MPI). [RFC8454] describes an information model for the Path Computation request.

Multiple protocol solutions can be used for communication between different controller hierarchical levels. This document assumes that the controllers are communicating using YANG-based protocols (e.g., NETCONF [RFC6241] or RESTCONF [RFC8040]).

Path Computation Elements (PCEs), controllers and orchestrators perform their operations based on Traffic Engineering Databases (TED). Such TEDs can be described, in a technology agnostic way, with the YANG data model for TE Topologies [RFC8795]. Furthermore, the technology specific details of the TED are modelled in the technology specific topology models, e.g., the [I-D.ietf-ccamp-otn-topo-yang] for Optical Transport Network (OTN) Optical Data Unit (ODU) technologies, which augment the common TE topology model in [RFC8795].

The availability of such topology models allows the provisioning of the TED using YANG-based protocols (e.g., NETCONF or RESTCONF). Furthermore, it enables a PCE/controller performing the necessary abstractions or modifications and offering this customized topology to another PCE/controller or high level orchestrator.

The tunnels that can be provided over the networks described with the topology models can be also set-up, deleted and modified via YANG-based protocols (e.g., NETCONF or RESTCONF) using the TE tunnel YANG data model [I-D.ietf-teas-yang-te].

This document defines a YANG data model [RFC7950] for an RPC to request path computation, which complements the solution defined in [I-D.ietf-teas-yang-te], to configure a TE tunnel path in "compute-only" mode.

The YANG data model definition does not make any assumption about whether that the client or the server implement a "PCE" functionality, as defined in [RFC4655], and the Path Computation Element Communication Protocol (PCEP) protocol, as defined in [RFC5440].

Moreover, this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

The YANG data model defined in this document conforms to the Network Management Datastore Architecture [RFC8342].

1.1. Terminology

TED:

The traffic engineering database is a collection of all TE information about all TE nodes and TE links in a given network.

PCE:

A Path Computation Element (PCE) is an entity that is capable of computing a network path or route based on a network graph, and of applying computational constraints during the computation. The PCE entity is an application that can be located within a network node or component, on an out-of-network server, etc. For example, a PCE would be able to compute the path of a TE Label Switched Path (LSP) by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. [RFC4655].

Domain:

TE information is the data relating to nodes and TE links that is used in the process of selecting a TE path. TE information is usually only available within a network. We call such a zone of visibility of TE information a domain. An example of a domain may be an IGP area or an Autonomous System. [RFC7926]

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

1.2. Tree Diagram

Tree diagrams used in this document follow the notation 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
inet	ietf-inet-types	[RFC6991]
te-types	ietf-te-types	[RFC8776]
te	ietf-te	[RFCYYYY]
te-pc	ietf-te-path-computation	RFCXXXX

Table 1: Prefixes and corresponding YANG modules

RFC Editor Note: Please replace XXXX with the RFC number assigned to this document. Please replace RFC YYYY with the RFC number of [I-D.ietf-teas-yang-te] once it has been published. Please remove this note.

2. Use Cases

This section presents some use cases, where a client needs to request underlying SDN controllers for path computation.

The use of the YANG data model defined in this document is not restricted to these use cases but can be used in any other use case when deemed useful.

The presented uses cases have been grouped, depending on the different underlying topologies: a) Packet/Optical Integration; b) multi-domain Traffic Engineered (TE) Networks; and c) Data Center Interconnections. Use cases d) and e) respectively present how to apply this YANG data model for standard multi-domain PCE i.e. Backward Recursive Path Computation [RFC5441] and Hierarchical PCE [RFC6805].

2.1. Packet/Optical Integration

In this use case, an optical domain is used to provide connectivity to some nodes of a packet domain (see Figure 1).

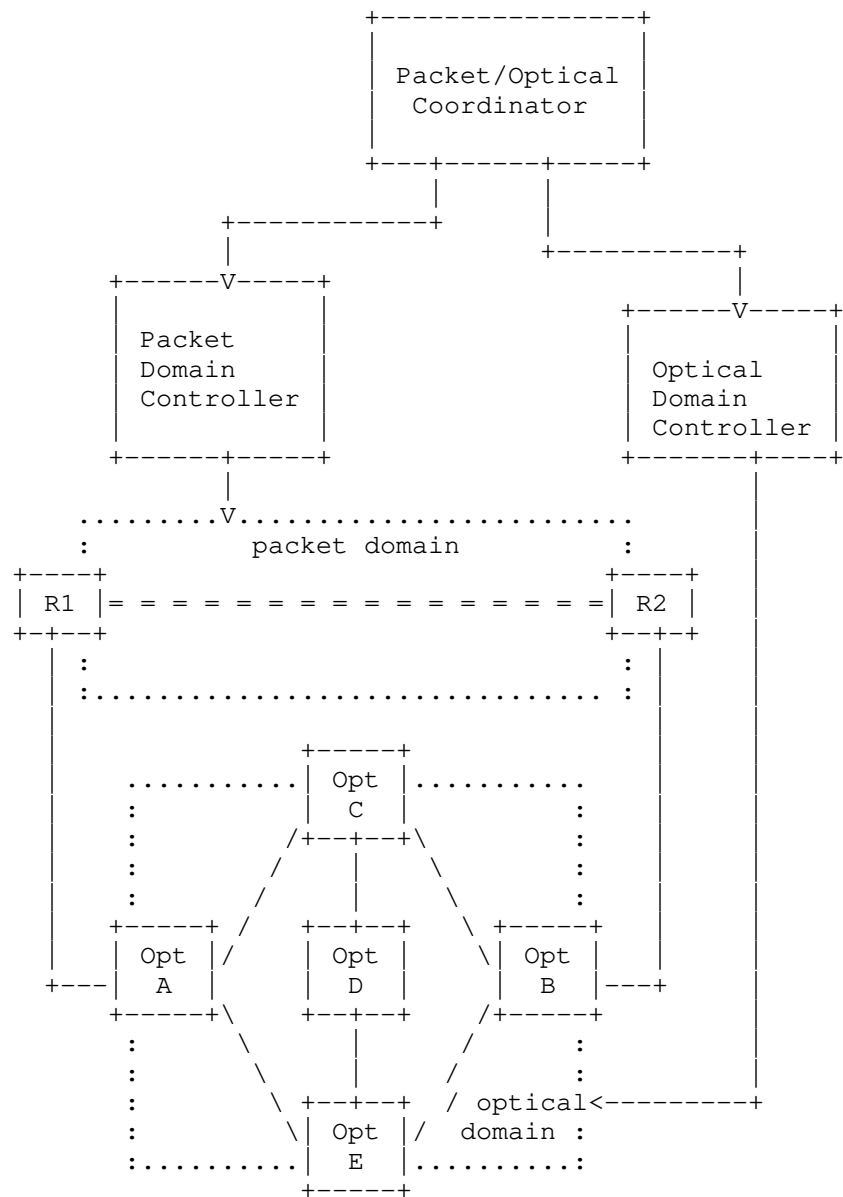


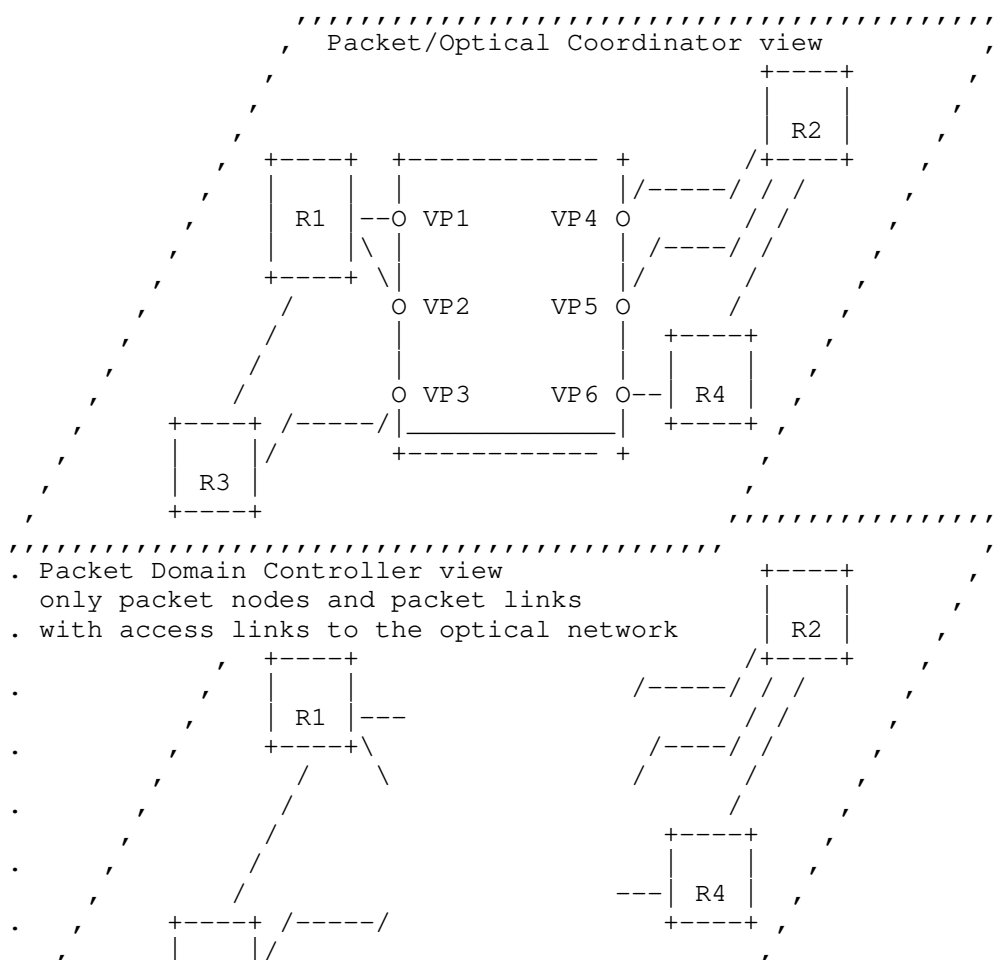
Figure 1: Packet/Optical Integration use case

Figure 1 as well as Figure 2 below only show a partial view of the packet network connectivity, before additional packet connectivity is provided by the optical network.

It is assumed that the Optical Domain Controller provides to the Packet/Optical Coordinator an abstracted view of the optical network. A possible abstraction could be to represent the whole optical network as one "virtual node" with "virtual ports" connected to the access links, as shown in Figure 2.

It is also assumed that Packet Domain Controller can provide the Packet/Optical Coordinator the information it needs to set up connectivity between packet nodes through the optical network (e.g., the access links).

The path computation request helps the Packet/Optical Coordinator to know the real connections that can be provided by the optical network.



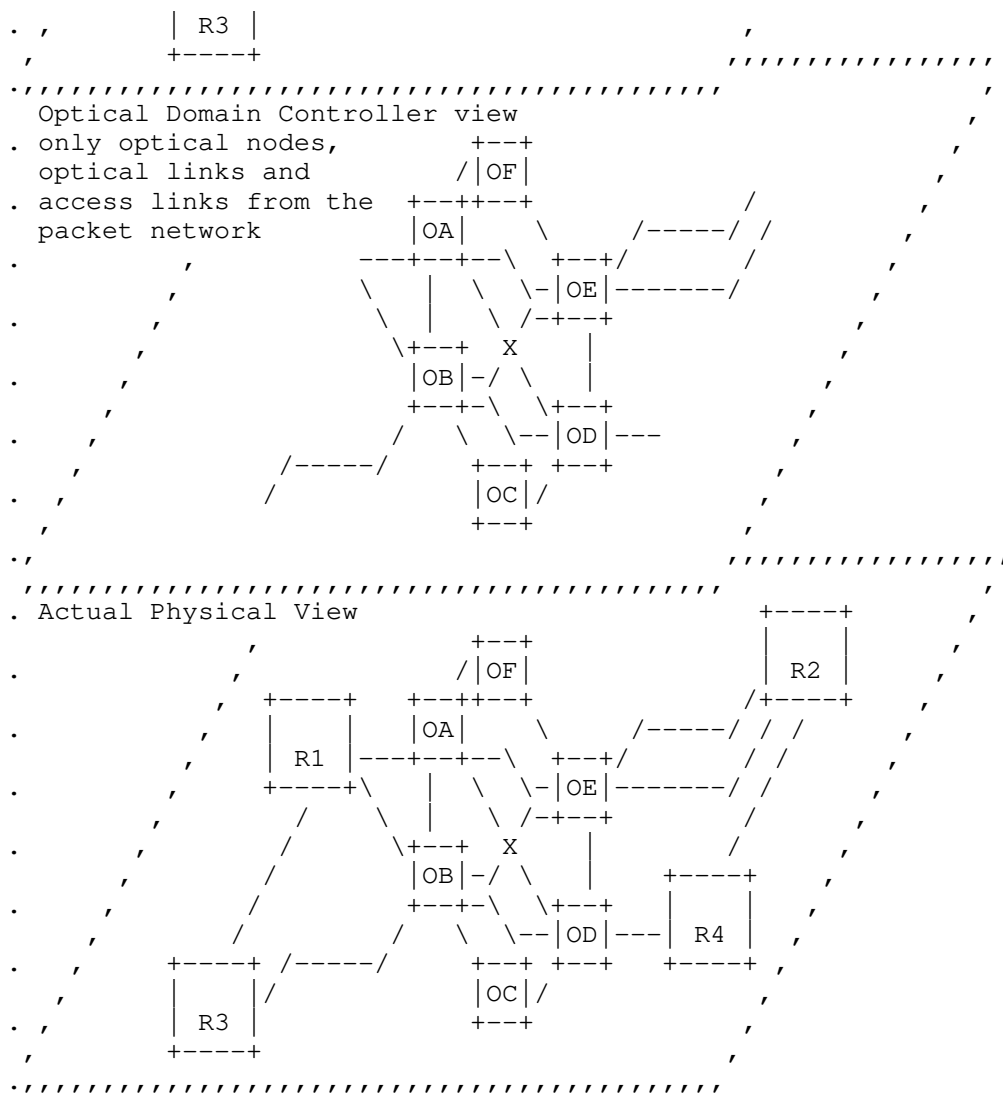


Figure 2: Packet and Optical Topology Abstractions

In this use case, the Packet/Optical Coordinator needs to set up an optimal underlying path for an IP link between R1 and R2.

As depicted in Figure 2, the Packet/Optical Coordinator has only an "abstracted view" of the physical network, and it does not know the feasibility or the cost of the possible optical paths (e.g., VP1-VP4 and VP2-VP5), which depend on the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The Packet/Optical Coordinator can request the underlying Optical Domain Controller to compute a set of potential optimal paths, taking into account optical constraints. Then, based on its own constraints, policy and knowledge (e.g. cost of the access links), it can choose which one of these potential paths to use to set up the optimal end-to-end path crossing optical network.

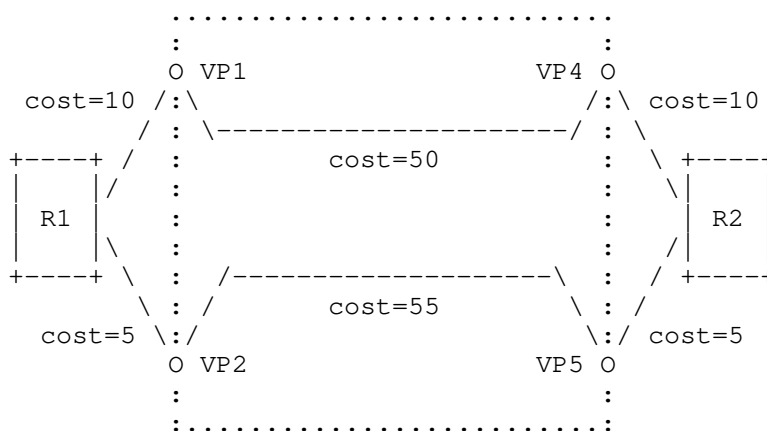


Figure 3: Packet/Optical Integration path computation example

For example, in Figure 3, the Packet/Optical Coordinator can request the Optical Domain Controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to set up the optimal end-to-end path using the VP2-VP5 optical path even if this is not the optimal path from the optical domain perspective.

Considering the dynamicity of the connectivity constraints of an optical domain, it is possible that a path computed by the Optical Domain Controller when requested by the Packet/Optical Coordinator is no longer valid/available when the Packet/Optical Coordinator requests it to be set up. This is further discussed in Section 3.3.

2.2. Multi-domain TE networks

In this use case there are two TE domains which are interconnected together by multiple inter-domains links.

A possible example could be a multi-domain optical network.

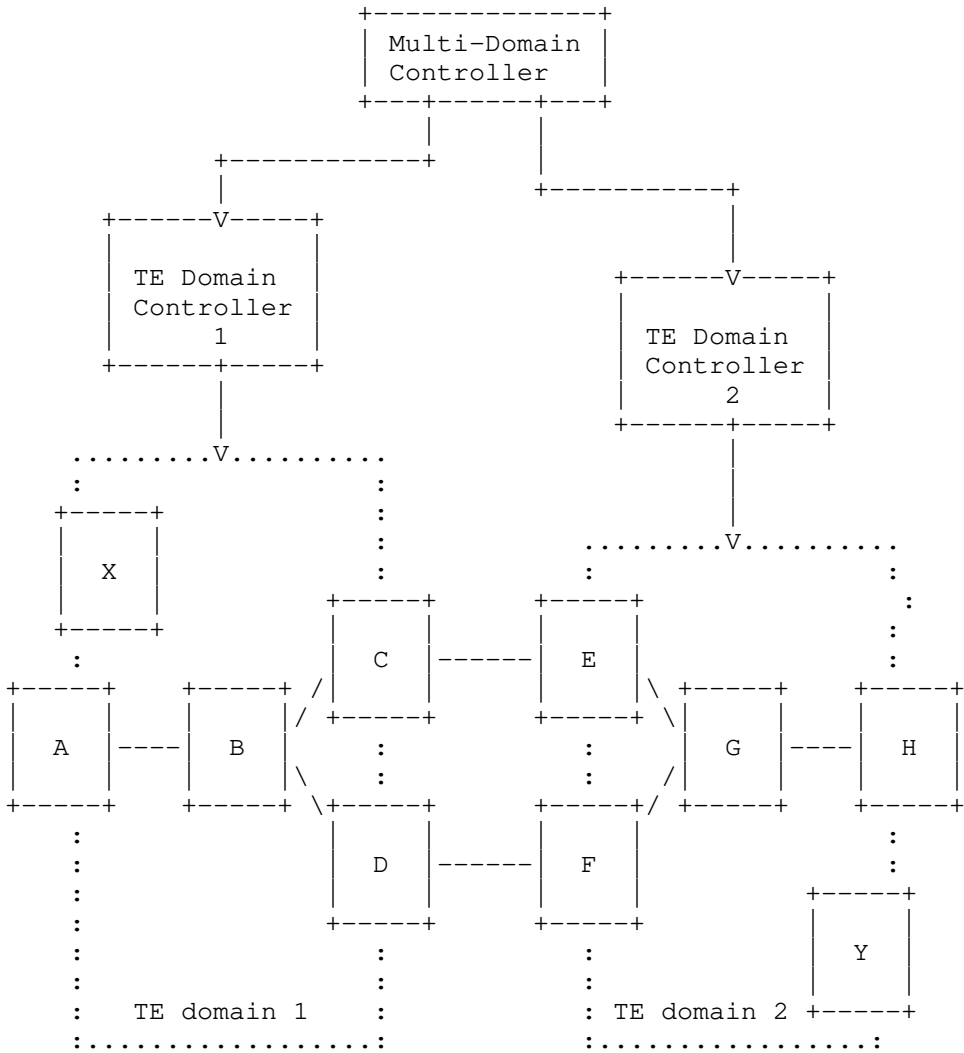


Figure 4: Multi-domain multi-link interconnection

In order to set up an end-to-end multi-domain TE path (e.g., between nodes A and H), the Multi-Domain Controller needs to know the feasibility or the cost of the possible TE paths within the two TE domains, which depend on the current status of the physical resources within each TE domain. This is more challenging in case of optical networks because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

In order to set up a multi-domain TE path (e.g., between nodes A and H), the Multi-Domain Controller can request the TE Domain Controllers to compute a set of intra-domain optimal paths and take decisions based on the information received. For example:

- * The Multi-Domain Controller asks TE Domain Controllers to provide set of paths between A-C, A-D, E-H and F-H
- * TE Domain Controllers return a set of feasible paths with the associated costs: the path A-C is not part of this set (in optical networks, it is typical to have some paths not being feasible due to optical constraints that are known only by the Optical Domain Controller)
- * The Multi-Domain Controller will select the path A-D-F-H since it is the only feasible multi-domain path and then request the TE Domain Controllers to set up the A-D and F-H intra-domain paths
- * If there are multiple feasible paths, the Multi-Domain Controller can select the optimal path knowing the cost of the intra-domain paths (provided by the TE domain controllers) and the cost of the inter-domain links (known by the Multi-Domain Controller)

This approach may have some scalability issues when the number of TE domains is quite big (e.g. 20).

In this case, it would be worthwhile using the abstract TE topology information provided by the TE Domain Controllers to limit the number of potential optimal end-to-end paths and then request path computation from fewer TE Domain Controllers in order to decide what the optimal path within this limited set is.

For more details, see Section 3.2.3.

2.3. Data Center Interconnections

In these use case, there is a TE domain which is used to provide connectivity between Data Centers which are connected with the TE domain using access links.

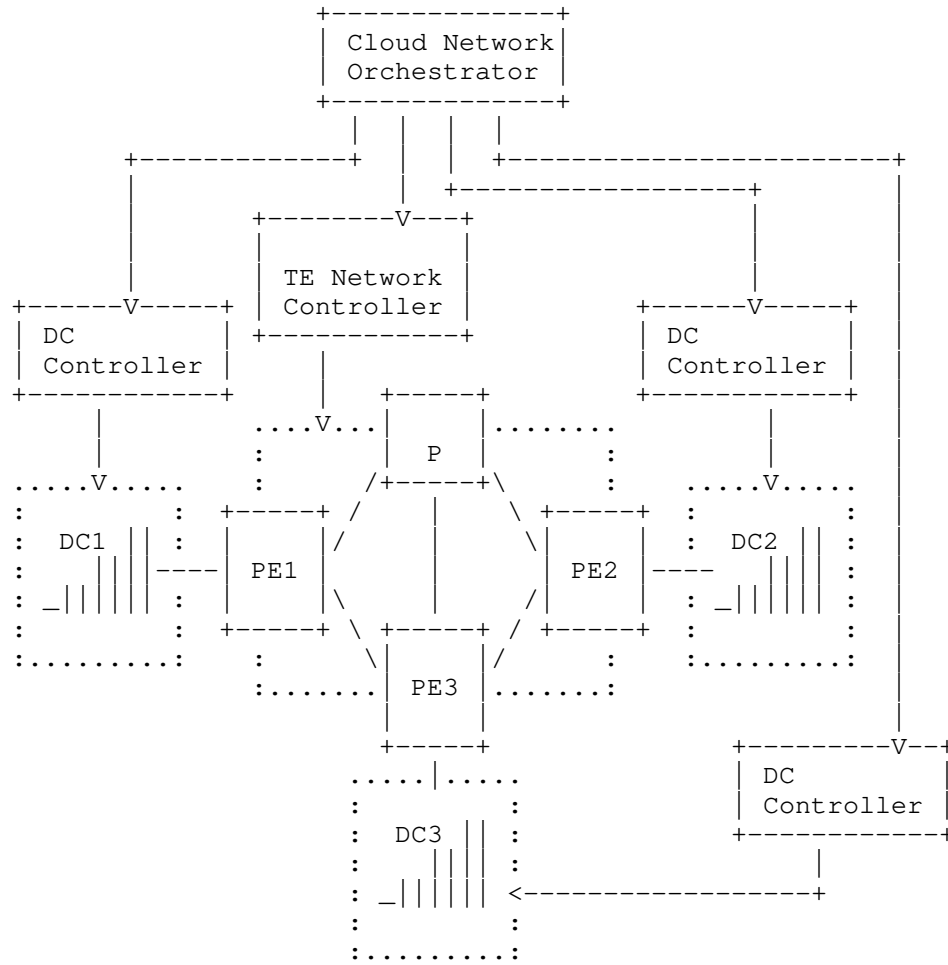


Figure 5: Data Center Interconnection use case

In this use case, there is the need to transfer data from Data Center 1 (DC1) to either DC2 or DC3 (e.g. workload migration).

The optimal decision depends both on the cost of the TE path (DC1-DC2 or DC1-DC3) and of the Data Center resources within DC2 or DC3.

The Cloud Network Orchestrator needs to make a decision for optimal connection based on TE network constraints and Data Center resources.

It may not be able to make this decision because it has only an abstract view of the TE network (as in Section 2.1).

The Cloud Network Orchestrator can request to the TE Network Controller to compute the cost of the possible TE paths (e.g., DC1-DC2 and DC1-DC3) and to the DC Controller to provide the information it needs about the required Data Center resources within DC2 and DC3 and then it can take the decision about the optimal solution based on this information and its policy.

2.4. Backward Recursive Path Computation scenario

[RFC5441] has defined the Virtual Source Path Tree (VSPT) TLV within PCE Reply Object in order to compute inter-domain paths following a "Backward Recursive Path Computation" (BRPC) method. The main principle is to forward the PCE request message up to the destination domain. Then, each PCE involved in the computation will compute its part of the path and send it back to the requester through PCE Response message. The resulting computation is spread from destination PCE to source PCE. Each PCE is in charge of merging the path it received with the one it calculated. At the end, the source PCE merges its local part of the path with the received one to achieve the end-to-end path.

Figure 6 below show a typical BRPC scenario where 3 PCEs cooperate to compute inter-domain paths.

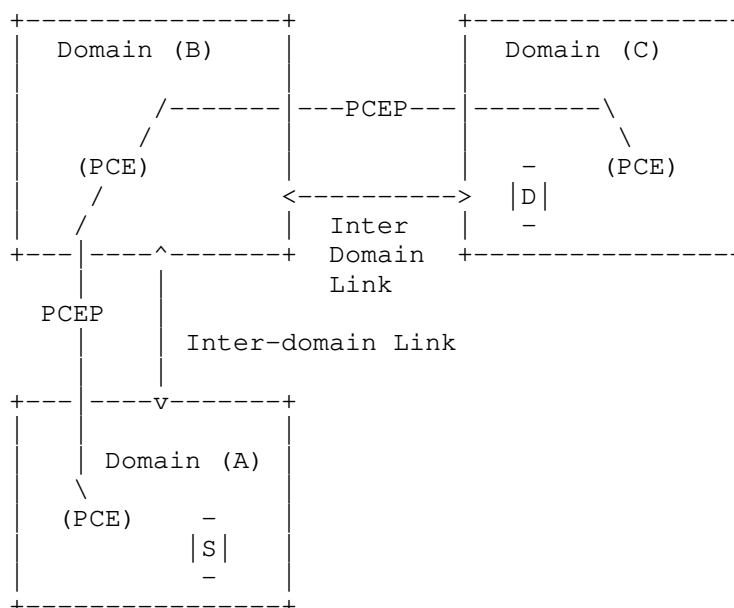


Figure 6: BRPC Scenario

In this use case, a client can use the YANG data model defined in this document to request path computation from the PCE that controls the source of the tunnel. For example, a client can request to the PCE of domain A to compute a path from a source S, within domain A, to a destination D, within domain C. Then PCE of domain A will use PCEP protocol, as per [RFC5441], to compute the path from S to D and in turn gives the final answer to the requester.

2.5. Hierarchical PCE scenario

[RFC6805] has defined an architecture and extensions to the PCE standard to compute inter-domain path following a hierarchical method. Two new roles have been defined: parent PCE and child PCE. The parent PCE is in charge to coordinate the end-to-end path computation. For that purpose it sends to each child PCE involved in the multi-domain path computation a PCE Request message to obtain the local part of the path. Once received all answer through PCE Response message, the parent PCE will merge the different local parts of the path to achieve the end-to-end path.

Figure 7 below shows a typical hierarchical scenario where a parent PCE request end-to-end path to the different child PCE. Note that a PCE could take independently the role of child or parent PCE depending of which PCE will request the path.

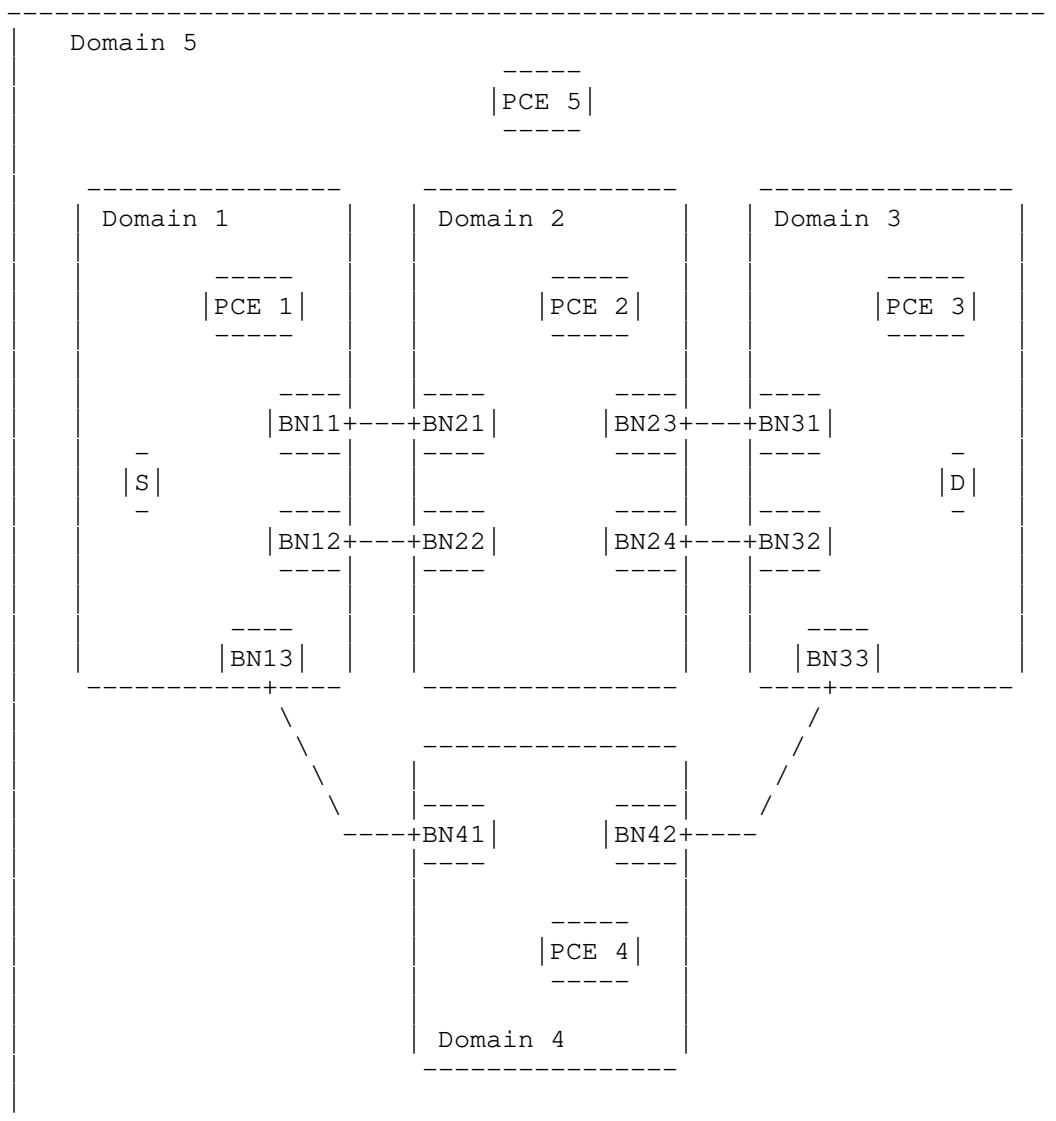


Figure 7: Hierarchical domain topology from RFC6805

In this use case, a client can use the YANG data model defined in this document to request to the parent PCE a path from a source S to a destination D. The parent PCE will in turn contact the child PCEs through PCEP protocol to compute the end-to-end path and then return the computed path to the client, using the YANG data model defined in this document. For example the YANG data model can be used to

request to PCE5 acting as parent PCE to compute a path from source S, within domain 1, to destination D, within domain 3. PCE5 will contact child PCEs of domain 1, 2 and 3 to obtain local part of the end-to-end path through the PCEP protocol. Once received the PCE Response message, it merges the answers to compute the end-to-end path and send it back to the client.

3. Motivations

This section provides the motivation for the YANG data model defined in this document.

Section 3.1 describes the motivation for a YANG data model to request path computation.

Section 3.2 describes the motivation for a YANG data model which complements the TE topology YANG data model defined in [RFC8795].

Section 3.3 describes the motivation for a YANG RPC which complements the TE tunnel YANG data model defined in [I-D.ietf-teas-yang-te].

3.1. Motivation for a YANG Model

3.1.1. Benefits of common data models

The YANG data model for requesting path computation is closely aligned with the YANG data models that provide (abstract) TE topology information, i.e., [RFC8795] as well as that are used to configure and manage TE tunnels, i.e., [I-D.ietf-teas-yang-te].

There are many benefits in aligning the data model used for path computation requests with the YANG data models used for TE topology information and for TE tunnels configuration and management:

- * There is no need for an error-prone mapping or correlation of information.
- * It is possible to use the same endpoint identifiers in path computation requests and in the topology modeling.
- * The attributes used for path computation constraints are the same as those used when setting up a TE tunnel.

3.1.2. Benefits of a single interface

The system integration effort is typically lower if a single, consistent interface is used by controllers, i.e., one data modeling language (i.e., YANG) and a common protocol (e.g., NETCONF or RESTCONF).

Practical benefits of using a single, consistent interface include:

1. Simple authentication and authorization: The interface between different components has to be secured. If different protocols have different security mechanisms, ensuring a common access control model may result in overhead. For instance, there may be a need to deal with different security mechanisms, e.g., different credentials or keys. This can result in increased integration effort.
2. Consistency: Keeping data consistent over multiple different interfaces or protocols is not trivial. For instance, the sequence of actions can matter in certain use cases, or transaction semantics could be desired. While ensuring consistency within one protocol can already be challenging, it is typically cumbersome to achieve that across different protocols.
3. Testing: System integration requires comprehensive testing, including corner cases. The more different technologies are involved, the more difficult it is to run comprehensive test cases and ensure proper integration.
4. Middle-box friendliness: Provider and consumer of path computation requests may be located in different networks, and middle-boxes such as firewalls, NATs, or load balancers may be deployed. In such environments it is simpler to deploy a single protocol. Also, it may be easier to debug connectivity problems.
5. Tooling reuse: Implementers may want to implement path computation requests with tools and libraries that already exist in controllers and/or orchestrators, e.g., leveraging the rapidly growing eco-system for YANG tooling.

3.1.3. Extensibility

Path computation is only a subset of the typical functionality of a controller. In many use cases, issuing path computation requests comes along with the need to access other functionality on the same system. In addition to obtaining TE topology, for instance also configuration of services (set-up/modification/deletion) may be required, as well as:

1. Receiving notifications for topology changes as well as integration with fault management
2. Performance management such as retrieving monitoring and telemetry data
3. Service assurance, e.g., by triggering OAM functionality
4. Other fulfilment and provisioning actions beyond tunnels and services, such as changing QoS configurations

YANG is a very extensible and flexible data modeling language that can be used for all these use cases.

3.2. Interactions with TE topology

The use cases described in Section 2 have been described assuming that the topology view exported by each underlying controller to its client is aggregated using the "virtual node model", defined in [RFC7926].

TE topology information, e.g., as provided by [RFC8795], could in theory be used by an underlying controller to provide TE information to its client thus allowing a PCE available within its client to perform multi-domain path computation on its own, without requesting path computations to the underlying controllers.

In case the client does not implement a PCE function, as discussed in Section 1, it could not perform path computation based on TE topology information and would instead need to request path computation from the underlying controllers to get the information it needs to find the optimal end-to-end path.

In case the client implements a PCE function, as discussed in Section 1, the TE topology information needs to be complete and accurate, which would bring to scalability issues.

Using TE topology to provide a "virtual link model" aggregation, as described in [RFC7926], may be insufficient, unless the aggregation provides complete and accurate information, which would still cause scalability issues, as described in Section 3.2.1 below.

Using TE topology abstraction, as described in [RFC7926], may lead to compute an unfeasible path, as described in [RFC7926] in Section 3.2.2 below.

Therefore when computing an optimal multi-domain path, there is a scalability trade-off between providing complete and accurate TE information and the number of path computation requests to the underlying controllers.

The TE topology information used, in a complimentary way, to reduce the number for path computation requests to the underlying controllers, are described in Section 3.2.3 below.

3.2.1. TE topology aggregation

Using the TE topology model, as defined in [RFC8795], the underlying controller can export the whole TE domain as a single TE node with a "detailed connectivity matrix" (which provides specific TE attributes, such as delay, Shared Risk Link Groups (SRLGs) and other TE metrics, between each ingress and egress links).

The information provided by the "detailed connectivity matrix" would be equivalent to the information that should be provided by "virtual link model" as defined in [RFC7926].

For example, in the Packet/Optical Integration use case, described in Section 2.1, the Optical Domain Controller can make the information shown in Figure 3 available to the Packet/Optical Coordinator as part of the TE topology information and the Packet/Optical Coordinator could use this information to calculate on its own the optimal path between R1 and R2, without requesting any additional information to the Optical Domain Controller.

However, when designing the amount of information to provide within the "detailed connectivity matrix", there is a tradeoff to be considered between accuracy (i.e., providing "all" the information that might be needed by the PCE available within the client) and scalability.

Figure 8 below shows another example, similar to Figure 3, where there are two possible Optical paths between VP1 and VP4 with different properties (e.g., available bandwidth and cost).



Figure 8: Packet/Optical Integration path computation Example with multiple choices

If the information in the "detailed connectivity matrix" is not complete/accurate, we can have the following drawbacks:

- * If only the VP1-VP4 path with available bandwidth of 2 Gb/s and cost 50 is reported, the client's PCE will fail to compute a 5 Gb/s path between routers R1 and R2, although this would be feasible;
- * If only the VP1-VP4 path with available bandwidth of 10 Gb/s and cost 60 is reported, the client's PCE will compute, as optimal, the 1 Gb/s path between R1 and R2 going through the VP2-VP5 path within the optical domain while the optimal path would actually be the one going through the VP1-VP4 sub-path (with cost 50) within the optical domain.

Reporting all the information, as in Figure 8, using the "detailed connectivity matrix", is quite challenging from a scalability perspective. The amount of this information is not just based on number of end points (which would scale as N-square), but also on many other parameters, including client rate, user constraints/policies for the service, e.g. max latency < N ms, max cost, etc., exclusion policies to route around busy links, min Optical Signal to Noise Ratio (OSNR) margin, max pre-Forward Error Correction (FEC) Bit Error Rate (BER) etc. All these constraints could be different based on connectivity requirements.

It is also worth noting that the "connectivity matrix" has been originally defined in Wavelength Switched Optical Networks (WSO), [RFC7446], to report the connectivity constraints of a physical node within the Wavelength Division Multiplexing (WDM) network: the information it contains is pretty "static" and therefore, once taken and stored in the TE data base, it can be always being considered valid and up-to-date in path computation request.

The "connectivity matrix" is sometimes confused with "optical reach table" that contain multiple (e.g. k-shortest) regen-free reachable paths for every A-Z node combination in the network. Optical reach tables can be calculated offline, utilizing vendor optical design and planning tools, and periodically uploaded to the Controller: these optical path reach tables are fairly static. However, to get the connectivity matrix, between any two sites, either a regen free path can be used, if one is available, or multiple regen free paths are concatenated to get from the source to the destination, which can be a very large combination. Additionally, when the optical path within optical domain needs to be computed, it can result in different paths based on input objective, constraints, and network conditions. In summary, even though "optical reach table" is fairly static, which regen free paths to build the connectivity matrix between any source and destination is very dynamic, and is done using very sophisticated routing algorithms.

Using the "basic connectivity matrix" with an abstract node to abstract the information regarding the connectivity constraints of an Optical domain, would make this information more "dynamic" since the connectivity constraints of an optical domain can change over time because some optical paths that are feasible at a given time may become unfeasible at a later time when e.g., another optical path is established.

The information in the "detailed connectivity matrix" is even more dynamic since the establishment of another optical path may change some of the parameters (e.g., delay or available bandwidth) in the "detailed connectivity matrix" while not changing the feasibility of the path.

There is therefore the need to keep the information in the "detailed connectivity matrix" updated which means that there another tradeoff between the accuracy (i.e., providing "all" the information that might be needed by the client's PCE) and having up-to-date information. The more the information is provided and the longer it takes to keep it up-to-date which increases the likelihood that the client's PCE computes paths using not updated information.

It seems therefore quite challenging to have a "detailed connectivity matrix" that provides accurate, scalable and updated information to allow the client's PCE to take optimal decisions by its own.

Considering the example in Figure 8 with the approach defined in this document, the client, when it needs to set up an end-to-end path, it can request the Optical Domain Controller to compute a set of optimal paths (e.g., for VP1-VP4 and VP2-VP5) and take decisions based on the information received:

- * When setting up a 5 Gb/s path between routers R1 and R2, the Optical Domain Controller may report only the VP1-VP4 path as the only feasible path: the Packet/Optical Coordinator can successfully set up the end-to-end path passing through this optical path;
- * When setting up a 1 Gb/s path between routers R1 and R2, the Optical Domain Controller (knowing that the path requires only 1 Gb/s) can report both the VP1-VP4 path, with cost 50, and the VP2-VP5 path, with cost 65. The Packet/Optical Coordinator can then compute the optimal path which is passing through the VP1-VP4 sub-path (with cost 50) within the optical domain.

3.2.2. TE topology abstraction

Using the TE topology model, as defined in [RFC8795], the underlying controller can export to its client an abstract TE topology, composed by a set of TE nodes and TE links, representing the abstract view of the topology under its control.

For example, in the multi-domain TE network use case, described in Section 2.2, the TE Domain Controller 1 can export a TE topology encompassing the TE nodes A, B, C and D and the TE links interconnecting them. In a similar way, the TE Domain Controller 2 can export a TE topology encompassing the TE nodes E, F, G and H and the TE links interconnecting them.

In this example, for simplicity reasons, each abstract TE node maps with each physical node, but this is not necessary.

In order to set up a multi-domain TE path (e.g., between nodes A and H), the Multi-Domain Controller can compute by its own an optimal end-to-end path based on the abstract TE topology information provided by the domain controllers. For example:

- * Multi-Domain Controller can compute, based on its own TED data, the optimal multi-domain path being A-B-C-E-G-H, and then request the TE Domain Controllers to set up the A-B-C and E-G-H intra-domain paths
- * But, during path set-up, the TE Domain Controller may find out that A-B-C intra-domain path is not feasible (as discussed in Section 2.2, in optical networks it is typical to have some paths not being feasible due to optical constraints that are known only by the Optical Domain Controller), while only the path A-B-D is feasible
- * So what the Multi-Domain Controller has computed is not good and it needs to re-start the path computation from scratch

As discussed in Section 3.2.1, providing more extensive abstract information from the TE Domain Controllers to the Multi-Domain Controller may lead to scalability problems.

In a sense this is similar to the problem of routing and wavelength assignment within an optical domain. It is possible to do first routing (step 1) and then wavelength assignment (step 2), but the chances of ending up with a good path is low. Alternatively, it is possible to do combined routing and wavelength assignment, which is known to be a more optimal and effective way for optical path set-up. Similarly, it is possible to first compute an abstract end-to-end path within the Multi-Domain Controller (step 1) and then compute an intra-domain path within each optical domain (step 2), but there are more chances not to find a path or to get a suboptimal path than by performing multiple per-domain path computations and then stitching them together.

3.2.3. Complementary use of the TE topology

As discussed in Section 2.2, there are some scalability issues with path computation requests in a multi-domain TE network with many TE domains, in terms of the number of requests to send to the TE Domain Controllers. It would therefore be worthwhile using the abstract TE topology information provided by the TE Domain Controllers to limit the number of requests.

An example can be described considering the multi-domain abstract TE topology shown in Figure 9. In this example, an end-to-end TE path between domains A and F needs to be set up. The transit TE domain should be selected between domains B, C, D and E.

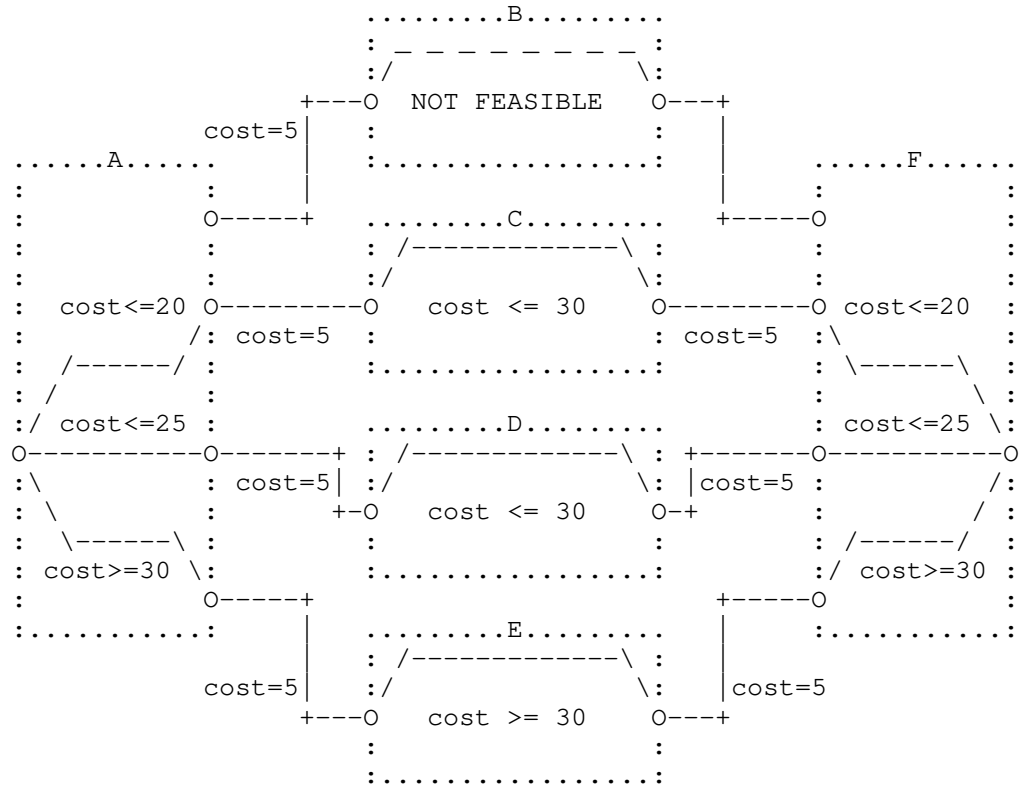


Figure 9: Multi-domain with many domains (Topology information)

The actual cost of each intra-domain path is not known a priori from the abstract topology information. The Multi-Domain Controller only knows, from the TE topology provided by the underlying domain controllers, the feasibility of some intra-domain paths and some upper-bound and/or lower-bound cost information. With this information, together with the cost of inter-domain links, the Multi-Domain Controller can understand by its own that:

- * Domain B cannot be selected as the path connecting domains A and F is not feasible;
- * Domain E cannot be selected as a transit domain since it is known from the abstract topology information provided by domain controllers that the cost of the multi-domain path A-E-F (which is 100, in the best case) will be always be higher than the cost of the multi-domain paths A-D-F (which is 90, in the worst case) and A-C-F (which is 80, in the worst case).

Therefore, the Multi-Domain Controller can understand by its own that the optimal multi-domain path could be either A-D-F or A-C-F but it cannot know which one of the two possible option actually provides the optimal end-to-end path.

The Multi-Domain Controller can therefore request path computation only to the TE Domain Controllers A, D, C and F (and not to all the possible TE Domain Controllers).

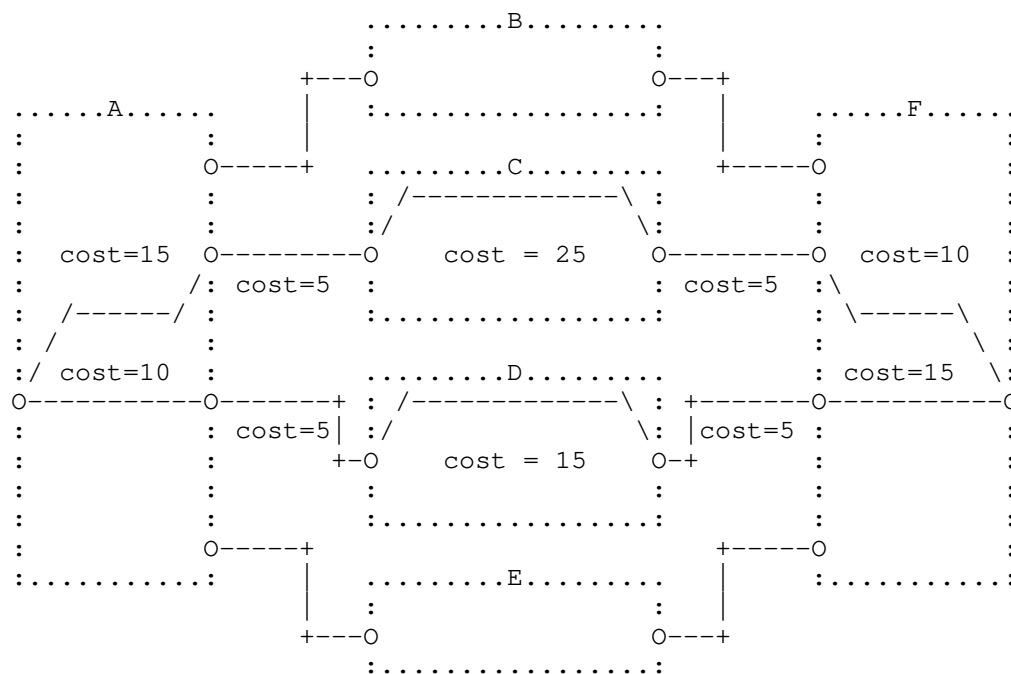


Figure 10: Multi-domain with many domains (Path Computation information)

Based on these requests, the Multi-Domain Controller can know the actual cost of each intra-domain paths which belongs to potential optimal end-to-end paths, as shown in Figure 10, and then compute the optimal end-to-end path (e.g., A-D-F, having total cost of 50, instead of A-C-F having a total cost of 70).

3.3. Path Computation RPC

The TE tunnel YANG data model, defined in [I-D.ietf-teas-yang-te], can support the need to request path computation, as described in section 5.1.2 of [I-D.ietf-teas-yang-te].

This solution is stateful since the state of each created "compute-only" TE tunnel path needs to be maintained, in the YANG datastores (at least in the running datastore and operational datastore), and updated, when underlying network conditions change.

The RPC mechanism allows requesting path computation using a simple atomic operation, without creating any state in the YANG datastores, and it is the natural option/choice, especially with stateless PCE.

It is very useful to provide both the options of using an RPC as well as of setting up TE tunnel paths in "compute-only" mode. It is suggested to use the RPC as much as possible and to rely on "compute-only" TE tunnel paths, when really needed.

Using the RPC solution would imply that the underlying controller (e.g., a PNC) computes a path twice during the process to set up an LSP: at time T1, when its client (e.g., an MDSC) sends a path computation RPC request to it, and later, at time T2, when the same client (MDSC) creates a TE tunnel requesting the set-up of the LSP. The underlying assumption is that, if network conditions have not changed, the same path that has been computed at time T1 is also computed at time T2 by the underlying controller (e.g. PNC) and therefore the path that is set up at time T2 is exactly the same path that has been computed at time T1.

However, since the operation is stateless, there is no guarantee that the returned path would still be available when path set-up is requested: this does not cause major issues when the time between path computation and path set-up is short (especially if compared with the time that would be needed to update the information of a very detailed connectivity matrix).

In most of the cases, there is even no need to guarantee that the path that has been set up is the exactly same as the path that has been returned by path computation, especially if it has the same or even better metrics. Depending on the abstraction level applied by the server, the client may also not know the actual computed path.

The most important requirement is that the required global objectives (e.g., multi-domain path metrics and constraints) are met. For this reason a path verification phase is always necessary to verify that the actual path that has been set up meets the global objectives (for example in a multi-domain network, the resulting end-to-end path meets the required end-to-end metrics and constraints).

In most of the cases, even if the path being set up is not exactly the same as the path returned by path computation, its metrics and constraints are "good enough" and the path verification passes

successfully. In the few corner cases where the path verification fails, it is possible repeat the whole process (path computation, path set-up and path verification).

In case it is required to set up at T2 exactly the same path computed at T1, the RPC solution should not be used and, instead, a "compute-only" TE tunnel path should be set up, allowing also notifications in case the computed path has been changed.

In this case, at time T1, the client (MDSC) creates a TE tunnel in a compute-only mode in the running datastore and later, at time T2, changes the configuration of that TE tunnel (not to be any more in a compute-only mode) to trigger the set-up of the LSP over the path which have been computed at time T1 and reported in the operational datastore.

It is worth noting that also using the "compute-only" TE tunnel path, although increasing the likelihood that the computed path is available at path set-up, does not guarantee that because notifications may not be reliable or delivered on time. Path verification is needed also in this case.

The solution based on "compute-only" TE tunnel path has also the following drawbacks:

- * Several messages required for any path computation
- * Requires persistent storage in the underlying controller
- * Need for garbage collection for stranded paths
- * Process burden to detect changes on the computed paths in order to provide notifications update

3.3.1. Temporary reporting of the computed path state

This section describes an optional extension to the stateless behavior of the path computation RPC, where the underlying controller, after having received a path computation RPC request, maintains some "transient state" associated with the computed path, allowing the client to request the set-up of exactly that path, if still available.

This is similar to the "compute-only" TE tunnel path solution but, to avoid the drawbacks of the stateful approach, is leveraging the path computation RPC and the separation between configuration and operational datastore, as defined in the NMDA architecture [RFC8342].

The underlying controller, after having computed a path, as requested by a path computation RPC, also creates a TE tunnel instance within the operational datastore, to store that computed path. This would be similar to a "compute-only" TE tunnel path, with the only difference that there is no associated TE tunnel instance within the running datastore.

Since the underlying controller stores in the operational datastore the computed path based on an abstract topology it exposes, it also remembers, internally, which is the actual native path (physical path), within its native topology (physical topology), associated with that compute-only TE tunnel instance.

Afterwards, the client (e.g., MDSC) can request the set-up of that specific path by creating a TE tunnel instance (not in compute-only mode) in the running datastore using the same tunnel-name of the existing TE tunnel in the operational datastore: this will trigger the underlying controller to set up that path, if still available.

There are still cases where the path being set up is not exactly the same as the path that has been computed:

- * When the tunnel is configured with path constraints which are not compatible with the computed path;
- * When the tunnel set-up is requested after the resources of the computed path are no longer available;
- * When the tunnel set-up is requested after the computed path is no longer known (e.g. due to a server reboot) by the underlying controller.

In all these cases, the underlying controller should compute and set up a new path.

Therefore the "path verification" phase, as described in Section 3.3 above, is always needed to check that the path that has been set up is still "good enough".

Since this new approach is not completely stateless, garbage collection is implemented using a timeout that, when it expires, triggers the removal of the computed path from the operational datastore. This operation is fully controlled by the underlying controller without the need for any action to be taken by the client that is not able to act on the operational datastore. The default value of this timeout is 10 minutes but a different value may be configured by the client.

In addition, it is possible for the client to tag each path computation request with a transaction-id allowing for a faster removal of all the paths associated with a transaction-id, without waiting for their timers to expire.

The underlying controller can remove from the operational datastore all the paths computed with a given transaction-id which have not been set up either when it receives a Path Delete RPC request for that transaction-id or, automatically, right after the set-up of a path that has been previously computed with that transaction-id.

This possibility is useful when multiple paths are computed but, at most, only one is set up (e.g., in multi-domain path computation scenario scenarios). After the selected path has been set up (e.g., in one domain during multi-domain path set-up), all the other alternative computed paths can be automatically deleted by the underlying controller (since no longer needed). The client can also request, using the Path Delete RPC request, the underlying controller to remove all the computed paths, if none of them is going to be set up (e.g., in a transit domain not being selected by multi-domain path computation and so not being automatically deleted).

This approach is complimentary and not alternative to the timer which is always needed to avoid stranded computed paths being stored in the operational datastore when no path is set up and no explicit Path Delete RPC request is received.

4. Path computation and optimization for multiple paths

There are use cases, where it is advantageous to request path computation for a set of paths, through a network or through a network domain, using a single request [RFC5440].

In this case, sending a single request for multiple path computations, instead of sending multiple requests for each path computation, would reduce the protocol overhead and it would consume less resources (e.g., threads in the client and server).

In the context of a typical multi-domain TE network, there could be multiple choices for the ingress/egress points of a domain and the Multi-Domain Controller needs to request path computation between all the ingress/egress pairs to select the best pair. For example, in the example of Section 2.2, the Multi-Domain Controller needs to request the TE Network Controller 1 to compute the A-C and the A-D paths and to the TE Network Controller 2 to compute the E-H and the F-H paths.

It is also possible that the Multi-Domain Controller receives a request to set up a group of multiple end to end connections. The Multi-Domain Controller needs to request each TE domain Controller to compute multiple paths, one (or more) for each end to end connection.

There are also scenarios where it can be needed to request path computation for a set of paths in a synchronized fashion.

One example could be computing multiple diverse paths. Computing a set of diverse paths in an unsynchronized fashion, leads to the possibility of not being able to satisfy the diversity requirement. In this case, it is preferable to compute a sub-optimal primary path for which a diversely routed secondary path exists.

There are also scenarios where it is needed to request optimizing a set of paths using objective functions that apply to the whole set of paths, see [RFC5541], e.g. to minimize the sum of the costs of all the computed paths in the set.

5. YANG data model for requesting Path Computation

This document define a YANG RPC to request path computation as an "augmentation" of tunnel-rpc, defined in [I-D.ietf-teas-yang-te]. This model provides the RPC input attributes that are needed to request path computation and the RPC output attributes that are needed to report the computed paths.

```

augment /te:tunnels-path-compute/te:input/te:path-compute-info:
  +-- path-request* [request-id]
  |   +-- request-id                               uint32
  |   .....
  +--ro response* [response-id]
  |   +--ro response-id                           uint32
  |   +--ro computed-paths-properties
  |       +--ro computed-path-properties* [k-index]
  |           +--ro k-index                       uint8
  |           +--ro path-properties
  |           .....

```

This model extensively re-uses the grouping defined in [I-D.ietf-teas-yang-te] and [RFC8776] to ensure maximal syntax and semantics commonality.

This YANG data model allows one RPC to include multiple path requests, each path request being identified by a request-id. Therefore, one RPC can return multiple responses, one for each path

request, being identified by a response-id equal to the corresponding request-id. Each response reports one or more computed paths, as requested by the k-requested-paths attribute. By default, each response reports one computed path.

5.1. Synchronization of multiple path computation requests

The YANG data model permits the synchronization of a set of multiple path requests (identified by specific request-id) all related to a "svec" container emulating the syntax of the Synchronization VECTOR (SVEC) PCEP object, defined in [RFC5440].

```
+-- synchronization* []
|   +-- svec
|   |   +-- relaxable?          boolean
|   |   +-- disjointness?      te-path-disjointness
|   |   +-- request-id-number*  uint32
|   +-- svec-constraints
|   |   +-- path-metric-bound* [metric-type]
|   |   |   +-- metric-type    identityref
|   |   |   +-- upper-bound?   uint64
|   +-- path-srlgs-lists
|   |   +-- path-srlgs-list* [usage]
|   |   |   +-- usage          identityref
|   |   |   +-- values*        srlg
|   +-- path-srlgs-names
|   |   +-- path-srlgs-name* [usage]
|   |   |   +-- usage          identityref
|   |   |   +-- names*         string
|   +-- exclude-objects
|   |   +-- excludes* []
|   |   |   +-- (type)?
|   |   |   |   +--:(numbered-node-hop)
|   |   |   |   |   +-- numbered-node-hop
|   |   |   |   |   |   +-- node-id      te-node-id
|   |   |   |   |   |   +-- hop-type?    te-hop-type
|   |   |   |   +--:(numbered-link-hop)
|   |   |   |   |   +-- numbered-link-hop
|   |   |   |   |   |   +-- link-tp-id    te-tp-id
|   |   |   |   |   |   +-- hop-type?     te-hop-type
|   |   |   |   |   |   +-- direction?    te-link-direction
|   |   |   |   +--:(unnumbered-link-hop)
|   |   |   |   |   +-- unnumbered-link-hop
|   |   |   |   |   |   +-- link-tp-id    te-tp-id
|   |   |   |   |   |   +-- node-id      te-node-id
|   |   |   |   |   |   +-- hop-type?    te-hop-type
|   |   |   |   |   |   +-- direction?    te-link-direction
|   |   |   +--:(as-number)
```

```

|
|   +-- as-number-hop
|   |   +-- as-number      inet:as-number
|   |   +-- hop-type?      te-hop-type
|   +--:(label)
|   |   +-- label-hop
|   |   |   +-- te-label
|   |   |   |   +-- (technology)?
|   |   |   |   |   +--:(generic)
|   |   |   |   |   |   +-- generic?
|   |   |   |   |   |   |   rt-types:generalized-label
|   |   |   +-- direction?      te-label-direction
+-- optimizations
|   +-- (algorithm)?
|   |   +--:(metric) {te-types:path-optimization-metric}?
|   |   |   +-- optimization-metric* [metric-type]
|   |   |   |   +-- metric-type      identityref
|   |   |   |   +-- weight?          uint8
|   |   +--:(objective-function)
|   |   |   {te-types:path-optimization-objective-
function}?
|   |   |   +-- objective-function
|   |   |   |   +-- objective-function-type?      identityref

```

The model, in addition to the metric types, defined in [RFC8776], which can be applied to each individual path request, supports also additional metric types, which apply to a set of synchronized requests, as referenced in [RFC5541]. These additional metric types are defined by the following YANG identities:

- * svec-metric-type: base YANG identity from which cumulative metric types identities are derived.
- * svec-metric-cumul-te: cumulative TE cost metric type, as defined in [RFC5541].
- * svec-metric-cumul-igp: cumulative IGP cost metric type, as defined in [RFC5541].
- * svec-metric-cumul-hop: cumulative Hop metric type, representing the cumulative version of the Hop metric type defined in [RFC8776].
- * svec-metric-aggregate-bandwidth-consumption: aggregate bandwidth consumption metric type, as defined in [RFC5541].
- * svec-metric-load-of-the-most-loaded-link: load of the most loaded link metric type, as defined in [RFC5541].

5.2. Returned metric values

This YANG data model provides a way to return the values of the metrics computed by the path computation in the output of RPC, together with other important information (e.g. srlg, affinities, explicit route), emulating the syntax of the "C" flag of the "METRIC" PCEP object [RFC5440]:

```

+--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 te-bandwidth
|   +--ro (technology)?
|   |   +--:(generic)
|   |   |   +--ro generic?    te-bandwidth
+--ro disjointness-type?
|   te-types:te-path-disjointness

```

It also allows the client to request which information (metrics, srlg and/or affinities) should be returned:

```

+-- request-id                                uint32
.....
+-- requested-metrics* [metric-type]
|   +-- metric-type      identityref
+-- return-srlgs?                                boolean
+-- return-affinities?                            boolean
.....

```

This feature is essential for path computation in a multi-domain TE network as described in Section 2.2. In this case, the metrics returned by a path computation requested to a given underlying controller must be used by the client to compute the best end-to-end path. If they are missing, the client cannot compare different paths calculated by the underlying controllers and choose the best one for the optimal end-to-end (e2e) path.

5.3. Multiple Paths Requests for the same TE tunnel

The YANG data model allows including multiple requests for different paths intended to be used within the same tunnel or within different tunnels.

When multiple requested paths are intended to be used within the same tunnel (e.g., requesting path computation for the primary and secondary paths of a protected tunnel), the set of attributes that are intended to be configured on per-tunnel basis rather than on per-path basis are common to all these path requests. These attributes includes both attributes which can be configured only a per-tunnel basis (e.g., tunnel-name, source/destination TTP, encoding and switching-type) as well attributes which can be configured both on a per-tunnel and on a per-path basis (e.g., the te-bandwidth or the associations).

Therefore, a tunnel-attributes list is defined, within the path computation request RPC:

```

+-- tunnel-attributes* [tunnel-name]
|   +-- tunnel-name          string
|   +-- encoding?            identityref
|   +-- switching-type?      identityref
|   +-- source?              te-types:te-node-id
|   +-- destination?        te-types:te-node-id
|   +-- src-tunnel-tp-id?    binary
|   +-- dst-tunnel-tp-id?    binary
|   +-- bidirectional?       boolean
|   +-- association-objects
|   .....
|   +-- protection-type?     identityref
|   +-- restoration-type?    identityref
|   +-- restoration-scheme?  identityref
|   +-- te-topology-identifier
|   |   +-- provider-id?     te-global-id
|   |   +-- client-id?       te-global-id
|   |   +-- topology-id?     te-topology-id
|   +-- te-bandwidth
|   |   +-- (technology)?
|   |   |   +--:(generic)
|   |   |   +-- generic?     te-bandwidth
|   +-- link-protection?     identityref
|   +-- setup-priority?      uint8
|   +-- hold-priority?       uint8
|   +-- signaling-type?      identityref
|   +-- hierarchy
|   |   +-- dependency-tunnels
|   |   .....
|   |   +-- hierarchical-link
|   |   .....

```

The path requests that are intended to be used within the same tunnel should reference the same entry in the tunnel-attributes list. This allows:

- * avoiding repeating the same set of per-tunnel parameters on multiple requested paths;
- * the server to understand what attributes are intended to be configured on a per-tunnel basis (e.g., the te-bandwidth configured in the tunnel-attributes) and what attributes are intended to be configured on a per-path basis (e.g., the te-bandwidth configured in the path-request). This could be useful especially when the server also creates a TE tunnel instance within the operational datastore to report the computed paths, as described in Section 3.3.1: in this case, the tunnel-name is also used as the suggested name for that TE tunnel instance.

The YANG data model allows also including requests for paths intended to modify existing tunnels (e.g., adding a protection path for an existing un-protected tunnel). In this case, the per-tunnel attributes are already provided in an existing TE tunnel instance and do not need to be re-configured in the path computation request RPC. Therefore, these requests should reference an existing TE tunnel instance.

It is also possible to request computing paths without indicating in which tunnel they are intended to be used (e.g., in case of an unprotected tunnel). In this case, the per-tunnel attributes could be provided together with the per-path attributes in the path request, without using the tunnel-attributes list.

The choices below are defined to distinguish the cases above:

- * whether the per-tunnel attributes are configured by reference (providing a leafref), to:
 - either a TE tunnel instance, if it exists;
 - or to an entry of the tunnel-attributes list, if the TE tunnel instance does not exist;
- * or by value, providing the set of tunnel attributes within the path request:

```

+-- (tunnel-attributes)?
+--:(reference)
+-- tunnel-reference
+-- (tunnel-exist)?
+--:(tunnel-ref)
+-- tunnel-ref                                te:tunnel-ref
+--:(tunnel-attributes-ref)
+-- tunnel-attributes-ref                    leafref
+-- path-name?                               string
+-- (path-role)
+--:(primary-path)
.....
+--:(value)
+-- tunnel-name?                             string
.....
+-- encoding?                                identityref
+-- switching-type?                           identityref
.....

```

5.3.1. Tunnel attributes specified by value

The (value) case provides the set of attributes that are configured only on per-tunnel basis (e.g., tunnel-name, source/destination TTP, encoding and switching-type).

In this case, it is assumed that the requested path will be the only path within a tunnel.

If the requested path is a transit segment of a multi-domain end-to-end path, it can be of any type (primary, secondary, reverse-primary or reverse-secondary), as specified by the (path-role) choice:

```

| | | | +-- (path-role)?
| | | | | +--:(primary-path)
| | | | | +--:(secondary-path)
| | | | | | +-- secondary-path!
| | | | | | | +-- primary-path-name?    string
| | | | | +--:(primary-reverse-path)
| | | | | | +-- primary-reverse-path!
| | | | | | | +-- primary-path-name?    string
| | | | | +--:(secondary-reverse-path)
| | | | | | +-- secondary-reverse-path
| | | | | | | +-- primary-path-name?    string
| | | | | | | +-- primary-reverse-path-name? string
| | | | | .....

```

In all the other cases, the requested path can only be a primary path.

The secondary-path, the primary-reverse-path and the secondary-reverse-path are presence container used to indicate the role of the path: by default, the path is assumed to be a primary path.

They optionally can contain the name of the primary-path under which the requested path could be associated within the YANG tree structure defined in [I-D.ietf-teas-yang-te], which could be useful when the server also creates a TE tunnel instance within the operational datastore to report the computed paths, as described in Section 3.3.1: in this case, the tunnel-name and the path names are also used as the suggested name for that TE tunnel and path instances.

5.3.2. Tunnel attributes specified by reference

The (reference) case provides the information needed to associate multiple path requests that are intended to be used within the same tunnel.

In order to indicate the role of the path being requested within the intended tunnel (e.g., primary or secondary path), the (path-role) choice is defined:

```

+-- (path-role)
+--:(primary-path)
|   +-- primary-path!
|       .....
+--:(secondary-path)
|   +-- secondary-path
|       .....
+--:(primary-reverse-path)
|   +-- primary-reverse-path
|       .....
+--:(secondary-reverse-path)
|   +-- secondary-reverse-path
|       +-- preference?                uint8
|       +-- protection-type?           identityref
|       +-- restoration-type?          identityref
|       +-- restoration-scheme?        identityref
|       +-- primary-reverse-path-ref* []
|       +-- (primary-reverse-path-exist)?
|           +--:(path-ref)
|           |   +-- primary-path-ref    leafref
|           +--:(path-request-ref)
|           |   +-- path-request-ref    leafref

```

The primary-path is a presence container used to indicate that the requested path is intended to be used as a primary path. It can also contain some attributes which are configured only on primary paths (e.g., the k-requested-paths).

The secondary-path container indicates that the requested path is intended to be used as a secondary path and it contains at least references to one or more primary paths which can use it as a candidate secondary path:

```

|-- secondary-path
| .....
| |-- primary-path-ref* []
| | |-- (primary-path-exist)?
| | |--: (path-ref)
| | | |-- primary-path-ref      leafref
| | |--: (path-request-ref)
| | | |-- path-request-ref      leafref

```


A requested secondary path can reference any requested primary paths, and, in case they are intended to be used within an existing TE tunnel, it could also reference any existing primary-paths.

The secondary-path container can also contain some attributes which are configured only on secondary paths (e.g., the protection-type).

The primary-reverse-path container indicates that the requested path is intended to be used as a primary reverse path and it contains only the reference to the primary path which is intended to use it as a reverse path:

```

+-- primary-reverse-path
+-- (primary-path-exist)?
+--: (path-ref)
|   +-- primary-path-ref      leafref
+--: (path-request-ref)
|   +-- path-request-ref      leafref

```

A requested primary reverse path can reference either a requested primary path, or, in case it is intended to be used within an existing TE tunnel, an existing primary-path.

The secondary-reverse-path container indicates that the requested path is intended to be used as a secondary reverse path and it contains at least references to one or more primary paths, whose primary reverse path can use it as a candidate secondary reverse path:

```

+-- secondary-reverse-path
.....
+-- primary-reverse-path-ref* []
    +-- (primary-reverse-path-exist)?
        +--: (path-ref)
            |   +-- primary-path-ref      leafref
        +--: (path-request-ref)
            |   +-- path-request-ref      leafref

```

A requested secondary reverse path can reference any requested primary paths, and, in case they are intended to be used within an existing TE tunnel, it could reference also existing primary-paths.

The secondary-reverse-path container can also contain some attributes which are configured only on secondary reverse paths (e.g., the protection-type).

In case the requested path is a transit segment of a multi-domain end-to-end path, the primary-path may not be needed to be setup/computed. However, the request for path computation of a secondary-path or a primary-reverse or of a secondary-reverse-path requires that the primary-path exists or it is requested within the same RPC request. In the latter case, the path request for the primary-path should have an empty ERO to indicate to the server that path computation is not requested and no path properties will therefore be returned in the RPC response.

5.4. Multi-Layer Path Computation

The models supports requesting multi-layer path computation following the same approach based on dependency tunnels, as defined in [I-D.ietf-teas-yang-te].

The tunnel-attributes of a given client-layer path request can reference server-layer TE tunnels which can already exist in the YANG datastore or be specified in the tunnel-attributes list, within the same RPC request:

```

+-- dependency-tunnels
|   +-- dependency-tunnel* [name]
|   |   +-- name                -> /te:te/tunnels/tunnel/name
|   |   +-- encoding?           identityref
|   |   +-- switching-type?     identityref
|   |   +-- dependency-tunnel-attributes* [name]
|   |   |   +-- name                leafref
|   |   |   +-- encoding?         identityref
|   |   |   +-- switching-type?   identityref

```

In a similar way as in [I-D.ietf-teas-yang-te], the server-layer tunnel attributes should provide the information of what would be the dynamic link in the client layer topology supported by that tunnel, if instantiated:

```

+-- hierarchical-link
|   +-- local-te-node-id?        te-types:te-node-id
|   +-- local-te-link-tp-id?     te-types:te-tp-id
|   +-- remote-te-node-id?      te-types:te-node-id
|   +-- te-topology-identifier
|   |   +-- provider-id?        te-global-id
|   |   +-- client-id?          te-global-id
|   |   +-- topology-id?        te-topology-id

```

It is worth noting that since path computation RPC is stateless, the dynamic hierarchical links configured for the server-layer tunnel attributes cannot be used for path computation of any client-layer path unless explicitly referenced in the dependency-tunnel-attributes list within the same RPC request.

6. YANG data model for TE path computation

6.1. Tree diagram

Figure 11 below shows the tree diagram of the YANG data model defined in module `ietf-te-path-computation.yang`, defined in Section 6.2.

module: `ietf-te-path-computation`

```

augment /te:tunnels-path-compute/te:input/te:path-compute-info:
  +-- path-request* [request-id]
    +-- request-id                               uint32
    +-- (tunnel-attributes)?
      +--:(reference)
        +-- tunnel-reference
          +-- (tunnel-exist)?
            +--:(tunnel-ref)
              | +-- tunnel-ref                   te:tunnel-ref
            +--:(tunnel-attributes-ref)
              +-- tunnel-attributes-ref         leafref
          +-- path-name?                         string
          +-- (path-role)
            +--:(primary-path)
              +-- primary-path!
                +-- preference?                  uint8
                +-- k-requested-paths?          uint8
            +--:(secondary-path)
              +-- secondary-path
                +-- preference?                  uint8
                +-- protection-type?            identityref
                +-- restoration-type?           identityref
                +-- restoration-scheme?         identityref
                +-- primary-path-ref* []
                +-- (primary-path-exist)?
                  +--:(path-ref)
                    | +-- primary-path-ref       leafref
                  +--:(path-request-ref)
                    +-- path-request-ref         leafref
            +--:(primary-reverse-path)
              +-- primary-reverse-path
                +-- (primary-path-exist)?
                  +--:(path-ref)

```

```

|         | +-- primary-path-ref      leafref
|         +--:(path-request-ref)
|         +-- path-request-ref      leafref
+--:(secondary-reverse-path)
  +-- secondary-reverse-path
    +-- preference?                  uint8
    +-- protection-type?             identityref
    +-- restoration-type?            identityref
    +-- restoration-scheme?          identityref
    +-- primary-reverse-path-ref* []
    +-- (primary-reverse-path-exist)?
      +--:(path-ref)
      | +-- primary-path-ref      leafref
      +--:(path-request-ref)
      +-- path-request-ref      leafref
+--:(value)
  +-- tunnel-name?                  string
  +-- path-name?                    string
  +-- (path-role)?
    +--:(primary-path)
    +--:(secondary-path)
    | +-- secondary-path!
    |   +-- primary-path-name?      string
    +--:(primary-reverse-path)
    | +-- primary-reverse-path!
    |   +-- primary-path-name?      string
    +--:(secondary-reverse-path)
    | +-- secondary-reverse-path!
    |   +-- primary-path-name?      string
    |   +-- primary-reverse-path-name? string
  +-- k-requested-paths?            uint8
  +-- encoding?                     identityref
  +-- switching-type?               identityref
  +-- source?                       te-types:te-node-id
  +-- destination?                  te-types:te-node-id
  +-- src-tunnel-tp-id?             binary
  +-- dst-tunnel-tp-id?             binary
  +-- bidirectional?                boolean
  +-- te-topology-identifier
    +-- provider-id?                te-global-id
    +-- client-id?                  te-global-id
    +-- topology-id?                te-topology-id
+-- association-objects
  +-- association-object* [association-key]
    +-- association-key              string
    +-- type?                        identityref
    +-- id?                          uint16
    +-- source

```

```

    |
    |     +--- id?      te-gen-node-id
    |     +--- type?    enumeration
    | +--- association-object-extended* [association-key]
    |     +--- association-key    string
    |     +--- type?              identityref
    |     +--- id?                uint16
    |     +--- source
    |     |     +--- id?      te-gen-node-id
    |     |     +--- type?    enumeration
    |     +--- global-source?  uint32
    |     +--- extended-id?    yang:hex-string
+--- optimizations
    +--- (algorithm)?
        +---:(metric) {path-optimization-metric}?
            +--- optimization-metric* [metric-type]
                +--- metric-type                                identityref
                +--- weight?                                    uint8
                +--- explicit-route-exclude-objects
                    +--- route-object-exclude-object* [index]
                        +--- index                                uint32
                        +--- (type)?
                            +---:(numbered-node-hop)
                                +--- numbered-node-hop
                                    +--- node-id      te-node-id
                                    +--- hop-type?     te-hop-type
                            +---:(numbered-link-hop)
                                +--- numbered-link-hop
                                    +--- link-tp-id    te-tp-id
                                    +--- hop-type?     te-hop-type
                                    +--- direction?    te-link-direction
                            +---:(unnumbered-link-hop)
                                +--- unnumbered-link-hop
                                    +--- link-tp-id    te-tp-id
                                    +--- node-id        te-node-id
                                    +--- hop-type?     te-hop-type
                                    +--- direction?    te-link-direction
                            +---:(as-number)
                                +--- as-number-hop
                                    +--- as-number      inet:as-number
                                    +--- hop-type?     te-hop-type
                            +---:(label)
                                +--- label-hop
                                    +--- te-label
                                        +--- (technology)?
                                            +---:(generic)
                                                +--- generic?
                                                    rt-types:generalized-label
                                +--- direction?

```

```

|                                     te-label-direction
|                                     +---:(srlg)
|                                       +--- srlg
|                                         +--- srlg?    uint32
+--- explicit-route-include-objects
|   +--- route-object-include-object* [index]
|     +--- index                                uint32
|     +--- (type)?
|       +---:(numbered-node-hop)
|         +--- numbered-node-hop
|           +--- node-id      te-node-id
|           +--- hop-type?    te-hop-type
|       +---:(numbered-link-hop)
|         +--- numbered-link-hop
|           +--- link-tp-id    te-tp-id
|           +--- hop-type?    te-hop-type
|           +--- direction?    te-link-direction
|       +---:(unnumbered-link-hop)
|         +--- unnumbered-link-hop
|           +--- link-tp-id    te-tp-id
|           +--- node-id      te-node-id
|           +--- hop-type?    te-hop-type
|           +--- direction?    te-link-direction
|       +---:(as-number)
|         +--- as-number-hop
|           +--- as-number    inet:as-number
|           +--- hop-type?    te-hop-type
|       +---:(label)
|         +--- label-hop
|         +--- te-label
|           +--- (technology)?
|             +---:(generic)
|               +--- generic?
|                 rt-types:generalized-label
|           +--- direction?
|             te-label-direction
+--- tiebreakers
|   +--- tiebreaker* [tiebreaker-type]
|     +--- tiebreaker-type    identityref
+---:(objective-function)
|   {path-optimization-objective-function}?
|   +--- objective-function
|     +--- objective-function-type?    identityref
+--- named-path-constraint?          leafref
|   {te-types:named-path-constraints}?
+--- te-bandwidth
|   +--- (technology)?
|     +---:(generic)

```

```

|         +--- generic?      te-bandwidth
+--- link-protection?                               identityref
+--- setup-priority?                               uint8
+--- hold-priority?                                uint8
+--- signaling-type?                                identityref
+--- path-metric-bounds
|   +--- path-metric-bound* [metric-type]
|       +--- metric-type      identityref
|       +--- upper-bound?    uint64
+--- path-affinities-values
|   +--- path-affinities-value* [usage]
|       +--- usage            identityref
|       +--- value?          admin-groups
+--- path-affinity-names
|   +--- path-affinity-name* [usage]
|       +--- usage            identityref
|       +--- affinity-name* [name]
|           +--- name         string
+--- path-srlgs-lists
|   +--- path-srlgs-list* [usage]
|       +--- usage            identityref
|       +--- values*         srlg
+--- path-srlgs-names
|   +--- path-srlgs-name* [usage]
|       +--- usage            identityref
|       +--- names*          string
+--- disjointness?                                te-path-disjointness
+--- explicit-route-objects-always
|   +--- route-object-exclude-always* [index]
|       +--- index            uint32
|       +--- (type)?
|           +---:(numbered-node-hop)
|               +--- numbered-node-hop
|                   +--- node-id      te-node-id
|                   +--- hop-type?    te-hop-type
|           +---:(numbered-link-hop)
|               +--- numbered-link-hop
|                   +--- link-tp-id    te-tp-id
|                   +--- hop-type?     te-hop-type
|                   +--- direction?    te-link-direction
|           +---:(unnumbered-link-hop)
|               +--- unnumbered-link-hop
|                   +--- link-tp-id    te-tp-id
|                   +--- node-id      te-node-id
|                   +--- hop-type?     te-hop-type
|                   +--- direction?    te-link-direction
|           +---:(as-number)
|               +--- as-number-hop

```

```

    +--- as-number      inet:as-number
    +--- hop-type?      te-hop-type
    +---:(label)
      +--- label-hop
      +--- te-label
      +--- (technology)?
        +---:(generic)
          +--- generic?
            rt-types:generalized-label
      +--- direction?    te-label-direction
+--- route-object-include-exclude* [index]
+--- explicit-route-usage?      identityref
+--- index                      uint32
+--- (type)?
  +---:(numbered-node-hop)
    +--- numbered-node-hop
    +--- node-id      te-node-id
    +--- hop-type?    te-hop-type
  +---:(numbered-link-hop)
    +--- numbered-link-hop
    +--- link-tp-id    te-tp-id
    +--- hop-type?    te-hop-type
    +--- direction?    te-link-direction
  +---:(unnumbered-link-hop)
    +--- unnumbered-link-hop
    +--- link-tp-id    te-tp-id
    +--- node-id      te-node-id
    +--- hop-type?    te-hop-type
    +--- direction?    te-link-direction
  +---:(as-number)
    +--- as-number-hop
    +--- as-number      inet:as-number
    +--- hop-type?      te-hop-type
  +---:(label)
    +--- label-hop
    +--- te-label
    +--- (technology)?
      +---:(generic)
        +--- generic?
          rt-types:generalized-label
    +--- direction?    te-label-direction
  +---:(srlg)
    +--- srlg
    +--- srlg?    uint32
+--- path-in-segment!
+--- label-restrictions
  +--- label-restriction* [index]
  +--- restriction?      enumeration

```



```

+-- index                uint32
+-- label-start
|   +-- te-label
|   |   +-- (technology)?
|   |   |   +--:(generic)
|   |   |   +-- generic?    rt-types:generalized-label
|   |   +-- direction?     te-label-direction
+-- label-end
|   +-- te-label
|   |   +-- (technology)?
|   |   |   +--:(generic)
|   |   |   +-- generic?    rt-types:generalized-label
|   |   +-- direction?     te-label-direction
+-- label-step
|   +-- (technology)?
|   |   +--:(generic)
|   |   +-- generic?    int32
+-- range-bitmap?    yang:hex-string
+-- path-out-segment!
|   +-- label-restrictions
|   |   +-- label-restriction* [index]
|   |   +-- restriction?    enumeration
|   |   +-- index          uint32
|   |   +-- label-start
|   |   |   +-- te-label
|   |   |   |   +-- (technology)?
|   |   |   |   |   +--:(generic)
|   |   |   |   |   +-- generic?    rt-types:generalized-label
|   |   |   +-- direction?     te-label-direction
|   |   +-- label-end
|   |   |   +-- te-label
|   |   |   |   +-- (technology)?
|   |   |   |   |   +--:(generic)
|   |   |   |   |   +-- generic?    rt-types:generalized-label
|   |   |   +-- direction?     te-label-direction
|   |   +-- label-step
|   |   |   +-- (technology)?
|   |   |   |   +--:(generic)
|   |   |   |   +-- generic?    int32
|   |   +-- range-bitmap?    yang:hex-string
+-- requested-metrics* [metric-type]
|   +-- metric-type    identityref
+-- return-srlgs?                boolean
+-- return-affinities?           boolean
+-- requested-state!
|   +-- timer?                uint16
|   +-- transaction-id?      string
+-- tunnel-attributes* [tunnel-name]

```

```

+-- tunnel-name                string
+-- encoding?                  identityref
+-- switching-type?            identityref
+-- source?                    te-types:te-node-id
+-- destination?               te-types:te-node-id
+-- src-tunnel-tp-id?          binary
+-- dst-tunnel-tp-id?          binary
+-- bidirectional?             boolean
+-- association-objects
|   +-- association-object* [association-key]
|   |   +-- association-key    string
|   |   +-- type?              identityref
|   |   +-- id?                uint16
|   |   +-- source
|   |   |   +-- id?            te-gen-node-id
|   |   |   +-- type?          enumeration
|   |   +-- association-object-extended* [association-key]
|   |   |   +-- association-key    string
|   |   |   +-- type?              identityref
|   |   |   +-- id?                uint16
|   |   |   +-- source
|   |   |   |   +-- id?            te-gen-node-id
|   |   |   |   +-- type?          enumeration
|   |   |   +-- global-source?    uint32
|   |   |   +-- extended-id?      yang:hex-string
|   |   +-- protection-type?      identityref
|   |   +-- restoration-type?      identityref
|   |   +-- restoration-scheme?    identityref
|   +-- te-topology-identifier
|   |   +-- provider-id?    te-global-id
|   |   +-- client-id?      te-global-id
|   |   +-- topology-id?    te-topology-id
|   +-- te-bandwidth
|   |   +-- (technology)?
|   |   |   +--:(generic)
|   |   |   |   +-- generic?    te-bandwidth
|   +-- link-protection?        identityref
|   +-- setup-priority?          uint8
|   +-- hold-priority?           uint8
|   +-- signaling-type?          identityref
|   +-- hierarchy
|   |   +-- dependency-tunnels
|   |   |   +-- dependency-tunnel* [name]
|   |   |   |   +-- name          -> /te:te/tunnels/tunnel/name
|   |   |   |   +-- encoding?      identityref
|   |   |   |   +-- switching-type? identityref
|   |   |   +-- dependency-tunnel-attributes* [name]
|   |   |   |   +-- name          leafref

```

```

|         +--- encoding?          identityref
|         +--- switching-type?    identityref
+--- hierarchical-link
|         +--- local-te-node-id?    te-types:te-node-id
|         +--- local-te-link-tp-id? te-types:te-tp-id
|         +--- remote-te-node-id?   te-types:te-node-id
|         +--- te-topology-identifier
|         +--- provider-id?         te-global-id
|         +--- client-id?           te-global-id
|         +--- topology-id?         te-topology-id
+--- synchronization* []
|   +--- svec
|   |   +--- relaxable?             boolean
|   |   +--- disjointness?          te-path-disjointness
|   |   +--- request-id-number*     uint32
|   +--- svec-constraints
|   |   +--- path-metric-bound* [metric-type]
|   |   |   +--- metric-type        identityref
|   |   |   +--- upper-bound?       uint64
|   +--- path-srlgs-lists
|   |   +--- path-srlgs-list* [usage]
|   |   |   +--- usage              identityref
|   |   |   +--- values*            srlg
|   +--- path-srlgs-names
|   |   +--- path-srlgs-name* [usage]
|   |   |   +--- usage              identityref
|   |   |   +--- names*             string
|   +--- exclude-objects
|   |   +--- excludes* []
|   |   |   +--- (type)?
|   |   |   +---:(numbered-node-hop)
|   |   |   |   +--- numbered-node-hop
|   |   |   |   |   +--- node-id      te-node-id
|   |   |   |   |   +--- hop-type?    te-hop-type
|   |   |   +---:(numbered-link-hop)
|   |   |   |   +--- numbered-link-hop
|   |   |   |   |   +--- link-tp-id    te-tp-id
|   |   |   |   |   +--- hop-type?     te-hop-type
|   |   |   |   |   +--- direction?    te-link-direction
|   |   |   +---:(unnumbered-link-hop)
|   |   |   |   +--- unnumbered-link-hop
|   |   |   |   |   +--- link-tp-id    te-tp-id
|   |   |   |   |   +--- node-id      te-node-id
|   |   |   |   |   +--- hop-type?     te-hop-type
|   |   |   |   |   +--- direction?    te-link-direction
|   |   |   +---:(as-number)
|   |   |   |   +--- as-number-hop
|   |   |   |   |   +--- as-number     inet:as-number

```

```

|         |         +--- hop-type?      te-hop-type
+---:(label)
|         |         +--- label-hop
|         |         +--- te-label
|         |         |         +--- (technology)?
|         |         |         |         +---:(generic)
|         |         |         |         |         +--- generic?
|         |         |         |         |         |         rt-types:generalized-label
|         |         |         |         +--- direction?      te-label-direction
+--- optimizations
|   +--- (algorithm)?
|   |   +---:(metric) {te-types:path-optimization-metric}?
|   |   |   +--- optimization-metric* [metric-type]
|   |   |   |   +--- metric-type      identityref
|   |   |   |   +--- weight?          uint8
|   |   +---:(objective-function)
|   |   |   {te-types:path-optimization-objective-function}?
|   |   |   +--- objective-function
|   |   |   |   +--- objective-function-type?      identityref
augment /te:tunnels-path-compute/te:output/te:path-compute-result:
+---ro response* [response-id]
|   +---ro response-id          uint32
+---ro computed-paths-properties
|   +---ro computed-path-properties* [k-index]
|   |   +---ro k-index          uint8
|   |   +---ro path-properties
|   |   |   +---ro path-metric* [metric-type]
|   |   |   |   +---ro metric-type      identityref
|   |   |   |   +---ro accumulative-value?      uint64
|   |   +---ro path-affinities-values
|   |   |   +---ro path-affinities-value* [usage]
|   |   |   |   +---ro usage      identityref
|   |   |   |   +---ro value?      admin-groups
|   |   +---ro path-affinity-names
|   |   |   +---ro path-affinity-name* [usage]
|   |   |   |   +---ro usage      identityref
|   |   |   |   +---ro affinity-name* [name]
|   |   |   |   |   +---ro name      string
|   |   +---ro path-srlgs-lists
|   |   |   +---ro path-srlgs-list* [usage]
|   |   |   |   +---ro usage      identityref
|   |   |   |   +---ro values*      srlg
|   |   +---ro path-srlgs-names
|   |   |   +---ro path-srlgs-name* [usage]
|   |   |   |   +---ro usage      identityref
|   |   |   |   +---ro names*      string
|   +---ro path-route-objects
|   |   +---ro path-route-object* [index]

```

```

+--ro index                               uint32
+--ro (type)?
  +--:(numbered-node-hop)
    +--ro numbered-node-hop
      +--ro node-id      te-node-id
      +--ro hop-type?    te-hop-type
  +--:(numbered-link-hop)
    +--ro numbered-link-hop
      +--ro link-tp-id    te-tp-id
      +--ro hop-type?    te-hop-type
      +--ro direction?   te-link-direction
  +--:(unnumbered-link-hop)
    +--ro unnumbered-link-hop
      +--ro link-tp-id    te-tp-id
      +--ro node-id      te-node-id
      +--ro hop-type?    te-hop-type
      +--ro direction?   te-link-direction
  +--:(as-number)
    +--ro as-number-hop
      +--ro as-number    inet:as-number
      +--ro hop-type?    te-hop-type
  +--:(label)
    +--ro label-hop
      +--ro te-label
        +--ro (technology)?
          +--:(generic)
            +--ro generic?
              rt-types:generalized-label
        +--ro direction?
          te-label-direction
+--ro te-bandwidth
  +--ro (technology)?
  +--:(generic)
    +--ro generic?    te-bandwidth
+--ro disjointness-type?
  te-types:te-path-disjointness
+--ro computed-path-error-infos
  +--ro computed-path-error-info* []
    +--ro error-description?    string
    +--ro error-timestamp?      yang:date-and-time
    +--ro error-reason?         identityref
+--ro tunnel-ref?              te:tunnel-ref
+--ro (path-role)?
  +--:(primary)
    +--ro primary-path-ref?      leafref
  +--:(primary-reverse)
    +--ro primary-reverse-path-ref?  leafref
  +--:(secondary)

```

```

      | +--ro secondary-path-ref?          leafref
      +--:(secondary-reverse)
        +--ro secondary-reverse-path-ref?  leafref
augment /te:tunnels-actions/te:input/te:tunnel-info/te:filter-type:
  +--:(path-compute-transactions)
    +-- path-compute-transaction-id*      string
augment /te:tunnels-actions/te:output:
  +--ro path-computed-delete-result
    +--ro path-compute-transaction-id*    string

```

Figure 11: TE path computation tree diagram

6.2. YANG module

```

<CODE BEGINS> file "ietf-te-path-computation@2022-01-24.yang"
module iETF-te-path-computation {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";
  prefix te-pc;

  import iETF-te {
    prefix te;
    reference
      "RFCYYYY: 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
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }

  organization
    "Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web:  <http://tools.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>

    Editor:   Italo Busi
              <mailto:italo.busi@huawei.com>

    Editor:   Sergio Belotti
              <mailto:sergio.belotti@nokia.com>

```

Editor: Victor Lopez
<mailto:victor.lopez@nokia.com>

Editor: Oscar Gonzalez de Dios
<mailto:oscar.gonzalezdedios@telefonica.com>

Editor: Anurag Sharma
<mailto:ansha@google.com>

Editor: Yan Shi
<mailto:shiyang49@chinaunicom.cn>

Editor: Ricard Vilalta
<mailto:ricard.vilalta@cttc.es>

Editor: Karthik Sethuraman
<mailto:karthik.sethuraman@necam.com>

Editor: Michael Scharf
<mailto:michael.scharf@gmail.com>

Editor: Daniele Ceccarelli
<mailto:daniele.ceccarelli@ericsson.com>

";
description
"This module defines a YANG data model for requesting Traffic Engineering (TE) path computation. The YANG model defined in this document is based on RPCs augmenting the RPCs defined in the generic TE module (ietf-te).
The model fully conforms to the
Network Management Datastore Architecture (NMDA).

Copyright (c) 2022 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."

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

```
// this note
// replace the revision date with the module publication date
// the format is (year-month-day)

revision 2022-01-24 {
  description
    "Initial revision";
  reference
    "RFC XXXX: A YANG Data Model for requesting path computation";
}

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

/*
 * Identities
 */

identity svec-metric-type {
  description
    "Base identity for SVEC metric type.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-te {
  base svec-metric-type;
  description
    "Cumulative TE cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-igp {
  base svec-metric-type;
  description
    "Cumulative IGP cost.";
  reference
    "RFC5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP).";
}

identity svec-metric-cumul-hop {
  base svec-metric-type;
  description
    "Cumulative Hop path metric.";
```



```
    reference
      "RFC8776: Common YANG Data Types for Traffic Engineering.";
  }

  identity svec-metric-aggregate-bandwidth-consumption {
    base svec-metric-type;
    description
      "Aggregate bandwidth consumption.";
    reference
      "RFC5541: Encoding of Objective Functions in the Path
      Computation Element Communication Protocol (PCEP).";
  }

  identity svec-metric-load-of-the-most-loaded-link {
    base svec-metric-type;
    description
      "Load of the most loaded link.";
    reference
      "RFC5541: Encoding of Objective Functions in the Path
      Computation Element Communication Protocol (PCEP).";
  }

  identity tunnel-action-path-compute-delete {
    base te:tunnel-actions-type;
    description
      "Action type to delete the transient states
      of computed paths, as described in section 3.3.1 of
      RFC XXXX.";
    reference
      "RFC XXXX: A YANG Data Model for requesting path computation";
  }

  /*
   * Groupings
   */

  grouping protection-restoration-properties {
    description
      "This grouping defines the restoration and protection types
      for a path in the path computation request.";
    leaf protection-type {
      type identityref {
        base te-types:lsp-protection-type;
      }
      default "te-types:lsp-protection-unprotected";
      description
        "LSP protection type.";
    }
  }
```

```
    leaf restoration-type {
      type identityref {
        base te-types:lsp-restoration-type;
      }
      default "te-types:lsp-restoration-restore-any";
      description
        "LSP restoration type.";
    }
    leaf restoration-scheme {
      type identityref {
        base te-types:restoration-scheme-type;
      }
      default "te-types:restoration-scheme-preconfigured";
      description
        "LSP restoration scheme.";
    }
  } // grouping protection-restoration-properties

  grouping requested-info {
    description
      "This grouping defines the information (e.g., metrics)
       which is requested, in the path computation request, to be
       returned in the path computation response.";
    list requested-metrics {
      key "metric-type";
      description
        "The list of the requested metrics.
         The metrics listed here must be returned in the response.
         Returning other metrics in the response is optional.";
      leaf metric-type {
        type identityref {
          base te-types:path-metric-type;
        }
        description
          "The metric that must be returned in the response";
      }
    }
  }
  leaf return-srlgs {
    type boolean;
    default "false";
    description
      "If true, path srlgs must be returned in the response.
       If false, returning path srlgs in the response optional.";
  }
  leaf return-affinities {
    type boolean;
    default "false";
    description
```

```
        "If true, path affinities must be returned in the response.
        If false, returning path affinities in the response is
        optional.";
    }
} // grouping requested-info

grouping requested-state {
  description
    "Configuration for the transient state used
    to report the computed path";
  container requested-state {
    presence
      "Request temporary reporting of the computed path state";
    description
      "Configures attributes for the temporary reporting of the
      computed path state (e.g., expiration timer).";
    leaf timer {
      type uint16;
      units "minutes";
      default "10";
      description
        "The timeout after which the transient state reporting
        the computed path should be removed.";
    }
    leaf transaction-id {
      type string;
      description
        "The transaction-id associated with this path computation
        to be used for fast deletion of the transient states
        associated with multiple path computations.

        This transaction-id can be used to explicitly delete all
        the transient states of all the computed paths associated
        with the same transaction-id.

        When one path associated with a transaction-id is setup,
        the transient states of all the other computed paths
        with the same transaction-id are automatically removed.

        If not specified, the transient state is removed only
        when the timer expires (when the timer is specified)
        or not created at all (stateless path computation,
        when the timer is not specified).";
    }
  }
} // grouping requested-state

grouping reported-state {
```

```
description
  "This grouping defines the information, returned in the path
  computation response, reporting the transient state related
  to the computed path";
leaf tunnel-ref {
  type te:tunnel-ref;
  description
    "
      Reference to the tunnel that reports the transient state
      of the computed path.

      If no transient state is created, this attribute is
      omitted.
    ";
}
choice path-role {
  description
    "The transient state of the computed path can be reported
    as a primary, primary-reverse, secondary or
    a secondary-reverse path of a te-tunnel";
  case primary {
    leaf primary-path-ref {
      type leafref {
        path "/te:te/te:tunnels/"
          + "te:tunnel[te:name=current()/../tunnel-ref]/"
          + "te:primary-paths/te:primary-path/"
          + "te:name";
      }
      must '../tunnel-ref' {
        description
          "The primary-path name can only be reported
          if also the tunnel name is reported.";
      }
    }
    description
      "
        Reference to the primary-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
      ";
  }
} // case primary
case primary-reverse {
  leaf primary-reverse-path-ref {
    type leafref {
      path "/te:te/te:tunnels/"
        + "te:tunnel[te:name=current()/../tunnel-ref]/"
```

```
        + "te:primary-paths/te:primary-path/"
        + "te:name";
    }
    must '../tunnel-ref' {
        description
            "The primary-reverse-path name can only be reported
            if also the tunnel name is reported.";
    }
    description
        "
        Reference to the primary-reverse-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is omitted.
        ";
    }
} // case primary-reverse
case secondary {
    leaf secondary-path-ref {
        type leafref {
            path "/te:te/te:tunnels/"
                + "te:tunnel[te:name=current()../tunnel-ref]/"
                + "te:secondary-paths/te:secondary-path/"
                + "te:name";
        }
        must '../tunnel-ref' {
            description
                "The secondary-path name can only be reported
                if also the tunnel name is reported.";
        }
        description
            "
            Reference to the secondary-path that reports
            the transient state of the computed path.

            If no transient state is created,
            this attribute is omitted.
            ";
        }
    }
} // case secondary
case secondary-reverse {
    leaf secondary-reverse-path-ref {
        type leafref {
            path "/te:te/te:tunnels/"
                + "te:tunnel[te:name=current()../tunnel-ref]/"
                + "te:secondary-reverse-paths/"
                + "te:secondary-reverse-path/te:name";
        }
    }
}
```

```
    }
    must '../tunnel-ref' {
      description
        "The secondary-reverse-path name can only be reported
        if also the tunnel name is reported.";
    }
    description
      "
      Reference to the secondary-reverse-path that reports
      the transient state of the computed path.

      If no transient state is created,
      this attribute is omitted.
      ";
  }
} // case secondary
} // choice path
} // grouping reported-state

grouping synchronization-constraints {
  description
    "Global constraints applicable to synchronized path
    computation requests.";
  container svec-constraints {
    description
      "global svec constraints";
    list path-metric-bound {
      key "metric-type";
      description
        "list of bound metrics";
      leaf metric-type {
        type identityref {
          base svec-metric-type;
        }
        description
          "SVEC metric type.";
        reference
          "RFC5541: Encoding of Objective Functions in the Path
          Computation Element Communication Protocol (PCEP).";
      }
      leaf upper-bound {
        type uint64;
        description
          "Upper bound on SVEC metric";
      }
    }
  }
}
uses te-types:generic-path-srlgs;
```

```
    container exclude-objects {
      description
        "Resources to be excluded";
      list excludes {
        description
          "List of Explicit Route Objects to always exclude
           from synchronized path computation";
        uses te-types:explicit-route-hop;
      }
    }
  } // grouping synchronization-constraints

grouping synchronization-optimization {
  description
    "Optimizations applicable to synchronized path
     computation requests.";
  container optimizations {
    description
      "The objective function container that includes attributes
       to impose when computing a synchronized set of paths";
    choice algorithm {
      description
        "Optimizations algorithm.";
      case metric {
        if-feature "te-types:path-optimization-metric";
        list optimization-metric {
          key "metric-type";
          description
            "svec path metric type";
          leaf metric-type {
            type identityref {
              base svec-metric-type;
            }
            description
              "TE path metric type usable for computing a set of
               synchronized requests";
          }
          leaf weight {
            type uint8;
            description
              "Metric normalization weight";
          }
        }
      }
    }
    case objective-function {
      if-feature
        "te-types:path-optimization-objective-function";
      container objective-function {
```

```

        description
            "The objective function container that includes
            attributes to impose when computing a TE path";
        leaf objective-function-type {
            type identityref {
                base te-types:objective-function-type;
            }
            default "te-types:of-minimize-cost-path";
            description
                "Objective function entry";
        }
    }
}
}
} // grouping synchronization-optimization

grouping synchronization-info {
    description
        "Information for synchronized path computation requests.";
    list synchronization {
        description
            "List of Synchronization VECTors.";
        container svec {
            description
                "Synchronization VECTor";
            leaf relaxable {
                type boolean;
                default "true";
                description
                    "If this leaf is true, path computation process is
                    free to ignore svec content.
                    Otherwise, it must take into account this svec.";
            }
            uses te-types:generic-path-disjointness;
            leaf-list request-id-number {
                type uint32;
                description
                    "This list reports the set of path computation
                    requests that must be synchronized.";
            }
        }
        uses synchronization-constraints;
        uses synchronization-optimization;
    }
} // grouping synchronization-info

/*

```



```
* Augment TE RPCs
*/

augment "/te:tunnels-path-compute/te:input/te:path-compute-info" {
  description
    "Path Computation RPC input";
  list path-request {
    key "request-id";
    description
      "The list of the requested paths to be computed";
    leaf request-id {
      type uint32;
      mandatory true;
      description
        "Each path computation request is uniquely identified
        within the RPC request by the request-id-number.";
    }
  }
  choice tunnel-attributes {
    default "value";
    description
      "Whether the tunnel attributes are specified by value
      within this path computation request or by reference.
      The reference could be either to an existing te-tunnel
      or to an entry in the tunnel-attributes list";
    case reference {
      container tunnel-reference {
        description
          "Attributes for a requested path that belongs to
          either an exiting te-tunnel or to an entry in the
          tunnel-attributes list.";
        choice tunnel-exist {
          description
            "Whether the tunnel reference is to an existing
            te-tunnel or to an entry in the tunnel-attributes
            list";
          case tunnel-ref {
            leaf tunnel-ref {
              type te:tunnel-ref;
              mandatory true;
              description
                "The referenced te-tunnel instance";
            }
          } // case tunnel-ref
          case tunnel-attributes-ref {
            leaf tunnel-attributes-ref {
              type leafref {
                path "/te:tunnels-path-compute/"
                  + "te:path-compute-info/"

```

```
        + "te-pc:tunnel-attributes/te-pc:tunnel-name";
    }
    mandatory true;
    description
        "The referenced te-tunnel instance";
    }
} // case tunnel-attributes-ref
} // choice tunnel-exist
leaf path-name {
    type string;
    description
        "TE path name.";
}
choice path-role {
    mandatory true;
    description
        "Whether this path is a primary, or a reverse
        primary, or a secondary, or a reverse secondary
        path.";
    case primary-path {
        container primary-path {
            presence "Indicates that the requested path
                is a primary path";
            description
                "TE primary path";
            uses te:path-preference;
            uses te:k-requested-paths;
        } // container primary-path
    } // case primary-path
    case secondary-path {
        container secondary-path {
            description
                "TE secondary path";
            uses te:path-preference;
            uses protection-restoration-properties;
            list primary-path-ref {
                min-elements 1;
                description
                    "The list of primary paths that reference
                    this path as a candidate secondary path";
                choice primary-path-exist {
                    description
                        "Whether the path reference is to an existing
                        te-tunnel path or to another path request";
                    case path-ref {
                        leaf primary-path-ref {
                            type leafref {
                                path "/te:te/te:tunnels/te:tunnel"
```

```

        + "[te:name=current()/../../../../]"
        + "tunnel-ref]/te:primary-paths/"
        + "te:primary-path/te:name";
    }
    must ' ../../../../tunnel-ref' {
        description
            "The primary-path can be referenced
            if also the tunnel is referenced.";
    }
    mandatory true;
    description
        "The referenced primary path";
    }
} // case path-ref
case path-request-ref {
    leaf path-request-ref {
        type leafref {
            path "/te:tunnels-path-compute/"
                + "te:path-compute-info/"
                + "te-pc:path-request/"
                + "te-pc:request-id";
        }
        mandatory true;
        description
            "The referenced primary path request";
    }
} // case path-request-ref
} // choice primary-path-exist
} // list primary-path-ref
} // container secondary-path
} // case secondary-path
case primary-reverse-path {
    container primary-reverse-path {
        description
            "TE primary reverse path";
        choice primary-path-exist {
            description
                "Whether the path reference to the primary
                paths for which this path is the reverse-path
                is to an existing te-tunnel path or to
                another path request.";
            case path-ref {
                leaf primary-path-ref {
                    type leafref {
                        path "/te:te/te:tunnels/te:tunnel[te:name"
                            + "=current()/../../../../tunnel-ref]/"
                            + "te:primary-paths/te:primary-path/"
                            + "te:name";
                    }
                }
            }
        }
    }
}

```

```
    }
    must '../../tunnel-ref' {
      description
        "The primary-path can be referenced
        if also the tunnel is referenced.";
    }
    mandatory true;
    description
      "The referenced primary path";
  }
} // case path-ref
case path-request-ref {
  leaf path-request-ref {
    type leafref {
      path "/te:tunnels-path-compute/"
        + "te:path-compute-info/"
        + "te-pc:path-request/"
        + "te-pc:request-id";
    }
    mandatory true;
    description
      "The referenced primary path request";
  }
} // case path-request-ref
} // choice primary-path-exist
} // container primary-reverse-path
} // case primary-reverse-path
case secondary-reverse-path {
  container secondary-reverse-path {
    description
      "TE secondary reverse path";
    uses te:path-preference;
    uses protection-restoration-properties;
    list primary-reverse-path-ref {
      min-elements 1;
      description
        "The list of primary reverse paths that
        reference this path as a candidate
        secondary reverse path";
    }
    choice primary-reverse-path-exist {
      description
        "Whether the path reference is to an existing
        te-tunnel path or to another path request";
      case path-ref {
        leaf primary-path-ref {
          type leafref {
            path "/te:te/te:tunnels/te:tunnel"
              + "[te:name=current()../../../../]"
          }
        }
      }
    }
  }
}
```

```

        + "tunnel-ref]/te:primary-paths/"
        + "te:primary-path/te:name";
    }
    must '../.../tunnel-ref' {
        description
            "The primary-path can be referenced
            if also the tunnel is referenced.";
    }
    mandatory true;
    description
        "The referenced primary path";
    }
} // case path-ref
case path-request-ref {
    leaf path-request-ref {
        type leafref {
            path "/te:tunnels-path-compute/"
                + "te:path-compute-info/"
                + "te-pc:path-request/"
                + "te-pc:request-id";
        }
        mandatory true;
        description
            "The referenced primary reverse path
            request";
    }
} // case path-request-ref
} // choice primary-reverse-path-exist
} // list primary-reverse-path-ref
} // container secondary-reverse-path
} // case secondary-reverse-path
} // choice tunnel-path-role
}
} // case reference
case value {
    leaf tunnel-name {
        type string;
        description
            "TE tunnel name.";
    }
    leaf path-name {
        type string;
        description
            "TE path name.";
    }
}
choice path-role {
    when 'not (./source) and not (./destination) and
        not (./src-tunnel-tp-id) and

```

```
    not (./dst-tunnel-tp-id)' {
description
    "When the tunnel attributes are specified by value
    within this path computation, it is assumed that the
    requested path will be the only path of a tunnel.

    If the requested path is a transit segment path, it
    could be of any type. Otherwise it could only be a
    primary path.";
}
default primary-path;
description
    "Indicates whether the requested path is a primary
    path, a secondary path, a reverse primary path or a
    reverse secondary path.";
case primary-path {
description
    "The requested path is a primary path.";
}
container secondary-path {
presence
    "Indicates that the requested path is a secondary
    path.";
description
    "The name of the primary path which the requested
    primary reverse path belongs to.";
leaf primary-path-name {
type string;
description
    "TE primary path name.";
}
} // container secondary-path
container primary-reverse-path {
presence
    "Indicates that the requested path is a primary
    reverse path.";
description
    "The name of the primary path which the requested
    primary reverse path belongs to.";
leaf primary-path-name {
type string;
description
    "TE primary path name.";
}
} // container primary-reverse-path
container secondary-reverse-path {
presence
    "Indicates that the requested path is a secondary
```

```
        reverse path.";
    description
        "The names of the primary path and of the primary
        reverse path which the requested secondary reverse
        path belongs to.";
    leaf primary-path-name {
        type string;
        description
            "TE primary path name.";
    }
    leaf primary-reverse-path-name {
        type string;
        description
            "TE primary reverse path name.";
    }
    } // container primary-reverse-path
} // choice path-role
uses te:k-requested-paths;
uses te:encoding-and-switching-type;
uses te:tunnel-common-attributes;
uses te-types:te-topology-identifier;
} // case value
} // choice tunnel-attributes
uses te:path-compute-info;
uses requested-info;
uses requested-state;
}
list tunnel-attributes {
    key "tunnel-name";
    description
        "Tunnel attributes common to multiple request paths";
    leaf tunnel-name {
        type string;
        description
            "TE tunnel name.";
    }
    uses te:encoding-and-switching-type;
    uses te:tunnel-common-attributes;
    uses te:tunnel-associations-properties;
    uses protection-restoration-properties;
    uses te-types:tunnel-constraints;
    uses te:tunnel-hierarchy-properties {
        augment "hierarchy/dependency-tunnels" {
            description
                "Augment with the list of dependency tunnel requests.";
            list dependency-tunnel-attributes {
                key "name";
                description
```

```
        "A tunnel request entry that this tunnel request can
        potentially depend on.";
    leaf name {
        type leafref {
            path "/te:tunnels-path-compute/"
                + "te:path-compute-info/te-pc:tunnel-attributes/"
                + "te-pc:tunnel-name";
        }
        description
            "Dependency tunnel request name.";
    }
    uses te:encoding-and-switching-type;
}

}
}
}
uses synchronization-info;
} // path-compute rpc input

augment "/te:tunnels-path-compute/te:output/"
    + "te:path-compute-result" {
    description
        "Path Computation RPC output";
    list response {
        key "response-id";
        config false;
        description
            "response";
        leaf response-id {
            type uint32;
            description
                "The response-id has the same value of the
                corresponding request-id.";
        }
        uses te:path-computation-response;
        uses reported-state;
    }
} // path-compute rpc output

augment "/te:tunnels-actions/te:input/te:tunnel-info/"
    + "te:filter-type" {
    description
        "Augment Tunnels Action RPC input filter types";
    case path-compute-transactions {
        when "derived-from-or-self(..te:action-info/te:action, "
            + "'tunnel-action-path-compute-delete')";
        description
            "Path Delete Action RPC";
    }
}
```



```
    leaf-list path-compute-transaction-id {
      type string;
      description
        "The list of the transaction-id values of the
         transient states to be deleted";
    }
  }
} // path-delete rpc input

augment "/te:tunnels-actions/te:output" {
  description
    "Augment Tunnels Action RPC output with path delete result";
  container path-computed-delete-result {
    description
      "Path Delete RPC output";
    leaf-list path-compute-transaction-id {
      type string;
      description
        "The list of the transaction-id values of the
         transient states that have been successfully deleted";
    }
  }
} // path-delete rpc output
}
<CODE ENDS>
```

Figure 12: TE path computation YANG module

7. Security Considerations

This document describes use cases of requesting Path Computation using YANG data models, which could be used at the ABNO Control Interface [RFC7491] and/or between controllers in ACTN [RFC8453]. As such, it does not introduce any new security considerations compared to the ones related to YANG specification, ABNO specification and ACTN Framework defined in [RFC7950], [RFC7491] and [RFC8453].

The YANG module defined in this document is designed to be accessed via the NETCONF protocol [RFC6241] or RESTCONF protocol [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.

The YANG module defined in this document augments the "tunnels-path-compute" and the "tunnel-actions" RPCs defined in [I-D.ietf-teas-yang-te]. The security considerations provided in [I-D.ietf-teas-yang-te] are also applicable to the YANG module defined in this document.

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. These are the operations and their sensitivity/vulnerability:

"te-pc:response/computed-paths-properties": provides the same information provided by the "te:computed-paths-properties" defined in [I-D.ietf-teas-yang-te]. The security considerations provided in [I-D.ietf-teas-yang-te] for the TE tunnel state apply also to this subtree.

"te-pc:response/te-pc:tunnel-ref", "te-pc:response/te-pc:primary-path-ref", "te-pc:response/te-pc:primary-reverse-path-ref", "te-pc:response/te-pc:secondary-path-ref" and "te-pc:response/te-pc:secondary-reverse-path-ref" provides a reference where the same information provided in "te-pc:response/computed-paths-properties" is temporarily stored with the operational datastore (see Section 3.3.1). Therefore access to this information does not provide any additional security issue that the information provided with "te-pc:response/computed-paths-properties".

"/te:tunnels-actions": the YANG model defined in this document augments this action with a new action type that allows deleting the transient states of computed paths (see Section 3.3.1). A malicious use of this action would have no impact on the paths carrying live traffic but it would preclude the client from using the "transient states" to request the set-up of exactly that path, if still available.

The security considerations spelled out in the YANG specification [RFC7950] apply for this document as well.

8. IANA Considerations

This document registers the following URIs in the "ns" subregistry within the "IETF XML registry" [RFC3688].

URI: urn:ietf:params:xml:ns:yang:ietf-te-path-computation
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-path-computation
namespace: urn:ietf:params:xml:ns:yang:ietf-te-path-computation
prefix:    te-pc
reference:  this document
```

9. References

9.1. Normative References

- [I-D.ietf-teas-yang-te] Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I., and O. G. D. Dios, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-29, 7 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-teas-yang-te-29.txt>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <<https://www.rfc-editor.org/info/rfc5440>>.
- [RFC5441] Vasseur, JP., Ed., Zhang, R., Bitar, N., and JL. Le Roux, "A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths", RFC 5441, DOI 10.17487/RFC5441, April 2009, <<https://www.rfc-editor.org/info/rfc5441>>.
- [RFC5541] Le Roux, JL., Vasseur, JP., and Y. Lee, "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, DOI 10.17487/RFC5541, June 2009, <<https://www.rfc-editor.org/info/rfc5541>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [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>>.

9.2. Informative References

- [I-D.ietf-ccamp-otn-topo-yang]
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. G. D. Dios, "A YANG Data Model for Optical Transport Network Topology", Work in Progress, Internet-Draft, draft-ietf-ccamp-otn-topo-yang-14, 7 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-ccamp-otn-topo-yang-14.txt>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC6805] King, D., Ed. and A. Farrel, Ed., "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", RFC 6805, DOI 10.17487/RFC6805, November 2012, <<https://www.rfc-editor.org/info/rfc6805>>.
- [RFC7446] Lee, Y., Ed., Bernstein, G., Ed., Li, D., and W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, DOI 10.17487/RFC7446, February 2015, <<https://www.rfc-editor.org/info/rfc7446>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, DOI 10.17487/RFC7491, March 2015, <<https://www.rfc-editor.org/info/rfc7491>>.
- [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>>.
- [RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <<https://www.rfc-editor.org/info/rfc8453>>.
- [RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <<https://www.rfc-editor.org/info/rfc8454>>.

Appendix A. Examples

This section contains examples of use of the model with RESTCONF [RFC8040] and JSON encoding.

These examples show how path computation can be requested for the tunnels configuration provided in Appendix A of [I-D.ietf-teas-yang-te].

A.1. Basic Path Computation

This example uses the path computation RPC defined in this document to request the computation of the path for the tunnel defined in section 13.1 of [I-D.ietf-teas-yang-te].

In this case, the TE Tunnel has only one primary path with no specific constraints.

```
POST /restconf/operations/ietf-te:te:tunnels-path-compute HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
```

```
{
  "ietf-te:input": {
    "path-compute-info": {
      "ietf-te-path-computation:path-request": [
        {
          "request-id": 1,
          "tunnel-name": "Example_LSP_Tunnel_A_2",
          "encoding": "te-types:lsp-encoding-packet",
          "source": "10.0.0.1",
          "destination": "10.0.0.4",
          "bidirectional": "false",
          "signaling-type": "te-types:path-setup-rsvp"
        }
      ]
    }
  }
}
```

A.2. Path Computation with transient state

This example uses the path computation RPC defined in this document to request the computation of the path for the tunnel defined in section 13.1 of [I-D.ietf-teas-yang-te] requesting some transient state to be reported within the operational datastore, as described Section 3.3.1.

In this case, the TE Tunnel has only one primary path with no specific constraints.

```
POST /restconf/operations/ietf-te:te:tunnels-path-compute HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
```

```
{
  "ietf-te:input": {
    "path-compute-info": {
      "ietf-te-path-computation:path-request": [
        {
          "request-id": 2,
          "tunnel-name": "Example_LSP_Tunnel_A_2",
          "encoding": "te-types:lsp-encoding-packet",
          "source": "10.0.0.1",
          "destination": "10.0.0.4",
          "bidirectional": "false",
          "signaling-type": "te-types:path-setup-rsvp",
          "requested-state": {
            "transaction-id": "example"
          }
        }
      ]
    }
  }
}
```

A.3. Path Computation with Global Path Constraint

This example uses the path computation RPC defined in this document to request the computation of the path for the tunnel defined in section 13.3 of [I-D.ietf-teas-yang-te]. The 'named path constraint' is created in section 13.2 of [I-D.ietf-teas-yang-te] applies to this path computation request.

```
POST /restconf/operations/ietf-te:te:tunnels-path-compute HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
```

```
{
  "ietf-te:input": {
    "path-compute-info": {
      "ietf-te-path-computation:path-request": [
        {
          "request-id": 3,
          "tunnel-name": "Example_LSP_Tunnel_A_4_1",
          "path-name": "Simple_LSP_1",
          "encoding": "te-types:lsp-encoding-packet",
          "source": "10.0.0.1",
          "destination": "10.0.0.4",
          "bidirectional": "false",
          "signaling-type": "path-setup-rsvp",
          "named-path-constraint": "max-hop-3",
          "requested-state": {}
        }
      ]
    }
  }
}
```

A.4. Path Computation with Per-tunnel Path Constraint

This example uses the path computation RPC defined in this document to request the computation of the path for the tunnel defined in section 13.4 of [I-D.ietf-teas-yang-te], using a per tunnel path constraint.

```
POST /restconf/operations/ietf-te:te:tunnels-path-compute HTTP/1.1
Host: example.com
Content-Type: application/yang-data+json
```



```
{
  "ietf-te:input": {
    "path-compute-info": {
      "ietf-te-path-computation:path-request": [
        {
          "request-id": 4,
          "tunnel-name": "Example_LSP_Tunnel_A_4_2",
          "path-name": "path1",
          "encoding": "te-types:lsp-encoding-packet",
          "source": "10.0.0.1",
          "destination": "10.0.0.4",
          "bidirectional": "false",
          "signaling-type": "te-types:path-setup-rsvp",
          "path-metric-bounds": {
            "path-metric-bound": [
              {
                "metric-type": "te-types:path-metric-hop",
                "upper-bound": "3"
              }
            ]
          }
        }
      ]
    }
  }
}
```

A.5. Path Computation result

This example reports the output of the path computation RPC request described in Appendix A.4.

```
HTTP/1.1 200 OK
Host: example.com
Content-Type: application/yang-data+json
```

```

{
  "ietf-te:output": {
    "path-compute-result": {
      "ietf-te-path-computation:response": [
        {
          "response-id": 3,
          "computed-paths-properties": {
            "computed-path-properties": [
              {
                "k-index": "1",
                "path-properties": {
                  "path-route-objects": {
                    "path-route-object": [
                      {
                        "index": "1",
                        "numbered-node-hop": {
                          "node-id": "10.0.0.2"
                        }
                      },
                      {
                        "index": "2",
                        "numbered-node-hop": {
                          "node-id": "10.0.0.4"
                        }
                      }
                    ]
                  }
                }
              }
            ]
          },
          "tunnel-ref": "Example_LSP_Tunnel_A_4_1",
          "primary-path-ref": "path1"
        }
      ]
    }
  }
}

```

Acknowledgments

The authors would like to thank Igor Bryskin and Xian Zhang for participating in the initial discussions that have triggered this work and providing valuable insights.

The authors would like to thank the authors of the TE tunnel YANG data model [I-D.ietf-teas-yang-te], in particular Igor Bryskin, Vishnu Pavan Beeram, Tarek Saad and Xufeng Liu, for their inputs to the discussions and support in having consistency between the Path Computation and TE tunnel YANG data models.

The authors would like to thank Adrian Farrel, Dhruv Dhody, Igor Bryskin, Julien Meuric and Lou Berger for their valuable input to the discussions that has clarified that the path being set up is not necessarily the same as the path that has been previously computed and, in particular to Dhruv Dhody, for his suggestion to describe the need for a path verification phase to check that the actual path being set up meets the required end-to-end metrics and constraints.

The authors would like to thank Aihua Guo, Lou Berger, Shaolong Gan, Martin Bjorklund and Tom Petch for their useful comments on how to define XPath statements in YANG RPCs.

The authors would like to thank Haomian Zheng, Yanlei Zheng, Tom Petch, Aihua Guo and Martin Bjorklund for their review and valuable comments to this document.

Previous versions of document were prepared using 2-Word-v2.0.template.dot.

This document was prepared using kramdown.

Contributors

Victor Lopez
Nokia
Email: victor.lopez@nokia.com

Ricard Vilalta
CTTC
Email: ricard.vilalta@cttc.es

Karthik Sethuraman
NEC
Email: karthik.sethuraman@necam.com

Michael Scharf
Nokia
Email: michael.scharf@gmail.com

Dieter Beller
Nokia
Email: dieter.beller@nokia.com

Gianmarco Bruno
Ericsson
Email: gianmarco.bruno@ericsson.com

Francesco Lazzeri
Ericsson
Email: francesco.lazzeri@ericsson.com

Young Lee
Samsung Electronics
Email: younglee.tx@gmail.com

Carlo Perocchio
Ericsson
Email: carlo.perocchio@ericsson.com

Olivier Dugeon
Orange Labs
Email: olivier.dugeon@orange.com

Julien Meuric
Orange Labs
Email: julien.meuric@orange.com

Authors' Addresses

Italo Busi (editor)
Huawei Technologies
Email: italo.busi@huawei.com

Sergio Belotti (editor)
Nokia
Email: sergio.belotti@nokia.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Anurag Sharma
Google
Email: ansha@google.com

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

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 13 July 2022

V.P. Beeram
T. Saad
Juniper Networks
R. Gandhi
Cisco Systems, Inc.
X. Liu
Volta Networks
I. Bryskin
Individual
9 January 2022

A YANG Data Model for Resource Reservation Protocol (RSVP)
draft-ietf-teas-yang-rsvp-17

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

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 13 July 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
2.1. Prefixes in Data Node Names	3
2.2. Model Tree Diagram	4
3. Model Overview	4
3.1. Module(s) Relationship	5
3.2. Core Features	5
3.3. Optional Features	6
3.4. Data Model Structure	6
3.5. Model Notifications	8
4. RSVP Base YANG Model	9
4.1. Tree Diagram	9
4.2. YANG Module	13
5. RSVP Extended YANG Model	34
5.1. Tree Diagram	34
5.2. YANG Module	36
6. IANA Considerations	45
7. Security Considerations	46
8. Acknowledgement	47
9. Appendix A	47
10. Contributors	53
11. References	54
11.1. Normative References	54
11.2. Informative References	56
Authors' Addresses	57

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 encompasses 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.

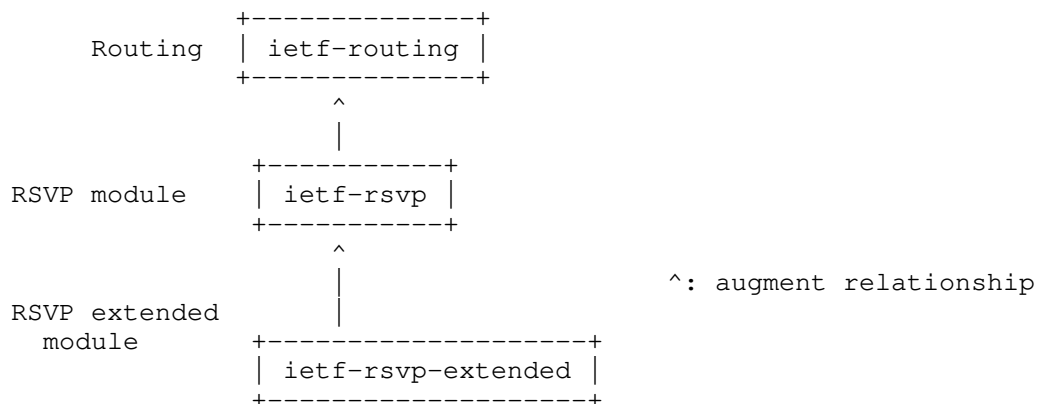


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

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
      +--rw 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)

```

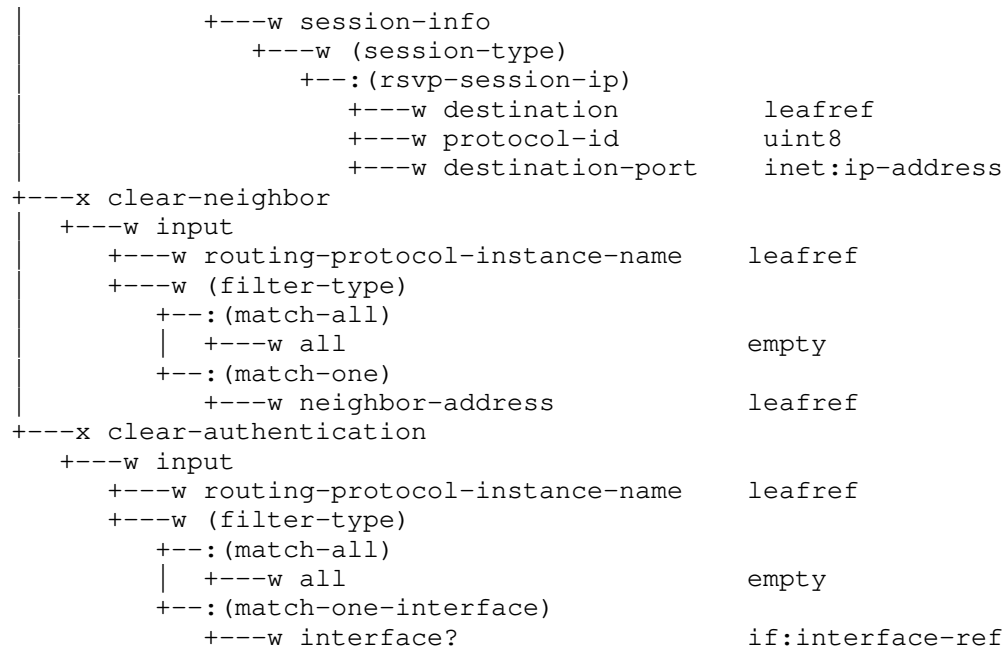


Figure 3: RSVP model tree diagram

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@2021-12-02.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 2021-12-02 {
  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 graceful-restart {
    description
      "RSVP graceful restart local parameters grouping.";
    container graceful-restart {
      description
        "Graceful restart local information.";
      leaf enabled {
        type boolean;
        default "false";
        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";
    default "120";
    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";
    default "120";
    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
       resynchronize 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;
      default "true";
      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 neighbor-graceful-restart {
  description
    "RSVP graceful restart neighbor parameters grouping.";
  container graceful-restart {
    description
      "Graceful restart information.";
    leaf neighbor-restart-time {
      type uint32;
      units "seconds";
      default "120";
      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";
      default "120";
      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 {
```

```
    description
      "Helper mode information.";
    leaf neighbor-restart-time-remaining {
      type uint32;
      units "seconds";
      config false;
      description
        "Number of seconds remaining for neighbor to send Hello
        message after restart.";
      reference "RFC5063";
    }
    leaf neighbor-recovery-time-remaining {
      type uint32;
      units "seconds";
      config false;
      description
        "Number of seconds remaining for neighbor to refresh.";
      reference "RFC5063";
    }
  }
  // helper-mode
}

grouping refresh-reduction {
  description
    "Top level grouping for RSVP refresh reduction parameters.";
  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";
  }
}

grouping authentication {
  description
    "Top level grouping for RSVP authentication parameters.";
  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;
      default "";
      description
        "An authentication key string.";
      reference
        "RFC2747: RSVP Cryptographic Authentication";
    }
    leaf crypto-algorithm {
      type identityref {
        base key-chain:crypto-algorithm;
      }
      mandatory true;
      description
        "Cryptographic algorithm associated with key.";
    }
  }
}

grouping hellos {
  description
    "Top level grouping for RSVP hellos parameters.";
  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.";
  }
}

grouping session-attributes {
  description
```

```
    "Top level grouping for RSVP session properties.";
  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;
    default "";
    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.";
      }
    }
    default "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";
                default "180";
                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;
                }
            }
        }
    }
}
```

```
    }
    mandatory "true";
    description
      "RSVP reservation style.";
  }
  leaf expires-in {
    type uint32;
    units "seconds";
    default "180";
    description
      "Time to expiry (in seconds).";
  }
}

grouping neighbor-attributes {
  description
    "Top level grouping for RSVP neighbor properties.";
  leaf address {
    type inet:ip-address;
    description
      "Address of the RSVP neighbor.";
  }
  leaf epoch {
    type uint32;
    default "0";
    description
      "Neighbor epoch.";
    reference "RFC5063";
  }
  leaf expiry-time {
    type uint32;
    units "seconds";
    default "180";
    description
      "Neighbor expiry time after which the neighbor state is
      purged if no states associated with it.";
  }
  uses neighbor-graceful-restart {
    description
      "Allows configuration applicable to all
      neighbors";
  }
  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";
}
}

grouping packet-statistics {
    description
        "Packet statistics grouping.";
    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.";
        }
    }
}

grouping message-statistics {
    description
        "RSVP protocol statistics grouping.";
    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.";
    }
  }
}
```

```
grouping errors-statistics {
  description
    "Error statistics grouping.";
  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.";
    }
  }
}

grouping statistics {
  description
    "RSVP statistic attributes.";
  container statistics {
    config false;
    description
      "RSVP statistics container.";
    uses message-statistics;
    uses packet-statistics;
    uses errors-statistics;
  }
}

grouping intf-attributes {
  description
    "Top level grouping for RSVP interface properties.";
  uses refresh-reduction;
  uses hellos;
  uses authentication;
  uses statistics;
}
```

```
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.";
        uses session-attributes;
      }
    }
    container neighbors {
      description
        "RSVP neighbors container";
      list neighbor {
        key "address";
        description
          "List of RSVP neighbors";
        uses neighbor-attributes;
      }
    }
  }
```

```
    }
  }
  uses graceful-restart;
}

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;
        }
    }
}
```



```

module: ietf-rsvp-extended

augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp
    /rsvp:graceful-restart:
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 4: 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 5 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@2021-12-02.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@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 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 2021-12-02 {
  description
    "Initial version.";
  reference
    "RFCXXXX: A YANG Data Model for Resource Reservation Protocol
    (RSVP)";
}

grouping graceful-restart-extended {
  description
    "Configuration parameters relating to RSVP Graceful-Restart.";
}

grouping authentication-extended {
  description
    "Configuration parameters relating to RSVP authentication.";
  leaf lifetime {
    type uint32 {
      range "30..86400";
    }
  }
}
```

```
    }
    units "seconds";
    default "30";
    description
      "Life time for each security association.";
    reference
      "RFC2747: RSVP Cryptographic Authentication";
  }
  leaf window-size {
    type uint32 {
      range "1..64";
    }
    default "2";
    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";
    }
    default "1";
    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";
  default "9000";
  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";
  }
  default "3";
  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";
    default "30";
    description
      "Set interval between successive refreshes";
    reference "RFC2205";
  }
  leaf refresh-misses {
    type uint32;
    default "9";
    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";
        }
        default "1500";
        description
            "Configure maximum size (bytes) of a single RSVP Bundle
            message.";
        reference "RFC2961";
    }
    leaf ack-hold-time {
        type uint32;
        units "milliseconds";
        default "9000";
        description
            "Configure hold time in milliseconds for sending RSVP ACK
            message(s).";
        reference "RFC2961";
    }
    leaf ack-max-size {
        type uint32;
        default "1500";
        description
            "Configure max size of a single RSVP ACK message.";
        reference "RFC2961";
    }
    leaf ack-retransmit-time {
        type uint32;
        units "milliseconds";
    }
}
```

```
    default "500";
    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";
    }
    default "1500";
    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 graceful restart*/
augment "/rt:routing/rt:control-plane-protocols/"
    + "rt:control-plane-protocol/rsvp:rsvp/"
    + "rsvp:graceful-restart" {
    description
        "RSVP graceful restart configuration extensions";
    uses graceful-restart-extended;
}

/**
 * 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 5: 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 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

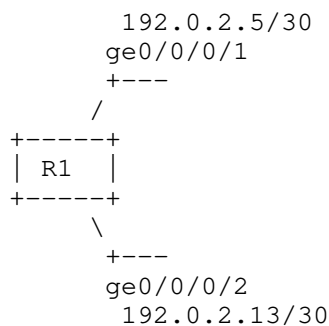


Figure 6: Example of network configuration.

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"
    }
}
],
},
},
},
}
```

Figure 7: Example RSVP JSON encoded data instance tree.

10. Contributors

Himanshu Shah
Ciena

Email: hshah@ciena.com

Xia Chen
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones
Brocade

Email: raqib@Brocade.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

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
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
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

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Prefixes in Data Node Names	3
2. Model Overview	4
2.1. Module Relationship	4
2.2. Model Tree Diagrams	5
2.2.1. RSVP-TE Model Tree Diagram	5
2.2.2. RSVP-TE MPLS Model Tree Diagram	9
2.3. YANG Modules	11
2.3.1. RSVP-TE YANG Module	11
2.3.2. RSVP-TE MPLS YANG Module	24
3. IANA Considerations	36
4. Security Considerations	37
5. Acknowledgement	38
6. Contributors	38
7. References	38
7.1. Normative References	38
7.2. Informative References	40
Authors' Addresses	41

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:

- o those augmenting or extending the base RSVP module that is defined in [I-D.ietf-teas-yang-rsvp]
- o those augmenting or extending the base TE module defined in [I-D.ietf-teas-yang-te]
- o 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.

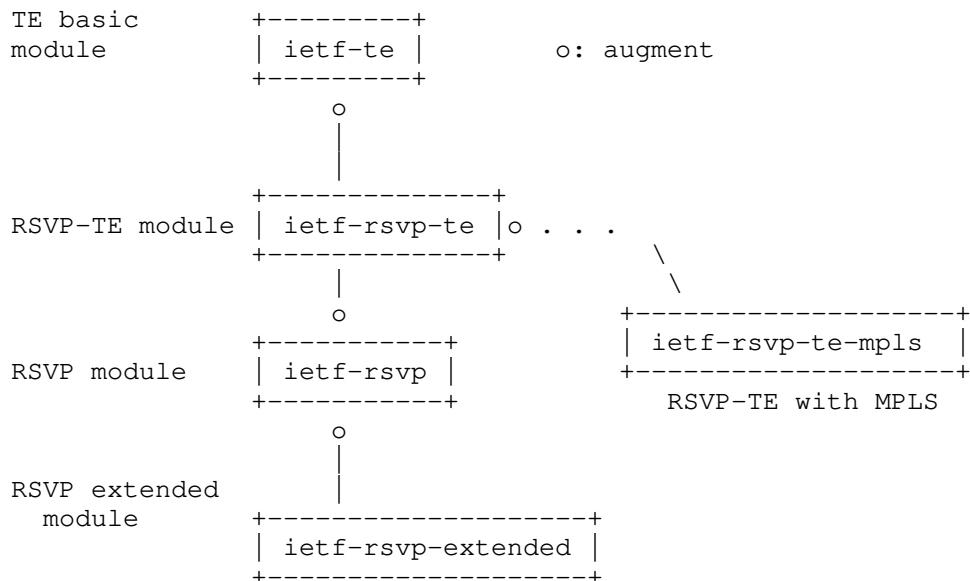


Figure 1: Relationship of RSVP and RSVP-TE modules with other protocol modules

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-sigaled-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-sigaled-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:tasaad.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 preemptions state grouping";
  container interface-softpreemption-state {
    description
      "The RSVP-TE interface preemptions 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 constraints 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 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/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 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2022

X. Liu
IBM Corporation
I. Bryskin
Individual
V. Beeram
T. Saad
Juniper Networks
H. Shah
Ciena
S. Litkowski
Cisco
March 1, 2022

YANG Data Model for SR and SR TE Topologies on MPLS Data Plane
draft-ietf-teas-yang-sr-te-topo-13

Abstract

This document defines a YANG data model for Segment Routing (SR) topology and Segment Routing (SR) traffic engineering (TE) topology, using MPLS data plane.

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 September 2, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Tree Diagrams	3
1.3. Prefixes in Data Node Names	3
2. Modeling Considerations	4
2.1. Segment Routing (SR) MPLS Topology	4
2.2. Segment Routing (SR) MPLS TE Topology	4
2.3. Relations to ietf-segment-routing	7
2.4. Topology Type Modeling	7
2.5. Topology Attributes	7
2.6. Node Attributes	7
2.7. Link Attributes	9
3. Model Structure	10
4. YANG Module	11
5. IANA Considerations	20
6. Security Considerations	21
7. References	23
7.1. Normative References	23
7.2. Informative References	25
Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations	26
A.1. SR MPLS Topology State Module	26
Appendix B. Data Tree Example	29
B.1. SR MPLS Topology with TE Not Enabled	30
B.2. SR MPLS Topology with TE Enabled	37
Appendix C. Contributors	52
Authors' Addresses	52

1. Introduction

This document defines a YANG [RFC7950] data model for describing the presentations of Segment Routing (SR) topology and Segment Routing (SR) traffic engineering (TE) topology. The version of the model limits the transport type to an MPLS dataplane.

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

- o augment
- o data model
- o data node

1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

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
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
l3t	ietf-l3-unicast-topology	[RFC8346]
sr-cmn	ietf-segment-routing-common	[RFC9020]
tet	ietf-te-topology	[RFC8795]
tet-pkt	ietf-te-topology-packet	[I-D.ietf-teas-yang-l3-te-topo]

Table 1: Prefixes and Corresponding YANG Modules

2. Modeling Considerations

2.1. Segment Routing (SR) MPLS Topology

The Layer 3 network topology model is discussed in [RFC8346]. The Segment Routing (SR) MPLS topology model proposed in this document augments and uses the `ietf-l3-unicast-topology` module defined in [RFC8346]. SR MPLS related attributes are covered in the `ietf-sr-mpls-topology` module.

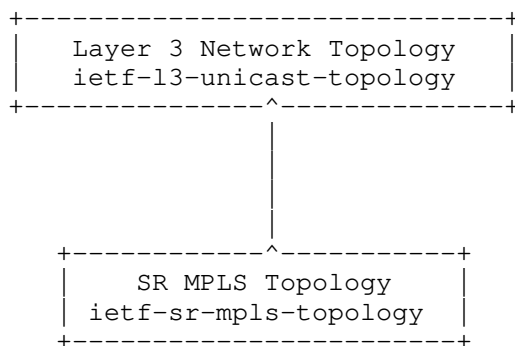


Figure 1: SR MPLS Topology Augmentation

2.2. Segment Routing (SR) MPLS TE Topology

A Segment Routing (SR) MPLS TE topology is an instance of SR MPLS topology with TE enabled. In order to instantiate an SR MPLS TE topology, the `ietf-sr-mpls-topology` module defined in this document can be used together with the `ietf-te-topology` module defined in [RFC8795] and the `ietf-te-topology-packet` module defined in [I-D.ietf-teas-yang-l3-te-topo]. All these modules directly or indirectly augment the `ietf-network-topology` module defined in [RFC8345], as shown in Figure 2.

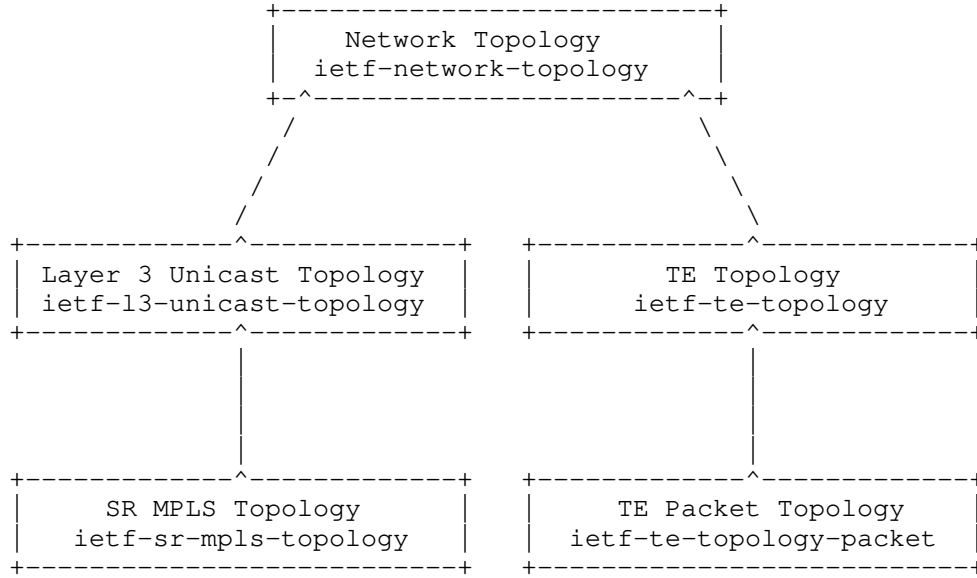


Figure 2: SR TE Topology Instance Inheritance Relations

Figure 3 shows the data structure of an SR TE topology instance. Because of the augmentation relationships shown in Figure 2, a data instance of an SR MPLS TE topology contains the capabilities from all these modules, so that the data includes the attributes from `ietf-network-topology`, `ietf-l3-unicast-topology`, `ietf-sr-mpls-topology`, `ietf-te-topology`, and `ietf-te-topology-packet`.

<pre> ietf-network-topology: network-id (key) network-types: { l3-unicast-topology: { sr-mpls{} } te-topology: { packet{} } } <other network topology attributes> </pre>		
<pre> ietf-l3-unicast-topology: <L3 unicast attributes> </pre>	<pre> ietf-te-topology: <TE attributes> </pre>	
<pre> ietf-sr-mpls-topology: <SR MPLS attributes> </pre>	<pre> ietf-te-topology-packet: <TE packet attributes> </pre>	

Figure 3: SR TE topology instance data structure

Each type of topology is indicated by a YANG presence container which augments "network-types" as defined in [RFC8345].² For the five types of topologies above, the data representations are:

Base network topology [RFC8345]:

```
/nw:networks/nw:network/nw:network-types
```

Layer 3 Unicast Topology [RFC8346]:

```
/nw:networks/nw:network/nw:network-types/l3t:l3-unicast-topology
```

SR MPLS Topology (defined in this document):

```
/nw:networks/nw:network/nw:network-types/l3t:l3-unicast-topology/
srmt:sr-mpls
```

TE Topology [RFC8795]:

```
/nw:networks/nw:network/nw:network-types/tet:te-topology
```

TE Packet Topology [I-D.ietf-teas-yang-l3-te-topo]:

```
/nw:networks/nw:network/nw:network-types/tet:te-topology/tet-
pkt:packet
```

2.3. Relations to ietf-segment-routing

[RFC9020] defines ietf-segment-routing that is a model intended to be used on network elements to configure or operate segment routing; ietf-sr-mpls-topology defined in this document is intended to be used on a controller for the network-wide operations such as path computation.

SR MPLS topology model shares many modeling constructs defined in ietf-segment-routing. The module ietf-sr-mpls-topology uses the types and groupings defined in ietf-segment-routing.

2.4. Topology Type Modeling

A new topology type is defined in this document, to indicate a topology that is a Segment Routing (SR) topology on an MPLS dataplane.

```
augment /nw:networks/nw:network/nw:network-types
  /l3t:l3-unicast-topology:
    +--rw sr-mpls!
```

Section 4.4.8 of RFC 8345 describes how network types are represented using nested presence container. In this document, the presence container sr-mpls is used for such a purpose.

2.5. Topology Attributes

The Segment Routing attributes with topology-wide impacts are modeled by augmenting the container "l3-topology-attributes" in the L3 topology model [RFC8346]. SRGB (Segment Routing Global Block) is covered in this augmentation. A SR domain is mapped to a topology in this model.

```
augment /nw:networks/nw:network/l3t:l3-topology-attributes:
  +--rw sr-mpls
    +--rw srgb* [lower-bound upper-bound]
      +--rw lower-bound    uint32
      +--rw upper-bound    uint32
```

2.6. Node Attributes

The Segment Routing attributes within the node scope are modeled by augmenting the sub tree /nw:networks/nw:network/nw:node/ in the L3 topology model [RFC8346].

The SR attributes that have node-scope impact are modeled by augmenting the container "l3-node-attributes" in the L3 topology

model, including the SR capabilities, SRGB (Segment Routing Global Block), and SRLB (Segment Routing Local Block) specified on this mode. This model also provides the information about how these SR attributes are learned:

The presence container `sr-mpls` is used to indicate that SR MPLS is enabled on this node when the container is present.

```
augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
  +--rw sr-mpls!
    +--rw srgb* [lower-bound upper-bound]
      | +--rw lower-bound    uint32
      | +--rw upper-bound    uint32
    +--rw srlb* [lower-bound upper-bound]
      | +--rw lower-bound    uint32
      | +--rw upper-bound    uint32
    +--rw msds {msd}?
      | +--rw node-msd* [msd-type]
      |   +--rw msd-type      identityref
      |   +--rw msd-value?    uint8
    +--ro information-source?      enumeration
    +--ro information-source-instance?  string
    +--ro information-source-state
      +--ro credibility-preference?  uint16
```

The SR attributes that are related to a IGP-Prefix segment are modeled by augmenting the list entry "prefix" in the L3 topology model:

```
augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes
  /l3t:prefix:
    +--rw sr-mpls!
      +--rw sids
        +--rw sid* [algorithm]
          +--rw value-type?      enumeration
          +--rw start-sid        uint32
          +--rw range?           uint32
          +--rw algorithm        identityref
          +--rw last-hop-behavior? enumeration
          | {sid-last-hop-behavior}?
          +--rw is-local?        boolean
          +--rw is-node?         boolean
          +--ro is-readvertisement?  boolean
```

2.7. Link Attributes

A link in the topology model connects the termination point on the source node to the termination point on the destination node. When such a link is instantiated, the bindings between the nodes and the corresponding Adj-SIDs are formed, and the resulting FIB entries are installed.

A link in the topology model is mapped to an SR Adjacency Segment, formed by a pair of interfaces on two respective adjacent nodes. The SR Adjacency Segment attributes are modeled by augmenting the link attributes of the L3 topology model [RFC8346]. The modeling structure is as follows:

```
augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
  +--rw sr-mpls!
    +--rw msds {msd}?
      +--rw link-msd* [msd-type]
        +--rw msd-type      identityref
        +--rw msd-value?    uint8
    +--rw sids
      +--rw sid* [value-type sid]
        +--rw value-type      enumeration
        +--rw sid              uint32
        +--rw address-family? enumeration
        +--rw is-eligible-for-protection? boolean
        +--rw is-local?       boolean
        +--rw is-part-of-set?  boolean
        +--rw is-persistent?   boolean
        +--rw is-on-lan?       boolean
        +--rw weight?          uint8
    +--ro information-source?   enumeration
    +--ro information-source-instance? string
    +--ro information-source-state
      +--ro credibility-preference? uint16
```

IGPs [RFC8665] [RFC8666] [RFC8667] and BGP-LS [RFC7752] [I-D.ietf-idr-bgp-ls-segment-routing-ext] can be supported by the model, the leaf "information-source" is used to indicate where the information is from.

The bundling capability of the Adjacency Segment is achieved by re-using the existing modeling construct (i.e. "bundle-stack-level") under /nw:networks/nw:network/nt:link/tet:te [RFC8795]

The presence container sr-mpls is used to indicate that SR SMPL is enabled on this link when the container is present.

3. Model Structure

The model tree structure of the Segment Routing (SR) topology module is as shown below:

```

module: ietf-sr-mpls-topology
  augment /nw:networks/nw:network/nw:network-types
    /l3t:l3-unicast-topology:
      +--rw sr-mpls!
  augment /nw:networks/nw:network/l3t:l3-topology-attributes:
    +--rw sr-mpls
      +--rw srgb* [lower-bound upper-bound]
        +--rw lower-bound    uint32
        +--rw upper-bound    uint32
  augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
    +--rw sr-mpls!
      +--rw srgb* [lower-bound upper-bound]
        | +--rw lower-bound    uint32
        | +--rw upper-bound    uint32
      +--rw srlb* [lower-bound upper-bound]
        | +--rw lower-bound    uint32
        | +--rw upper-bound    uint32
      +--rw msds {msd}?
        | +--rw node-msd* [msd-type]
        |   +--rw msd-type    identityref
        |   +--rw msd-value?  uint8
      +--ro information-source?      enumeration
      +--ro information-source-instance?  string
      +--ro information-source-state
        +--ro credibility-preference?  uint16
  augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes
    /l3t:prefix:
      +--rw sr-mpls!
      +--rw sids
        +--rw sid* [algorithm]
          +--rw value-type?      enumeration
          +--rw start-sid        uint32
          +--rw range?           uint32
          +--rw algorithm        identityref
          +--rw last-hop-behavior? enumeration
          | {sid-last-hop-behavior}?
          +--rw is-local?        boolean
          +--rw is-node?         boolean
          +--ro is-readvertisement? boolean
  augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
    +--rw sr-mpls!
    +--rw msds {msd}?

```

```

    +--rw link-msd* [msd-type]
      +--rw msd-type      identityref
      +--rw msd-value?    uint8
    +--rw sids
      +--rw sid* [value-type sid]
        +--rw value-type      enumeration
        +--rw sid              uint32
        +--rw address-family?  enumeration
        +--rw is-eligible-for-protection? boolean
        +--rw is-local?        boolean
        +--rw is-part-of-set?   boolean
        +--rw is-persistent?    boolean
        +--rw is-on-lan?        boolean
        +--rw weight?           uint8
    +--ro information-source?      enumeration
    +--ro information-source-instance? string
    +--ro information-source-state
      +--ro credibility-preference? uint16

```

4. YANG Module

This module references [RFC7752], [RFC8345], [RFC8346], [RFC8476], [RFC8491], [RFC8665], [RFC8666], [RFC8667], [RFC8814], [RFC9020], and [I-D.ietf-idr-bgp-ls-segment-routing-ext].

```

<CODE BEGINS> file "ietf-sr-mpls-topology@2021-10-23.yang"
module ietf-sr-mpls-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sr-mpls-topology";
  prefix "srmt";

  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";
  }
  import ietf-l3-unicast-topology {
    prefix "l3t";
    reference "RFC 8346: A YANG Data Model for Layer 3 Topologies";
  }
  import ietf-segment-routing-common {
    prefix "sr-cmn";
    reference "RFC 9020: YANG Data Model for Segment Routing";
  }

```

```
}

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@cisco.com>

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

  Editor:    Stephane Litkowski
             <mailto:stephane.litkowski@orange.com>";

description
  "YANG data model for representing and manipulating Segment
  Routing Topologies on MPLS Data Plane.

  Copyright (c) 2021 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 2021-10-23 {
  description "Initial revision";
  reference
```

```
    "RFC XXXX: YANG Data Model for SR and SR TE Topologies on MPLS
      Data Plane";
  }

  identity msd-base-type {
    description
      "Base identity for MSD Type";
  }

  identity base-mpls-msd {
    base msd-base-type;
    description
      "Base MPLS Imposition MSD.";
    reference
      "RFC 8491: Signaling Maximum SID Depth (MSD) Using IS-IS";
  }

  identity erld-msd {
    base msd-base-type;
    description
      "ERLD-MSD is defined to advertise the ERLD.";
    reference
      "RFC 8662: Entropy Label for Source Packet Routing in
        Networking (SPRING) Tunnels";
  }

  feature msd {
    description
      "Support of signaling MSD (Maximum SID Depth) in IGP.";
    reference
      "RFC 8476: Signaling Maximum SID Depth (MSD) Using OSPF.
        RFC 8491: Signaling Maximum SID Depth (MSD) Using IS-IS.
        RFC 8814: Signaling Maximum SID Depth (MSD) Using the Border
        Gateway Protocol - Link State.";
  }

  grouping sr-mpls-topology-type {
    description
      "Identifies the SR-MPLS topology type. This type of network
        topologies use Segment Routing (SR) technology over the MPLS
        data plane";
    container sr-mpls {
      presence "Indicates SR-MPLS topology";
      description
        "Its presence identifies the SR MPLS topology type.";
    }
  }
}
```

```
augment "/nw:networks/nw:network/nw:network-types/"
+ "l3t:l3-unicast-topology" {
  description
    "Defines the SR MPLS topology type.";
  uses sr-mpls-topology-type;
}

augment "/nw:networks/nw:network/l3t:l3-topology-attributes" {
  when "../nw:network-types/l3t:l3-unicast-topology/srmt:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment topology configuration";
  uses sr-mpls-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node/l3t:l3-node-attributes" {
  when "../nw:network-types/l3t:l3-unicast-topology/"
  + "srmt:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment node configuration.";
  uses sr-node-attributes;
}

augment "/nw:networks/nw:network/nw:node/l3t:l3-node-attributes"
+ "/l3t:prefix" {
  when "../nw:network-types/l3t:l3-unicast-topology/"
  + "srmt:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment node prefix.";
  uses sr-node-prefix-attributes;
}

augment "/nw:networks/nw:network/nt:link/l3t:l3-link-attributes" {
  when "../nw:network-types/l3t:l3-unicast-topology/"
  + "srmt:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment link configuration";
  uses sr-link-attributes;
}

grouping sr-mpls-topology-attributes {
  description "SR MPLS topology scope attributes.";
  container sr-mpls {
    description
      "Containing SR attributes.";
  }
}
```

```
    uses sr-cmn:srgb;
  } // sr
} // sr-mpls-topology-attributes

grouping information-source-attributes {
  description
    "The attributes identifying source that has provided the
    related information, and the source credibility.";
  leaf information-source {
    type enumeration {
      enum "unknown" {
        description "The source is unknown.";
      }
      enum "locally-configured" {
        description "Configured entity.";
      }
      enum "ospfv2" {
        description "OSPFv2.";
        reference
          "RFC 8665: OSPF Extensions for Segment Routing.";
      }
      enum "ospfv3" {
        description "OSPFv3.";
        reference
          "RFC 8666: OSPFv3 Extensions for Segment Routing.";
      }
      enum "isis" {
        description "ISIS.";
        reference
          "RFC 8667: IS-IS Extensions for Segment Routing.";
      }
      enum "bgp-ls" {
        description "BGP-LS.";
        reference
          "RFC 7752: North-Bound Distribution of Link-State and
          Traffic Engineering (TE) Information Using BGP.
          I-D.ietf-idr-bgp-ls-segment-routing-ext:
          BGP Link-State extensions for Segment Routing.";
      }
      enum "system-processed" {
        description "System processed entity.";
      }
      enum "other" {
        description "Other 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.";
        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.";
        }
    }
} // information-source-attributes

grouping sr-node-attributes {
    description "SR node scope attributes.";
    container sr-mpls {
        presence "Presence indicates SR is enabled.";
        description
            "Containing SR attributes.";
        uses sr-cmn:srgb;
        uses sr-cmn:srlb;
        container msds {
            if-feature "msd";
            description
                "MSDs on the node.";
            list node-msd {
                key "msd-type";
                leaf msd-type {
                    type identityref {
                        base msd-base-type;
                    }
                    description
                        "MSD Type.";
                }
                leaf msd-value {
```

```
        type uint8;
        description
            "MSD value, in the range of 0-255. Node MSD is the
             lowest MSD supported by the node.";
    }
    description
        "List of node MSDs. A node may have a list of MSD values,
         with at most one value for each msd-type.";
    reference
        "RFC 8491: Signaling Maximum SID Depth (MSD) Using
         IS-IS";
    }
}
// Operational state data
uses information-source-attributes;
} // sr
} // sr-node-attributes

grouping sr-node-prefix-attributes {
    description "Containing SR attributes for a prefix.";
    container sr-mpls {
        presence "Presence indicates SR is enabled.";
        description
            "Containing SR attributes for a prefix.";
        container sids {
            description
                "Containing Prefix SIDs assigned to this prefix.";
            list sid {
                key "algorithm";
                description
                    "A list of SIDs with their properties.";
                uses sr-cmn:prefix-sid-attributes;
                uses sr-cmn:last-hop-behavior;
                leaf is-local {
                    type boolean;
                    default false;
                    description
                        "'true' if the SID is local.";
                }
                leaf is-node {
                    type boolean;
                    default false;
                    description
                        "'true' if the Prefix-SID refers to the router
                         identified by the prefix. Typically, the leaf
                         'is-node' (N-Flag) is set on Prefix-SIDs attached to a
                         router loopback address.";
                }
            }
        }
    }
}
```



```
    leaf is-readvertisement {
      type boolean;
      config false;
      description
        "'true' if the prefix to which this Prefix-SID is
        attached, has been propagated by the router from
        another topology by redistribution.";
      reference
        "RFC 8667: IS-IS Extensions for Segment Routing.
        Sec 2.1.";
    }
  }
} // sr
} // sr-node-prefix-attributes

grouping sr-link-attributes {
  description "SR link scope attributes";
  container sr-mpls {
    presence "Presence indicates SR is enabled.";
    description
      "Containing SR attributes.";

    container msds {
      if-feature "msd";
      description
        "MSDs on the link.";
      list link-msd {
        key "msd-type";
        leaf msd-type {
          type identityref {
            base msd-base-type;
          }
          description
            "MSD Type.";
        }
        leaf msd-value {
          type uint8;
          description
            "MSD value, in the range of 0-255.";
        }
      }
      description
        "List of link MSDs. A link may have a list of MSD values,
        with at most one value for each msd-type.";
      reference
        "RFC 8491: Signaling Maximum SID Depth (MSD) Using
        IS-IS";
    }
  }
}
```

```
}
container sids {
  description
    "Containing Adjacency SIDs assigned to this link.";
  list sid {
    key "value-type sid";
    description
      "A list of SIDs with their properties.";
    uses sr-cmn:sid-value-type;
    leaf sid {
      type uint32;
      mandatory true;
      description
        "Adjacency SID, which can be either IGP-Adjacency SID
        or BGP PeerAdj SID, depending on the context.";
    }
    leaf address-family {
      type enumeration {
        enum "ipv4" {
          description
            "The Adj-SID refers to an adjacency with outgoing
            IPv4 encapsulation.";
        }
        enum "ipv6" {
          description
            "The Adj-SID refers to an adjacency with outgoing
            IPv6 encapsulation.";
        }
      }
      default "ipv4";
      description
        "This leaf defines the F-Flag (Address-Family flag) of
        the SID.";
    }
    leaf is-eligible-for-protection {
      type boolean;
      default false;
      description
        "'true' if the SID is is eligible for protection.";
      reference
        "RFC 8402: Segment Routing Architecture. Sec. 3.4.";
    }
    leaf is-local {
      type boolean;
      default false;
      description
        "'true' if the SID is local.";
    }
  }
}
```

```
    leaf is-part-of-set {
      type boolean;
      default false;
      description
        "'true' if the SID is part of a set.";
    }
    leaf is-persistent {
      type boolean;
      default true;
      description
        "'true' if the SID is persistently allocated.";
    }
    leaf is-on-lan {
      type boolean;
      default false;
      description
        "'true' if on a lan.";
    }
    leaf weight {
      type uint8;
      description
        "The value represents the weight of the SID for the
        purpose of load balancing. The use of the weight
        is defined in RFC 8402.";
      reference
        "RFC 8402: Segment Routing Architecture. Sec. 3.4.";
    }
  }
}
uses information-source-attributes;
} // sr
} // sr-tp-attributes
}
<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-sr-mpls-topology
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-sr-mpls-topology-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 [RFC6020]:

name: ietf-sr-mpls-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-sr-mpls-topology
prefix: srmt
reference: RFC XXXX

name: ietf-sr-mpls-topology-state
namespace: urn:ietf:params:xml:ns:yang:ietf-sr-mpls-topology-state
prefix: srmt-s
reference: RFC XXXX

6. 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/l3t:l3-unicast-topology/sr-mpls`

This subtree specifies the SR MPLS topology type. Modifying the configurations can make SR MPLS topology type invalid and cause interruption to all SR networks.

`/nw:networks/nw:network/l3t:l3-topology-attributes/sr`

This subtree specifies the topology-wide configurations, including the SRGB (Segment Routing Global Block). Modifying the configurations here can cause traffic disabled or rerouted in this topology and the connected topologies.

`/nw:networks/nw:network/nw:node/l3t:l3-node-attributes`

This subtree specifies the SR configurations for nodes. Modifying the configurations in this subtree can add, remove, or modify SR nodes, causing traffic disabled or rerouted in the specified nodes and the related TE topologies.

`/nw:networks/nw:network/nt:link/l3t:l3-link-attributes/sr`

This subtree specifies the configurations for SR Adjacency Segments. Modifying the configurations in this subtree can add, remove, or modify SR Adjacency Segments causing traffic disabled or rerouted on the specified SR adjacencies, the related nodes, and the related SR MPLS 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/l3t:l3-unicast-topology/sr-mpls`

Unauthorized access to this subtree can disclose the SR MPLS topology type.

`/nw:networks/nw:network/l3t:l3-topology-attributes/sr`

Unauthorized access to this subtree can disclose the topology-wide configurations, including the SRGB (Segment Routing Global Block).

`/nw:networks/nw:network/nw:node/l3t:l3-node-attributes`

Unauthorized access to this subtree can disclose the operational state information of the SR nodes.

/nw:networks/nw:network/nt:link/l3t:l3-link-attributes/sr
Unauthorized access to this subtree can disclose the operational
state information of SR Adjacency Segments.

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>>.
- [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>>.
- [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>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [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>>.
- [RFC8476] Tantsura, J., Chunduri, U., Aldrin, S., and P. Psenak, "Signaling Maximum SID Depth (MSD) Using OSPF", RFC 8476, DOI 10.17487/RFC8476, December 2018, <<https://www.rfc-editor.org/info/rfc8476>>.
- [RFC8491] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling Maximum SID Depth (MSD) Using IS-IS", RFC 8491, DOI 10.17487/RFC8491, November 2018, <<https://www.rfc-editor.org/info/rfc8491>>.
- [RFC8814] Tantsura, J., Chunduri, U., Talaulikar, K., Mirsky, G., and N. Triantafyllis, "Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State", RFC 8814, DOI 10.17487/RFC8814, August 2020, <<https://www.rfc-editor.org/info/rfc8814>>.
- [RFC9020] Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", RFC 9020, DOI 10.17487/RFC9020, May 2021, <<https://www.rfc-editor.org/info/rfc9020>>.
- [I-D.ietf-teas-yang-l3-te-topo]
Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H., and O. G. D. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-12 (work in progress), October 2021.

7.2. Informative References

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.
- [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>>.
- [I-D.ietf-idr-bgp-ls-segment-routing-ext]
Previdi, S., Talaulikar, K., Filsfils, C., Gredler, H., and M. Chen, "Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing", draft-ietf-idr-bgp-ls-segment-routing-ext-18 (work in progress), April 2021.

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

The YANG module `ietf-sr-mpls-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-sr-mpls-topology-state`, is defined as state model, which mirrors the module `ietf-sr-mpls-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-sr-mpls-topology-state`, is redundant and SHOULD NOT be supported by implementations that support NMDA.

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

A.1. SR MPLS Topology State Module

This module references [RFC8345] and [RFC8346].

```
<CODE BEGINS> file "ietf-sr-mpls-topology-state@2021-10-23.yang"
module ietf-sr-mpls-topology-state {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-sr-mpls-topology-state";
  prefix "srmt-s";

  import ietf-sr-mpls-topology {
    prefix "srmt";
  }
  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";
  }
  import ietf-l3-unicast-topology-state {
    prefix "l3t-s";
    reference "RFC 8346: A YANG Data Model for Layer 3 Topologies";
  }
  import ietf-segment-routing-common {
```

```
    prefix "sr-cmn";  
    reference "RFC 9020: YANG Data Model for Segment Routing";  
}
```

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@cisco.com>>

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

Editor: Stephane Litkowski
<<mailto:stephane.litkowski@orange.com>>";

description

"YANG data model for representing operational state information
of Segment Routing Topologies on MPLS data plane, when NMDA is
not supported.

Copyright (c) 2021 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 2021-10-23 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for SR and SR TE Topologies on MPLS
    Data Plane";
}

augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
+ "l3t-s:l3-unicast-topology" {
  description
    "Defines the SR MPLS topology type.";
  uses srmt:sr-mpls-topology-type;
}

augment "/nw-s:networks/nw-s:network/"
+ "l3t-s:l3-topology-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
+ "srmt-s:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment topology configuration";
  uses srmt:sr-mpls-topology-attributes;
}

augment "/nw-s:networks/nw-s:network/nw-s:node/"
+ "l3t-s:l3-node-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
+ "srmt-s:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment node configuration.";
  uses srmt:sr-node-attributes;
}

augment "/nw-s:networks/nw-s:network/nw-s:node/"
+ "l3t-s:l3-node-attributes/l3t-s:prefix" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
+ "srmt-s:sr-mpls" {
    description "Augment only for SR MPLS topology.";
  }
  description "Augment node prefix.";
  uses srmt:sr-node-prefix-attributes;
}

augment "/nw-s:networks/nw-s:network/nt-s:link/"
+ "l3t-s:l3-link-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
+ "srmt-s:sr-mpls" {
```

```
        description "Augment only for SR MPLS topology.";
    }
    description "Augment link configuration";
    uses srmt:sr-link-attributes;
}

grouping sr-mpls-topology-attributes {
    description "SR MPLS topology scope attributes.";
    container sr-mpls {
        description
            "Containing SR attributes.";
        uses sr-cmn:srgb;
    } // sr
} // sr-mpls-topology-attributes
}
<CODE ENDS>
```

Appendix B. Data Tree Example

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-sr-mpls-topology" for the topology that is depicted in the following diagram.

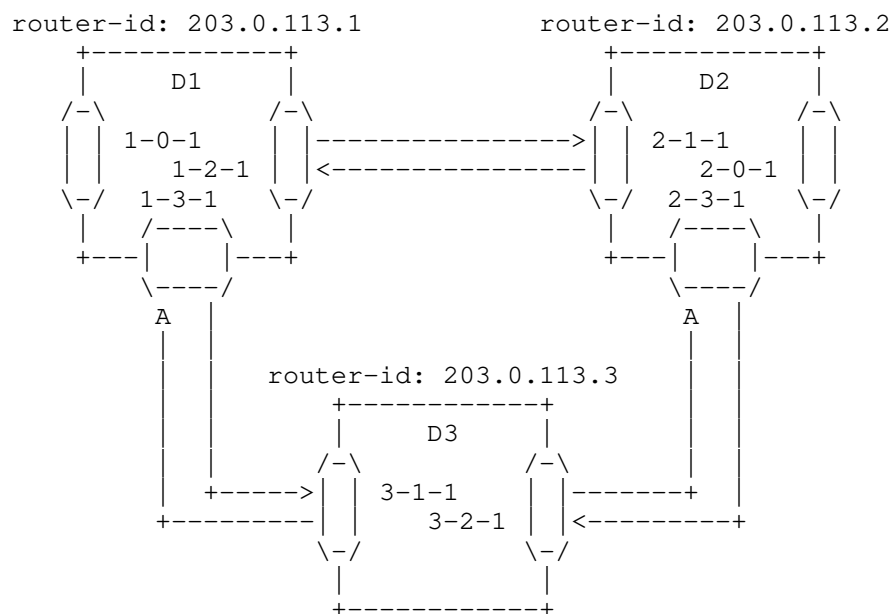


Figure 4: Example SR MPLS Topology

The corresponding instance data tree is depicted below. Note that some lines have been wrapped to adhere to the 72-character line limitation of RFCs.

B.1. SR MPLS Topology with TE Not Enabled

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-types": {
          "ietf-l3-unicast-topology:l3-unicast-topology": {
            "ietf-sr-mpls-topology:sr-mpls": {}
          }
        },
        "network-id": "sr-mpls-topo-example",
        "ietf-l3-unicast-topology:l3-topology-attributes": {
          "ietf-sr-mpls-topology:sr-mpls": {
            "srgb": [
              {
                "lower-bound": 16000,
                "upper-bound": 23999
              }
            ]
          }
        }
      }
    ]
  }
}
```

```

    ]
  }
},
"node": [
  {
    "node-id": "D1",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "1-0-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
          "unnumbered-id": 101
        }
      },
      {
        "tp-id": "1-2-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
          "unnumbered-id": 121
        }
      },
      {
        "tp-id": "1-3-1",
        "ietf-l3-unicast-topology:l3-termination-point-attributes": {
          "unnumbered-id": 131
        }
      }
    ],
    "ietf-l3-unicast-topology:l3-node-attributes": {
      "router-id": ["203.0.113.1"],
      "prefix": [
        {
          "prefix": "203.0.113.1/32",
          "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {
              "sid": [
                {
                  "algorithm": "prefix-sid-algorithm-shortest-path",
                  "start-sid": 101,
                  "range": 1,
                  "is-local": false,
                  "is-node": true
                }
              ]
            }
          }
        }
      ]
    },
    "ietf-sr-mpls-topology:sr-mpls": {
      "srgb": [

```

```

        {
            "lower-bound": 16000,
            "upper-bound": 23999
        }
    ],
    "srlb": [
        {
            "lower-bound": 15000,
            "upper-bound": 15999
        }
    ]
}
},
{
    "node-id": "D2",
    "ietf-network-topology:termination-point": [
        {
            "tp-id": "2-0-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id": 201
            }
        },
        {
            "tp-id": "2-1-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id": 211
            }
        },
        {
            "tp-id": "2-3-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id": 231
            }
        }
    ],
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "router-id": ["203.0.113.2"],
        "prefix": [
            {
                "prefix": "203.0.113.2/32",
                "ietf-sr-mpls-topology:sr-mpls": {
                    "sids": {
                        "sid": [
                            {
                                "algorithm": "prefix-sid-algorithm-shortest-path",
                                "start-sid": 102,
                                "range": 1,

```

```

        "is-local": false,
        "is-node": true
      }
    ]
  }
},
"ietf-sr-mpls-topology:sr-mpls": {
  "srgb": [
    {
      "lower-bound": 16000,
      "upper-bound": 23999
    }
  ],
  "srlb": [
    {
      "lower-bound": 15000,
      "upper-bound": 15999
    }
  ]
}
},
{
  "node-id": "D3",
  "ietf-network-topology:termination-point": [
    {
      "tp-id": "3-1-1",
      "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 311
      }
    },
    {
      "tp-id": "3-2-1",
      "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 321
      }
    }
  ],
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "router-id": ["203.0.113.3"],
    "prefix": [
      {
        "prefix": "203.0.113.3/32",
        "ietf-sr-mpls-topology:sr-mpls": {
          "sids": {
            "sid": [

```



```

        {
"algorithm": "prefix-sid-algorithm-shortest-path",
    "start-sid": 103,
    "range": 1,
    "is-local": false,
    "is-node": true
        }
    ]
}
}
],
"ietf-sr-mpls-topology:sr-mpls": {
    "srgb": [
        {
            "lower-bound": 16000,
            "upper-bound": 23999
        }
    ],
    "srlb": [
        {
            "lower-bound": 15000,
            "upper-bound": 15999
        }
    ]
}
}
],
"ietf-network-topology:link": [
    {
        "link-id": "D1,1-2-1,D2,2-1-1",
        "source": {
            "source-node": "D1",
            "source-tp": "1-2-1"
        },
        "destination": {
            "dest-node": "D2",
            "dest-tp": "2-1-1"
        },
        "ietf-l3-unicast-topology:l3-link-attributes": {
            "metric1": "100",
            "ietf-sr-mpls-topology:sr-mpls": {
                "sids": {
                    "sid": [
                        {
                            "value-type": "index",
                            "sid": 121,

```

```

        "is-local": true
      }
    ]
  }
}
},
{
  "link-id": "D2,2-1-1,D1,1-2-1",
  "source": {
    "source-node": "D2",
    "source-tp": "2-1-1"
  },
  "destination": {
    "dest-node": "D1",
    "dest-tp": "1-2-1"
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric1": "100",
    "ietf-sr-mpls-topology:sr-mpls": {
      "sids": {
        "sid": [
          {
            "value-type": "index",
            "sid": 211,
            "is-local": true
          }
        ]
      }
    }
  }
},
{
  "link-id": "D1,1-3-1,D3,3-1-1",
  "source": {
    "source-node": "D1",
    "source-tp": "1-3-1"
  },
  "destination": {
    "dest-node": "D3",
    "dest-tp": "3-1-1"
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric1": "100",
    "ietf-sr-mpls-topology:sr-mpls": {
      "sids": {
        "sid": [
          {

```

```

        "value-type": "index",
        "sid": 131,
        "is-local": true
    }
    ]
}
}
},
{
    "link-id": "D3,3-1-1,D1,1-3-1",
    "source": {
        "source-node": "D3",
        "source-tp": "3-1-1"
    },
    "destination": {
        "dest-node": "D1",
        "dest-tp": "1-3-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {
                "sid": [
                    {
                        "value-type": "index",
                        "sid": 311,
                        "is-local": true
                    }
                ]
            }
        }
    }
},
{
    "link-id": "D2,2-3-1,D3,3-2-1",
    "source": {
        "source-node": "D2",
        "source-tp": "2-3-1"
    },
    "destination": {
        "dest-node": "D3",
        "dest-tp": "3-2-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {

```

```

        "sid": [
            {
                "value-type": "index",
                "sid": 231,
                "is-local": true
            }
        ]
    },
}
},
{
    "link-id": "D3,3-2-1,D2,2-3-1",
    "source": {
        "source-node": "D3",
        "source-tp": "3-2-1"
    },
    "destination": {
        "dest-node": "D2",
        "dest-tp": "2-3-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {
                "sid": [
                    {
                        "value-type": "index",
                        "sid": 321,
                        "is-local": true
                    }
                ]
            }
        }
    }
}
]
}
}
}
}
}

```

B.2. SR MPLS Topology with TE Enabled

In this section, the example below shows an instance data of a overlay topology as shown in Figure 4. Some attributes are from the underlay topology shown in Figure 5.

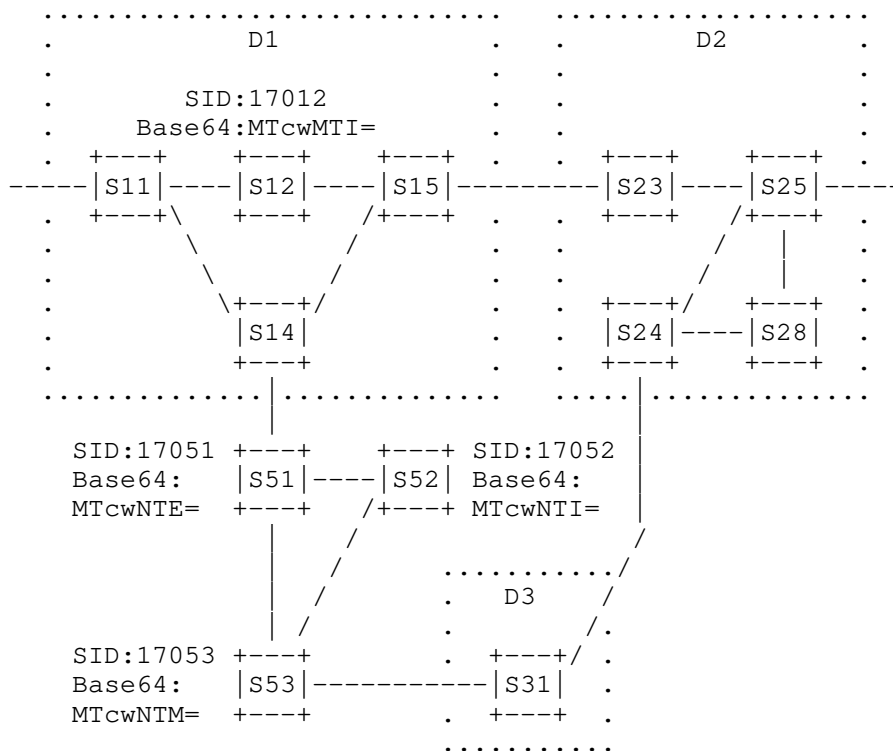


Figure 5: Example Underlay SR MPLS TE Topology

```

{
  "ietf-network:networks": {
    "network": [
      {
        "network-types": {
          "ietf-l3-unicast-topology:l3-unicast-topology": {
            "ietf-sr-mpls-topology:sr-mpls": {}
          },
          "ietf-te-topology:te-topology": {
            "ietf-te-topology-packet:packet": {}
          }
        },
        "network-id": "sr-mpls-te-topo-example",
        "ietf-l3-unicast-topology:l3-topology-attributes": {
          "ietf-sr-mpls-topology:sr-mpls": {
            "srgb": [
              {
                "lower-bound": 16000,

```

```

        "upper-bound": 23999
      }
    ]
  },
  "ietf-te-topology:te-topology-identifier": {
    "provider-id": 0,
    "client-id": 0,
    "topology-id": "sr-mpls-te-topo-example"
  },
  "ietf-te-topology:te": {
    "optimization-criterion": "ietf-te-types:of-minimize-cost-path"
  },
  "node": [
    {
      "node-id": "D1",
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "1-0-1",
          "ietf-l3-unicast-topology:l3-termination-point-attributes": {
            "unnumbered-id": 101
          },
          "ietf-te-topology:te-tp-id": 101,
          "ietf-te-topology:te": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-psc1",
                "encoding": "lsp-encoding-packet",
                "max-lsp-bandwidth": [
                  {
                    "priority": 7,
                    "te-bandwidth": {
                      "generic": "0x1p22"
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  ],
  {
    "tp-id": "1-2-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
      "unnumbered-id": 121
    },
    "ietf-te-topology:te-tp-id": 121,
    "ietf-te-topology:te": {
      "interface-switching-capability": [

```

```

        {
            "switching-capability": "switching-psc1",
            "encoding": "lsp-encoding-packet",
            "max-lsp-bandwidth": [
                {
                    "priority": 7,
                    "te-bandwidth": {
                        "generic": "0x1p22"
                    }
                }
            ]
        }
    ]
}
},
{
    "tp-id": "1-3-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 131
    },
    "ietf-te-topology:te-tp-id": 131,
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "switching-capability": "switching-psc1",
                "encoding": "lsp-encoding-packet",
                "max-lsp-bandwidth": [
                    {
                        "priority": 7,
                        "te-bandwidth": {
                            "generic": "0x1p22"
                        }
                    }
                ]
            }
        ]
    }
}
],
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "router-id": ["203.0.113.1"],
        "prefix": [
            {
                "prefix": "203.0.113.1/32",
                "ietf-sr-mpls-topology:sr-mpls": {
                    "sids": {
                        "sid": [
                            {

```

```
"algorithm": "prefix-sid-algorithm-shortest-path",
    "start-sid": 101,
    "range": 1,
    "is-local": false,
    "is-node": true
  }
]
}
}
],
"ietf-sr-mpls-topology:sr-mpls": {
  "srgb": [
    {
      "lower-bound": 16000,
      "upper-bound": 23999
    }
  ],
  "srlb": [
    {
      "lower-bound": 15000,
      "upper-bound": 15999
    }
  ]
},
"ietf-te-topology:te-node-id": "203.0.113.1",
"ietf-te-topology:te": {
  "te-node-attributes": {
    "admin-status": "up",
    "domain-id": 1001,
    "is-abstract": [null],
    "signaling-address": [
      "203.0.113.1"
    ],
  },
  "connectivity-matrices": {
    "is-allowed": true,
    "path-constraints": {
      "te-bandwidth": {
        "generic": "0x1p20"
      },
    },
    "path-metric-bounds": {
      "path-metric-bound": [
        {
          "metric-type": "path-metric-delay-average",
          "upper-bound": 15000
        }
      ]
    }
  ]
}
```



```
    }
  },
  "ietf-te-topology-packet:throttle": {
    "threshold-out": {
      "two-way-delay": 18000
    }
  },
  "connectivity-matrix": [
    {
      "id": 1,
      "from": {
        "tp-ref": "1-0-1"
      },
      "to": {
        "tp-ref": "1-2-1"
      },
      "is-allowed": true,
      "underlay": {
        "enabled": true,
        "primary-path": {
          "network-ref": "underlay-example",
          "path-element": [
            {
              "path-element-id": 1,
              "label-hop": {
                "te-label": {
                  "generic": "MTcwMTI=",
                  "direction": "forward"
                }
              }
            }
          ]
        }
      }
    }
  ],
  {
    "id": 2,
    "from": {
      "tp-ref": "1-2-1"
    },
    "to": {
      "tp-ref": "1-0-1"
    },
    "is-allowed": true,
    "underlay": {
      "enabled": true,
      "primary-path": {
        "network-ref": "underlay-example",
```

```

        "path-element": [
            {
                "path-element-id": 1,
                "label-hop": {
                    "te-label": {
                        "generic": "MTcwMTI=",
                        "direction": "forward"
                    }
                }
            }
        ]
    }
}
},
{
    "node-id": "D2",
    "ietf-network-topology:termination-point": [
        {
            "tp-id": "2-0-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id": 201
            },
            "ietf-te-topology:te-tp-id": 201,
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "switching-capability": "switching-pscl",
                        "encoding": "lsp-encoding-packet",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 7,
                                "te-bandwidth": {
                                    "generic": "0x1p22"
                                }
                            }
                        ]
                    }
                ]
            }
        }
    ]
},
{
    "tp-id": "2-1-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {

```

```

        "unnumbered-id": 211
    },
    "ietf-te-topology:te-tp-id": 211,
    "ietf-te-topology:te": {
        "interface-switching-capability": [
            {
                "switching-capability": "switching-pscl",
                "encoding": "lsp-encoding-packet",
                "max-lsp-bandwidth": [
                    {
                        "priority": 7,
                        "te-bandwidth": {
                            "generic": "0x1p22"
                        }
                    }
                ]
            }
        ]
    }
},
{
    "tp-id": "2-3-1",
    "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 231
    }
},
{
    "ietf-l3-unicast-topology:l3-node-attributes": {
        "router-id": ["203.0.113.2"],
        "prefix": [
            {
                "prefix": "203.0.113.2/32",
                "ietf-sr-mpls-topology:sr-mpls": {
                    "sids": {
                        "sid": [
                            {
                                "algorithm": "prefix-sid-algorithm-shortest-path",
                                "start-sid": 102,
                                "range": 1,
                                "is-local": false,
                                "is-node": true
                            }
                        ]
                    }
                }
            }
        ]
    }
},
{
    "ietf-sr-mpls-topology:sr-mpls": {

```

```

        "srgb": [
            {
                "lower-bound": 16000,
                "upper-bound": 23999
            }
        ],
        "srlb": [
            {
                "lower-bound": 15000,
                "upper-bound": 15999
            }
        ]
    }
},
"ietf-te-topology:te-node-id": "203.0.113.2",
"ietf-te-topology:te": {
    "te-node-attributes": {
        "admin-status": "up",
        "domain-id": 1001,
        "is-abstract": [null],
        "signaling-address": [
            "203.0.113.2"
        ]
    }
}
},
{
    "node-id": "D3",
    "ietf-network-topology:termination-point": [
        {
            "tp-id": "3-1-1",
            "ietf-l3-unicast-topology:l3-termination-point-attributes": {
                "unnumbered-id": 311
            },
            "ietf-te-topology:te-tp-id": 311,
            "ietf-te-topology:te": {
                "interface-switching-capability": [
                    {
                        "switching-capability": "switching-psc1",
                        "encoding": "lsp-encoding-packet",
                        "max-lsp-bandwidth": [
                            {
                                "priority": 7,
                                "te-bandwidth": {
                                    "generic": "0x1p22"
                                }
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        }
      ]
    },
    {
      "tp-id": "3-2-1",
      "ietf-l3-unicast-topology:l3-termination-point-attributes": {
        "unnumbered-id": 321
      }
    }
  ],
  "ietf-l3-unicast-topology:l3-node-attributes": {
    "router-id": ["203.0.113.3"],
    "prefix": [
      {
        "prefix": "203.0.113.3/32",
        "ietf-sr-mpls-topology:sr-mpls": {
          "sids": {
            "sid": [
              {
                "algorithm": "prefix-sid-algorithm-shortest-path",
                "start-sid": 103,
                "range": 1,
                "is-local": false,
                "is-node": true
              }
            ]
          }
        }
      }
    ]
  },
  "ietf-sr-mpls-topology:sr-mpls": {
    "srgb": [
      {
        "lower-bound": 16000,
        "upper-bound": 23999
      }
    ],
    "srlb": [
      {
        "lower-bound": 15000,
        "upper-bound": 15999
      }
    ]
  }
},
"ietf-te-topology:te-node-id": "203.0.113.3",
"ietf-te-topology:te": {

```

```
        "te-node-attributes": {
          "admin-status": "up",
          "domain-id": 1001,
          "signaling-address": [
            "203.0.113.3"
          ]
        }
      }
    ],
    "ietf-network-topology:link": [
      {
        "link-id": "D1,1-2-1,D2,2-1-1",
        "source": {
          "source-node": "D1",
          "source-tp": "1-2-1"
        },
        "destination": {
          "dest-node": "D2",
          "dest-tp": "2-1-1"
        },
        "ietf-l3-unicast-topology:l3-link-attributes": {
          "metric1": "100",
          "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {
              "sid": [
                {
                  "value-type": "index",
                  "sid": 121,
                  "is-local": true
                }
              ]
            }
          }
        },
        "ietf-te-topology:te": {
          "te-link-attributes": {
            "interface-switching-capability": [
              {
                "switching-capability": "switching-psc1",
                "encoding": "lsp-encoding-packet",
                "ietf-te-topology-packet:packet-switch-capable": {
                  "minimum-lsp-bandwidth": "0x1p20"
                }
              }
            ]
          }
        }
      }
    ]
  }
}
```

```

    }
  },
  {
    "link-id": "D2,2-1-1,D1,1-2-1",
    "source": {
      "source-node": "D2",
      "source-tp": "2-1-1"
    },
    "destination": {
      "dest-node": "D1",
      "dest-tp": "1-2-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
      "metric1": "100",
      "ietf-sr-mpls-topology:sr-mpls": {
        "sids": {
          "sid": [
            {
              "value-type": "index",
              "sid": 211,
              "is-local": true
            }
          ]
        }
      }
    },
    "ietf-te-topology:te": {
      "te-link-attributes": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-psc1",
            "encoding": "lsp-encoding-packet",
            "ietf-te-topology-packet:packet-switch-capable": {
              "minimum-lsp-bandwidth": "0x1p20"
            }
          }
        ]
      }
    }
  },
  {
    "link-id": "D1,1-3-1,D3,3-1-1",
    "source": {
      "source-node": "D1",
      "source-tp": "1-3-1"
    },
    "destination": {

```

```
    "dest-node": "D3",
    "dest-tp": "3-1-1"
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric1": "100",
    "ietf-sr-mpls-topology:sr-mpls": {
      "sids": {
        "sid": [
          {
            "value-type": "index",
            "sid": 131,
            "is-local": true
          }
        ]
      }
    }
  },
  "ietf-te-topology:te": {
    "te-link-attributes": {
      "is-abstract": [null],
      "underlay": {
        "enabled": true,
        "primary-path": {
          "network-ref": "underlay-example",
          "path-element": [
            {
              "path-element-id": 1,
              "label-hop": {
                "te-label": {
                  "generic": "MTcwNTE=",
                  "direction": "forward"
                }
              }
            },
            {
              "path-element-id": 2,
              "label-hop": {
                "te-label": {
                  "generic": "MTcwNTI=",
                  "direction": "forward"
                }
              }
            },
            {
              "path-element-id": 3,
              "label-hop": {
                "te-label": {
                  "generic": "MTcwNTM=",
```



```

        "direction": "forward"
      }
    }
  ]
},
"interface-switching-capability": [
  {
    "switching-capability": "switching-psc1",
    "encoding": "lsp-encoding-packet",
    "ietf-te-topology-packet:packet-switch-capable": {
      "minimum-lsp-bandwidth": "0x1p20"
    }
  }
]
},
{
  "link-id": "D3,3-1-1,D1,1-3-1",
  "source": {
    "source-node": "D3",
    "source-tp": "3-1-1"
  },
  "destination": {
    "dest-node": "D1",
    "dest-tp": "1-3-1"
  },
  "ietf-l3-unicast-topology:l3-link-attributes": {
    "metric1": "100",
    "ietf-sr-mpls-topology:sr-mpls": {
      "sids": {
        "sid": [
          {
            "value-type": "index",
            "sid": 311,
            "is-local": true
          }
        ]
      }
    }
  },
  "ietf-te-topology:te": {
    "te-link-attributes": {
      "is-abstract": [null],
      "interface-switching-capability": [

```

```

        {
            "switching-capability": "switching-psc1",
            "encoding": "lsp-encoding-packet",
            "ietf-te-topology-packet:packet-switch-capable":
            {
                "minimum-lsp-bandwidth": "0x1p20"
            }
        }
    ]
}
},
{
    "link-id": "D2,2-3-1,D3,3-2-1",
    "source": {
        "source-node": "D2",
        "source-tp": "2-3-1"
    },
    "destination": {
        "dest-node": "D3",
        "dest-tp": "3-2-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {
        "metric1": "100",
        "ietf-sr-mpls-topology:sr-mpls": {
            "sids": {
                "sid": [
                    {
                        "value-type": "index",
                        "sid": 231,
                        "is-local": true
                    }
                ]
            }
        }
    }
},
{
    "link-id": "D3,3-2-1,D2,2-3-1",
    "source": {
        "source-node": "D3",
        "source-tp": "3-2-1"
    },
    "destination": {
        "dest-node": "D2",
        "dest-tp": "2-3-1"
    },
    "ietf-l3-unicast-topology:l3-link-attributes": {

```

```
    "metric1": "100",
    "ietf-sr-mpls-topology:sr-mpls": {
      "sids": {
        "sid": [
          {
            "value-type": "index",
            "sid": 321,
            "is-local": true
          }
        ]
      }
    }
  }
}
```

Appendix C. Contributors

Jeff Tantsura
Email: jefftant.ietf@gmail.com

Yingzhen Qu
Email: yingzhen.qu@huawei.com

Authors' Addresses

Xufeng Liu
IBM Corporation

EMail: xufeng.liu.ietf@gmail.com

Igor Bryskin
Individual

EMail: i_bryskin@yahoo.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

Stephane Litkowski
Cisco

EMail: slitkows.ietf@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 11 August 2022

T. Saad
Juniper Networks
R. Gandhi
Cisco Systems Inc
X. Liu
Volta Networks
V.P. Beeram
Juniper Networks
I. Bryskin
Individual
O. Gonzalez de Dios
Telefonica
7 February 2022

A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths
and Interfaces
draft-ietf-teas-yang-te-29

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 is divided into YANG modules that classify data into generic, device-specific, technology agnostic, and technology-specific elements.

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 11 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	3
2.1. Prefixes in Data Node Names	4
2.2. Model Tree Diagrams	4
3. Design Considerations	5
3.1. State Data Organization	5
4. Model Overview	6
4.1. Module Relationship	6
5. TE YANG Model	7
5.1. Module Structure	7
5.1.1. TE Globals	9
5.1.2. TE Tunnels	12
5.1.3. TE LSPs	19
5.2. Tree Diagram	19
5.3. YANG Module	60
6. TE Device YANG Model	98
6.1. Module Structure	99
6.1.1. TE Interfaces	99
6.2. Tree Diagram	100
6.3. YANG Module	102
7. Notifications	116
8. TE Generic and Helper YANG Modules	117
9. IANA Considerations	117
10. Security Considerations	117
11. Acknowledgement	119
12. Contributors	119
13. Appendix A: Data Tree Examples	119
13.1. Basic Tunnel Setup	120
13.2. Global Named Path Constraints	121
13.3. Tunnel with Global Path Constraint	121
13.4. Tunnel with Per-tunnel Path Constraint	122
13.5. Tunnel State	123

14. References	124
14.1. Normative References	124
14.2. Informative References	127
Authors' Addresses	128

1. Introduction

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

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

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

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

2. Requirements Language

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

The 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

2.1. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
te-types	ietf-te-types	[RFC8776]
te-packet-types	ietf-te-packet-types	[RFC8776]
te	ietf-te	this document
te-dev	ietf-te-device	this document

Table 1: Prefixes and corresponding YANG modules

2.2. Model Tree Diagrams

The tree diagrams extracted from the module(s) defined in this document are given in subsequent sections as per the syntax defined in [RFC8340].

3. Design Considerations

This document describes a generic TE YANG data model that is independent of any dataplane technology. One of the design objectives is to allow specific data plane technology models to reuse the TE generic data model and possibly augment it with technology specific data.

The elements of the generic TE YANG data model, including TE Tunnels, LSPs, and interfaces have leaf(s) that identify the technology layer where they reside. For example, the LSP encoding type can identify the technology associated with a TE Tunnel or LSP.

Also, the generic TE YANG data model does not cover signaling protocol data. The signaling protocol used to instantiate TE LSPs are outside the scope of this document and expected to be covered by augmentations defined in other document(s).

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

- * The generic TE YANG data model 'ietf-te' contains device independent data and can be used to model data off a device (e.g. on a TE controller). The device-specific TE data is defined in module 'ietf-te-device' as shown in Figure 1,
- * 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 be in either augmentations, or deviations to the model defined in this document.

4.1. Module Relationship

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

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

The TE data model for specific instances of signaling protocol are outside the scope of this document and are defined in other documents. For example, the RSVP-TE YANG model augmentation of the TE model is covered in [I-D.ietf-teas-yang-rsvp].

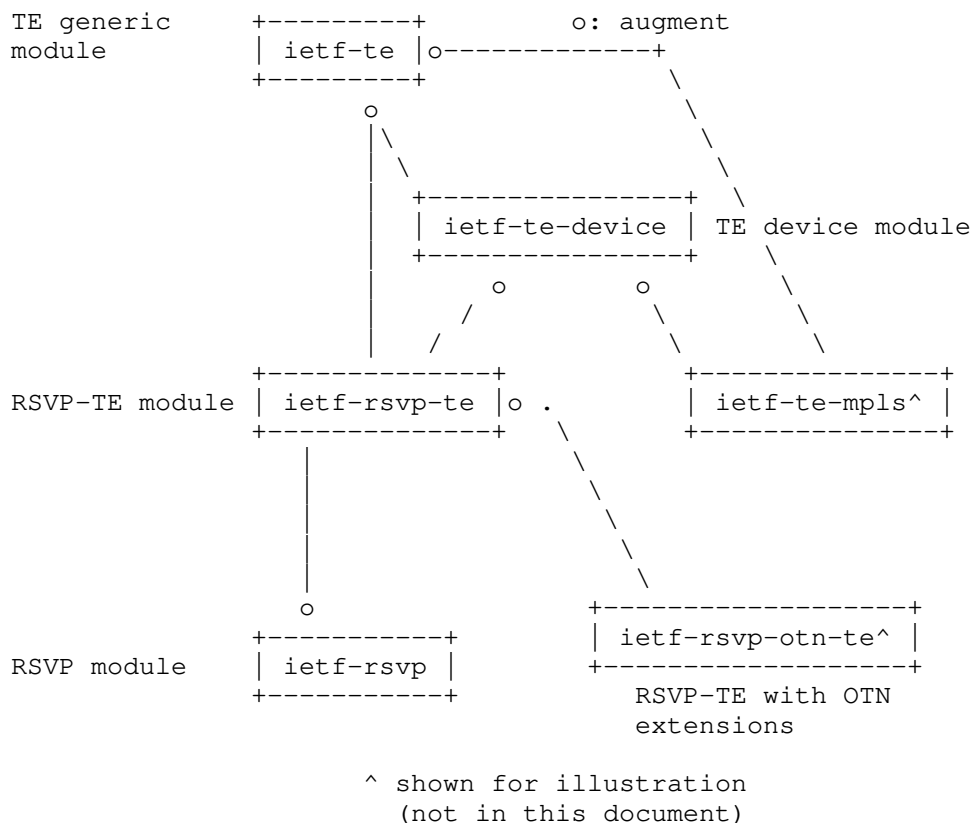


Figure 1: Relationship of TE module(s) with signaling protocol modules

5. TE YANG Model

The generic TE YANG module ('ietf-te') is meant to manage and operate a TE network. This includes creating, modifying and retrieving TE Tunnels, LSPs, and interfaces and their associated attributes (e.g. Administrative-Groups, SRLGs, etc.).

The detailed tree structure is provided in Figure 2.

5.1. Module Structure

The 'ietf-te' uses three main containers grouped under the main 'te' container (see Figure 2). The 'te' container is the top level container in the data model. The presence of the 'te' container enables TE function system wide. Below provides further descriptions of containers that exist under the 'te' top level container.

globals:

The 'globals' container maintains the set of global TE attributes that can be applicable to TE Tunnel(s) and interface(s).

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.

tunnels-path-compute:

A Remote Procedure Call (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.

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

```
rpcs:
  +---x tunnels-path-compute
  +---x tunnels-action
```

Figure 2: TE Tunnel model high-level YANG tree view

5.1.1. TE Globals

The 'globals' container covers properties that control TE features behavior system-wide, and its respective state (see Figure 3). The TE globals configuration include:

```

+--rw globals
|   +--rw named-admin-groups
|   |   +--rw named-admin-group* [name]
|   ..
|   +--rw named-srlgs
|   |   +--rw named-srlg* [name] {te-types:named-srlg-groups}?
|   ..
|   +--rw named-path-constraints
|   |   +--rw named-path-constraint* [name]
|   ..

```

Figure 3: TE globals 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 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 constraint 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 up of the following path constraints:

```

|   +--rw named-path-constraints
|       +--rw named-path-constraint* [name]
|           +--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?
|               |
|               +--rw te-path-disjointness
|       // ...
|           +--rw explicit-route-objects-always
|       // ...
|               |
|               +--rw route-object-exclude-always* [index]
|               |
|               +--rw route-object-include-exclude* [index]

```

Figure 4: Named path constraints YANG subtree

- o te-bandwidth: A YANG container that holds the technology agnostic TE bandwidth constraint.
- o link-protection: A YANG leaf that holds the link protection type constraint required for the links to be included in the computed path.
- o setup/hold priority: A YANG leaf that holds the LSP setup and hold admission priority as defined in [RFC3209].
- o signaling-type: A YANG leaf that holds the LSP setup type, such as RSVP-TE or SR.
- o path-metric-bounds: A YANG container that holds the set of metric bounds applicable on the computed TE tunnel path.

- o `path-affinities-values`: A YANG container that holds the set of affinity values and mask to be used during path computation.
- o `path-affinity-names`: A YANG container that holds the set of named affinity constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `path-srlgs-lists`: A YANG container that holds the set of SRLG values and corresponding inclusion or exclusions instruction to be used during path computation.
- o `path-srlgs-names`: A YANG container that holds the set of named SRLG constraints and corresponding inclusion or exclusions instruction for each to be used during path computation.
- o `disjointness`: The level of resource disjointness constraint that the secondary path of a TE tunnel has to adhere to.
- o `explicit-route-objects-always`: A YANG container that contains two route objects lists:
 - + `'route-object-exclude-always'`: a list of route entries to always exclude from the path computation.
 - + `'route-object-include-exclude'`: a list of route entries to include or exclude in the path computation.

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.

5.1.2. TE Tunnels

The 'tunnels' container holds the list of TE Tunnels that are provisioned on devices in the network (see Figure 5).

A TE Tunnel in the list is uniquely identified by a name. When the model is used to manage a specific device, the 'tunnels' list contains the TE Tunnels originating from the specific device. When the model is used to manage a TE controller, the 'tunnels' list contains all TE Tunnels and TE tunnel segments originating from device(s) that the TE controller manages.

The TE Tunnel model allows the configuration and management of the following TE tunnel related objects:

TE Tunnel:

A YANG container of one or more LSPs established between the source and destination TE Tunnel termination points. A TE Tunnel LSP is a connection-oriented service provided by the network layer for the delivery of client data between a source and the destination of the TE Tunnel termination points.

TE Tunnel Segment:

A part of a multi-domain TE Tunnel that is within a specific network domain.


```

+--rw tunnels
|   +--rw tunnel* [name]
|   |   +--rw name                               string
|   |   +--rw alias?                             string
|   |   +--rw identifier?                         uint32
|   |   +--rw color?                             uint32
|   |   +--rw description?                       string
|   |   +--ro operational-state?                 identityref
|   |   +--rw encoding?                         identityref
|   |   +--rw switching-type?                   identityref
|   |   +--rw admin-state?                     identityref
|   |   +--rw reoptimize-timer?                 uint16
|   |   +--rw source?                           te-types:te-node-id
|   |   +--rw destination?                     te-types:te-node-id
|   |   +--rw src-tunnel-tp-id?                 binary
|   |   +--rw dst-tunnel-tp-id?                 binary
|   |   +--rw controller
|   |   |   +--rw protocol-origin?               identityref
|   |   |   +--rw controller-entity-id?         string
|   |   +--rw bidirectional?                   boolean
|   |   +--rw association-objects
|   |   |   +--rw association-object* [association-key]
|   |
|   |   // ..
|   |   |
|   |   +--rw protection
|   |
|   |   // ..
|   |   |
|   |   +--rw restoration
|   |
|   |   // ..
|   |   |
|   |   +--rw te-topology-identifier
|   |
|   |   // ..
|   |   |
|   |   +--rw hierarchy
|   |
|   |   // ..

```

Figure 5: TE Tunnel list YANG subtree structure

The TE Tunnel has a number of attributes that are set directly under the tunnel (see 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.

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].

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].

reoptimize-timer:

A YANG leaf to set the interval period for tunnel reoptimization.

source/destination:

YANG leafs that define the tunnel source and destination node endpoints.

src-tunnel-tp-id/dst-tunnel-tp-id:

YANG leafs that hold the identifiers of source and destination TE Tunnel Termination Points (TTPs) [RFC8795] residing on the source and destination nodes. 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.

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.

bidirectional:

A YANG leaf that when present indicates the LSPs of a TE Tunnel are bidirectional and co-routed.

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.

```

+--rw hierarchy
|   +--rw dependency-tunnels
|   |   +--rw dependency-tunnel* [name]
|   |   |   +--rw name
|   |   |   |   -> ../../../../tunnels/tunnel/name
|   |   |   +--rw encoding?          identityref
|   |   |   +--rw switching-type?    identityref
|   |   +--rw hierarchical-link
|   |   |   +--rw local-te-node-id?    te-types:te-node-id
|   |   |   +--rw local-te-link-tp-id? te-types:te-tp-id
|   |   |   +--rw remote-te-node-id?   te-types:te-node-id
|   |   +--rw te-topology-identifier
|   |   |   +--rw provider-id?    te-global-id
|   |   |   +--rw client-id?     te-global-id
|   |   |   +--rw topology-id?   te-topology-id

```

Figure 6: TE Tunnel hierarchy YANG subtree

hierarchy:

A YANG container that holds hierarchy related properties of the TE Tunnel (see Figure 6. A TE LSP can be set up in MPLS or Generalized MPLS (GMPLS) networks to be used as a TE links to carry traffic in other (client) networks [RFC6107]. In this case, the model introduces the TE Tunnel hierarchical link endpoint parameters to identify the specific link in the client layer that the underlying TE Tunnel is associated with. The hierarchy container includes the following:

- 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.

5.1.2.1. TE Tunnel Paths

The TE Tunnel can be configured with a set of paths that define the tunnel forward and reverse paths as described in Figure 7. Moreover, a primary path can be specified a set of candidate secondary paths that can be visited to support path protection. The following describe further the list of paths associated with a TE Tunnel.

```

|      +--rw primary-paths
|      |   +--rw primary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      |   +
|      |   +--rw primary-reverse-path
|      |   |   +--rw name?
|      |   |   |   string
|  // ..
|      |   |
|      |   |   +--rw candidate-secondary-reverse-paths
|      |   |   |   +--rw candidate-secondary-reverse-path*
|      |   |   |   |   [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   +--rw candidate-secondary-paths
|      |   |   |   +--rw candidate-secondary-path* [secondary-path]
|      |   |   |   |   +--rw secondary-path
|      |   |   |   |   |   leafref
|      |   |   |   |   +--ro active?
|      |   |   |   |   |   boolean
|
|      +--rw secondary-paths
|      |   +--rw secondary-path* [name]
|      |   |   +--rw name
|      |   |   |   string
|  // ..
|      +--rw secondary-reverse-paths
|      |   +--rw secondary-reverse-path* [name]
|      |   |   +--rw name
|      |   |   |   string

```

Figure 7: TE Tunnel paths YANG tree structure

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. A secondary path contains attributes similar to a primary path.

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 common path attributes are shared for primary forward and reverse primary and secondary paths:

compute-only:

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

te-topology-identifier:

A YANG container that holds the topology identifier associated with the tunnel.

optimizations:

a YANG container that holds the optimization objectives that path computation will use to select a path.

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.

lsps:

a YANG container that holds a list of LSPs that are instantiated for this specific path.

5.1.3. TE LSPs

The 'lsps' container includes the set of TE LSP(s) that are instantiated. A TE LSP is identified by a 3-tuple ('tunnel-name', 'node', 'lsp-id').

When the model is used to manage a specific device, the 'lsps' list contains all TE LSP(s) that traverse the device (including ingressing, transiting and egressing the device).

When the model is used to manage a TE controller, the 'lsps' list contains all TE LSP(s) that traverse all network devices (including ingressing, transiting and egressing the device) that the TE controller manages.

5.2. Tree Diagram

Figure 8 shows the tree diagram of the generic TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te
+--rw te!
  +--rw globals
    +--rw named-admin-groups
      +--rw named-admin-group* [name]
        {te-types:extended-admin-groups,te-types:named-extend
ed-admin-groups}?
        +--rw name string
        +--rw bit-position? uint32
    +--rw named-srlgs
      +--rw named-srlg* [name] {te-types:named-srlg-groups}?
        +--rw name string
        +--rw value? te-types:srlg
        +--rw cost? uint32
    +--rw named-path-constraints
      +--rw named-path-constraint* [name]
        {te-types:named-path-constraints}?
        +--rw name string
        +--rw te-bandwidth
          +--rw (technology)?
            +--:(generic)
              +--rw generic? te-bandwidth
        +--rw link-protection? identityref
        +--rw setup-priority? uint8
        +--rw hold-priority? uint8
        +--rw signaling-type? identityref
        +--rw path-metric-bounds
          +--rw path-metric-bound* [metric-type]
            +--rw metric-type identityref
            +--rw upper-bound? uint64
        +--rw path-affinities-values
          +--rw path-affinities-value* [usage]
            +--rw usage identityref
            +--rw value? admin-groups
        +--rw path-affinity-names
          +--rw path-affinity-name* [usage]
            +--rw usage identityref
            +--rw affinity-name* [name]
              +--rw name string
        +--rw path-srlgs-lists
          +--rw path-srlgs-list* [usage]
            +--rw usage identityref
            +--rw values* srlg
        +--rw path-srlgs-names
          +--rw path-srlgs-name* [usage]
            +--rw usage identityref
            +--rw names* string
        +--rw disjointness?

```



```

    te-path-disjointness
+--rw explicit-route-objects-always
+--rw route-object-exclude-always* [index]
+--rw index                               uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw hop-type?        te-hop-type
+--rw direction?       te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw node-id         te-node-id
+--rw hop-type?        te-hop-type
+--rw direction?       te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number        inet:as-number
+--rw hop-type?        te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
+--:(generic)
+--rw generic?
+--rw rt-types:generalized-label
+--rw direction?
+--rw te-label-direction
+--rw route-object-include-exclude* [index]
+--rw explicit-route-usage?            identityref
+--rw index                             uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id         te-node-id
+--rw hop-type?       te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id       te-tp-id
+--rw hop-type?        te-hop-type
+--rw direction?       te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop

```

```

    +---rw link-tp-id      te-tp-id
    +---rw node-id         te-node-id
    +---rw hop-type?       te-hop-type
    +---rw direction?      te-link-direction
+---:(as-number)
  +---rw as-number-hop
  +---rw as-number        inet:as-number
  +---rw hop-type?        te-hop-type
+---:(label)
  +---rw label-hop
  +---rw te-label
  +---rw (technology)?
  |   +---:(generic)
  |       +---rw generic?
  |           rt-types:generalized-label
  +---rw direction?
  |       te-label-direction
+---:(srlg)
  +---rw srlg
  +---rw srlg?            uint32
+---rw path-in-segment!
  +---rw label-restrictions
  +---rw label-restriction* [index]
  +---rw restriction?      enumeration
  +---rw index             uint32
  +---rw label-start
  |   +---rw te-label
  |       +---rw (technology)?
  |           +---:(generic)
  |               +---rw generic?
  |                   rt-types:generalized-label
  +---rw direction?
  |       te-label-direction
+---rw label-end
  +---rw te-label
  +---rw (technology)?
  |   +---:(generic)
  |       +---rw generic?
  |           rt-types:generalized-label
  +---rw direction?
  |       te-label-direction
+---rw label-step
  |   +---rw (technology)?
  |       +---:(generic)
  |           +---rw generic?      int32
  +---rw range-bitmap?      yang:hex-string
+---rw path-out-segment!
  +---rw label-restrictions

```

```

        +---rw label-restriction* [index]
            +---rw restriction?    enumeration
            +---rw index           uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?    int32
            +---rw range-bitmap?    yang:hex-string
+---rw tunnels
    +---rw tunnel* [name]
        +---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?             te-types:te-node-id
        +---rw destination?        te-types:te-node-id
        +---rw src-tunnel-tp-id?   binary
        +---rw dst-tunnel-tp-id?   binary
        +---rw bidirectional?      boolean
        +---rw controller
            +---rw protocol-origin? identityref
            +---rw controller-entity-id? string
        +---rw reoptimize-timer?   uint16
        +---rw association-objects
            +---rw association-object* [association-key]
                +---rw association-key    string
                +---rw type?              identityref

```

```

+--rw id?                               uint16
+--rw source
  +--rw id?      te-gen-node-id
  +--rw type?    enumeration
+--rw association-object-extended* [association-key]
  +--rw association-key    string
  +--rw type?              identityref
  +--rw id?                uint16
  +--rw source
    +--rw id?      te-gen-node-id
    +--rw type?    enumeration
  +--rw global-source?    uint32
  +--rw extended-id?      yang:hex-string
+--rw protection
  +--rw enable?                        boolean
  +--rw protection-type?               identityref
  +--rw protection-reversion-disable?  boolean
  +--rw hold-off-time?                 uint32
  +--rw wait-to-revert?                uint16
  +--rw aps-signal-id?                 uint8
+--rw restoration
  +--rw enable?                        boolean
  +--rw restoration-type?               identityref
  +--rw restoration-scheme?             identityref
  +--rw restoration-reversion-disable?  boolean
  +--rw hold-off-time?                 uint32
  +--rw wait-to-restore?                uint16
  +--rw wait-to-revert?                uint16
+--rw te-topology-identifier
  +--rw provider-id?    te-global-id
  +--rw client-id?      te-global-id
  +--rw topology-id?    te-topology-id
+--rw te-bandwidth
  +--rw (technology)?
    +--:(generic)
      +--rw generic?    te-bandwidth
+--rw link-protection?            identityref
+--rw setup-priority?             uint8
+--rw hold-priority?              uint8
+--rw signaling-type?             identityref
+--rw hierarchy
  +--rw dependency-tunnels
    +--rw dependency-tunnel* [name]
      +--rw name
        -> /te/tunnels/tunnel/name
      +--rw encoding?            identityref
      +--rw switching-type?      identityref
+--rw hierarchical-link

```

```

+--rw local-te-node-id?          te-types:te-node-id
+--rw local-te-link-tp-id?       te-types:te-tp-id
+--rw remote-te-node-id?         te-types:te-node-id
+--rw te-topology-identifier
  +--rw provider-id?             te-global-id
  +--rw client-id?               te-global-id
  +--rw topology-id?             te-topology-id
+--rw primary-paths
  +--rw primary-path* [name]
    +--rw name                    string
    +--rw path-computation-method? identityref
    +--rw path-computation-server
      +--rw id?                   te-gen-node-id
      +--rw type?                 enumeration
    +--rw compute-only?           empty
    +--rw use-path-computation?   boolean
    +--rw lockdown?               empty
    +--rw path-scope?             identityref
    +--rw preference?             uint8
    +--rw k-requested-paths?      uint8
    +--rw association-objects
      +--rw association-object* [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
      +--rw association-object-extended*
        [association-key]
        +--rw association-key     string
        +--rw type?               identityref
        +--rw id?                 uint16
        +--rw source
          +--rw id?               te-gen-node-id
          +--rw type?             enumeration
        +--rw global-source?      uint32
        +--rw extended-id?        yang:hex-string
+--rw optimizations
  +--rw (algorithm)?
    +--:(metric) {path-optimization-metric}?
      +--rw optimization-metric* [metric-type]
        +--rw metric-type
          identityref
        +--rw weight?
          uint8
        +--rw explicit-route-exclude-objects
          +--rw route-object-exclude-object*

```

```

[index]
+--rw index
|
|   uint32
+--rw (type)?
+---:(numbered-node-hop)
|   +--rw numbered-node-hop
|   |   +--rw node-id
|   |   |   te-node-id
|   |   +--rw hop-type?
|   |       te-hop-type
+---:(numbered-link-hop)
|   +--rw numbered-link-hop
|   |   +--rw link-tp-id
|   |   |   te-tp-id
|   |   +--rw hop-type?
|   |   |   te-hop-type
|   |   +--rw direction?
|   |       te-link-direction
+---:(unnumbered-link-hop)
|   +--rw unnumbered-link-hop
|   |   +--rw link-tp-id
|   |   |   te-tp-id
|   |   +--rw node-id
|   |   |   te-node-id
|   |   +--rw hop-type?
|   |   |   te-hop-type
|   |   +--rw direction?
|   |       te-link-direction
+---:(as-number)
|   +--rw as-number-hop
|   |   +--rw as-number
|   |   |   inet:as-number
|   |   +--rw hop-type?
|   |       te-hop-type
+---:(label)
|   +--rw label-hop
|   |   +--rw te-label
|   |   |   +--rw (technology)?
|   |   |   |   +---:(generic)
|   |   |   |   |   +--rw generic?
|   |   |   |       rt-types:ge
|   |   |   |       +--rw direction?
|   |   |   |       |   te-label-directio
+---:(srlg)
|   +--rw srlg
|   |   +--rw srlg?   uint32

```

```

+---rw explicit-route-include-objects
+---rw route-object-include-object*
    [index]
+---rw index
    |   uint32
+---rw (type)?
+---:(numbered-node-hop)
    +---rw numbered-node-hop
        +---rw node-id
            |   te-node-id
        +---rw hop-type?
            |   te-hop-type
+---:(numbered-link-hop)
    +---rw numbered-link-hop
        +---rw link-tp-id
            |   te-tp-id
        +---rw hop-type?
            |   te-hop-type
        +---rw direction?
            |   te-link-direction
+---:(unnumbered-link-hop)
    +---rw unnumbered-link-hop
        +---rw link-tp-id
            |   te-tp-id
        +---rw node-id
            |   te-node-id
        +---rw hop-type?
            |   te-hop-type
        +---rw direction?
            |   te-link-direction
+---:(as-number)
    +---rw as-number-hop
        +---rw as-number
            |   inet:as-number
        +---rw hop-type?
            |   te-hop-type
+---:(label)
    +---rw label-hop
        +---rw te-label
            +---rw (technology)?
                +---:(generic)
                    +---rw generic?
                        |   rt-types:ge
neralized-label
n
+---rw tiebreakers

```

```

    +---rw tiebreaker* [tiebreaker-type]
    +---rw tiebreaker-type identityref
+---:(objective-function)
    {path-optimization-objective-function}?
    +---rw objective-function
    +---rw objective-function-type?
        identityref
+---rw named-path-constraint? leafref
    {te-types:named-path-constraints}?
+---rw te-bandwidth
    +---rw (technology)?
    +---:(generic)
        +---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
    +---rw path-metric-bound* [metric-type]
    +---rw metric-type identityref
    +---rw upper-bound? uint64
+---rw path-affinities-values
    +---rw path-affinities-value* [usage]
    +---rw usage identityref
    +---rw value? admin-groups
+---rw path-affinity-names
    +---rw path-affinity-name* [usage]
    +---rw usage identityref
    +---rw affinity-name* [name]
    +---rw name string
+---rw path-srlgs-lists
    +---rw path-srlgs-list* [usage]
    +---rw usage identityref
    +---rw values* srlg
+---rw path-srlgs-names
    +---rw path-srlgs-name* [usage]
    +---rw usage identityref
    +---rw names* string
+---rw disjointness?
    te-path-disjointness
+---rw explicit-route-objects-always
    +---rw route-object-exclude-always* [index]
    +---rw index uint32
    +---rw (type)?
    +---:(numbered-node-hop)
        +---rw numbered-node-hop
        +---rw node-id te-node-id
        +---rw hop-type? te-hop-type

```


bel

```

+---:(numbered-link-hop)
|   +---rw numbered-link-hop
|       +---rw link-tp-id      te-tp-id
|       +---rw hop-type?      te-hop-type
|       +---rw direction?     te-link-direction
+---:(unnumbered-link-hop)
|   +---rw unnumbered-link-hop
|       +---rw link-tp-id      te-tp-id
|       +---rw node-id        te-node-id
|       +---rw hop-type?      te-hop-type
|       +---rw direction?     te-link-direction
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number      inet:as-number
|       +---rw hop-type?     te-hop-type
+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?      identityref
+---rw index                      uint32
+---rw (type)?
+---:(numbered-node-hop)
|   +---rw numbered-node-hop
|       +---rw node-id      te-node-id
|       +---rw hop-type?    te-hop-type
+---:(numbered-link-hop)
|   +---rw numbered-link-hop
|       +---rw link-tp-id    te-tp-id
|       +---rw hop-type?     te-hop-type
|       +---rw direction?    te-link-direction
+---:(unnumbered-link-hop)
|   +---rw unnumbered-link-hop
|       +---rw link-tp-id    te-tp-id
|       +---rw node-id      te-node-id
|       +---rw hop-type?     te-hop-type
|       +---rw direction?    te-link-direction
+---:(as-number)
|   +---rw as-number-hop
|       +---rw as-number      inet:as-number
|       +---rw hop-type?     te-hop-type

```

bel

```

+---:(label)
|   +---rw label-hop
|       +---rw te-label
|           +---rw (technology)?
|               +---:(generic)
|                   +---rw generic?
|                       rt-types:generalized-la
|
|       +---rw direction?
|           te-label-direction
+---:(srlg)
|   +---rw srlg
|       +---rw srlg?    uint32
+---rw path-in-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?    enumeration
+---rw index            uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-end
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)
|               +---rw generic?
|                   rt-types:generalized-label
|       +---rw direction?
|           te-label-direction
+---rw label-step
|   +---rw (technology)?
|       +---:(generic)
|           +---rw generic?    int32
+---rw range-bitmap?    yang:hex-string
+---rw path-out-segment!
+---rw label-restrictions
+---rw label-restriction* [index]
+---rw restriction?    enumeration
+---rw index            uint32
+---rw label-start
|   +---rw te-label
|       +---rw (technology)?
|           +---:(generic)

```

```

|         |         |         +---rw generic?
|         |         |         rt-types:generalized-label
|         |         +---rw direction?
|         |         te-label-direction
+---rw label-end
|   +---rw te-label
|     +---rw (technology)?
|       +---:(generic)
|         +---rw generic?
|           rt-types:generalized-label
|     +---rw direction?
|       te-label-direction
+---rw label-step
|   +---rw (technology)?
|     +---:(generic)
|       +---rw generic?    int32
+---rw range-bitmap?      yang:hex-string
+---ro computed-paths-properties
+---ro computed-path-properties* [k-index]
+---ro k-index              uint8
+---ro path-properties
+---ro path-metric* [metric-type]
|   +---ro metric-type          identityref
|   +---ro accumulative-value?  uint64
+---ro path-affinities-values
|   +---ro path-affinities-value* [usage]
|     +---ro usage              identityref
|     +---ro value?            admin-groups
+---ro path-affinity-names
|   +---ro path-affinity-name* [usage]
|     +---ro usage              identityref
|     +---ro affinity-name* [name]
|       +---ro name             string
+---ro path-srlgs-lists
|   +---ro path-srlgs-list* [usage]
|     +---ro usage              identityref
|     +---ro values*           srlg
+---ro path-srlgs-names
|   +---ro path-srlgs-name* [usage]
|     +---ro usage              identityref
|     +---ro names*            string
+---ro path-route-objects
|   +---ro path-route-object* [index]
|     +---ro index
|       |                uint32
|     +---ro (type)?
|       +---:(numbered-node-hop)
|         +---ro numbered-node-hop

```

```

+--ro node-id      te-node-id
+--ro hop-type?
    te-hop-type
+--:(numbered-link-hop)
+--ro numbered-link-hop
+--ro link-tp-id    te-tp-id
+--ro hop-type?
    |
    te-hop-type
+--ro direction?
    te-link-direction
+--:(unnumbered-link-hop)
+--ro unnumbered-link-hop
+--ro link-tp-id    te-tp-id
+--ro node-id
    |
    te-node-id
+--ro hop-type?
    |
    te-hop-type
+--ro direction?
    te-link-direction
+--:(as-number)
+--ro as-number-hop
+--ro as-number
    |
    inet:as-number
+--ro hop-type?
    te-hop-type
+--:(label)
+--ro label-hop
+--ro te-label
    +--ro (technology)?
        |
        +--:(generic)
        |
        +--ro generic?
            rt-types:gener
+--ro direction?
    te-label-direction
+--ro te-bandwidth
+--ro (technology)?
    +--:(generic)
    +--ro generic?
        te-bandwidth
+--ro disjointness-type?
    te-types:te-path-disjointness
+--ro computed-path-error-infos
+--ro computed-path-error-info* []
+--ro error-description?
    string
+--ro error-timestamp?
    yang:date-and-time
+--ro error-reason?
    identityref
+--ro lsp-provisioning-error-infos
+--ro lsp-provisioning-error-info* []

```

```

    +--ro error-description?  string
    +--ro error-timestamp?    yang:date-and-time
    +--ro error-node-id?      te-types:te-node-id
    +--ro error-link-id?      te-types:te-tp-id
    +--ro lsp-id?             uint16
+--ro lsps
  +--ro lsp* [node lsp-id]
    +--ro tunnel-name?
      |      -> /te/lsps/lsp/tunnel-name
    +--ro node          -> /te/lsps/lsp/node
    +--ro lsp-id        -> /te/lsps/lsp/lsp-id
+--rw primary-reverse-path
  +--rw name?                                string
  +--rw path-computation-method?
    |      identityref
  +--rw path-computation-server
    |      +--rw id?      te-gen-node-id
    |      +--rw type?    enumeration
  +--rw compute-only?                        empty
  +--rw use-path-computation?
    |      boolean
  +--rw lockdown?                            empty
  +--ro path-scope?
    |      identityref
  +--rw association-objects
    +--rw association-object* [association-key]
      |      +--rw association-key  string
      |      +--rw type?            identityref
      |      +--rw id?              uint16
      |      +--rw source
      |        |      +--rw id?      te-gen-node-id
      |        |      +--rw type?    enumeration
    +--rw association-object-extended*
      |      [association-key]
      |      +--rw association-key  string
      |      +--rw type?            identityref
      |      +--rw id?              uint16
      |      +--rw source
      |        |      +--rw id?      te-gen-node-id
      |        |      +--rw type?    enumeration
      |      +--rw global-source?   uint32
      |      +--rw extended-id?     yang:hex-string
  +--rw optimizations
    +--rw (algorithm)?
      +--:(metric) {path-optimization-metric}?
        |      +--rw optimization-metric* [metric-type]
        |        |      +--rw metric-type
        |        |      |      identityref

```

```

+---rw weight?
|   uint8
+---rw explicit-route-exclude-objects
|   +---rw route-object-exclude-object*
|       [index]
|       +---rw index
|           |   uint32
|       +---rw (type)?
|           +---:(numbered-node-hop)
|               +---rw numbered-node-hop
|                   +---rw node-id
|                       |   te-node-id
|                   +---rw hop-type?
|                       |   te-hop-type
|           +---:(numbered-link-hop)
|               +---rw numbered-link-hop
|                   +---rw link-tp-id
|                       |   te-tp-id
|                   +---rw hop-type?
|                       |   te-hop-type
|                   +---rw direction?
|                       |   te-link-direction
|           +---:(unnumbered-link-hop)
|               +---rw unnumbered-link-hop
|                   +---rw link-tp-id
|                       |   te-tp-id
|                   +---rw node-id
|                       |   te-node-id
|                   +---rw hop-type?
|                       |   te-hop-type
|                   +---rw direction?
|                       |   te-link-direction
|           +---:(as-number)
|               +---rw as-number-hop
|                   +---rw as-number
|                       |   inet:as-number
|                   +---rw hop-type?
|                       |   te-hop-type
|           +---:(label)
|               +---rw label-hop
|                   +---rw te-label
|                       +---rw (technology)?
|                           +---:(generic)
|                               +---rw generic?
|                                   rt-types
:generalized-label
+---rw direction?
    te-label-direc

```



```

        +---rw direction?
            te-label-direct
tion
        +---rw tiebreakers
            +---rw tiebreaker* [tiebreaker-type]
                +---rw tiebreaker-type
                    identityref
        +---:(objective-function)
            {path-optimization-objective-function
}?:
        +---rw objective-function
            +---rw objective-function-type?
                identityref
        +---rw named-path-constraint? leafref
            {te-types:named-path-constraints}?
        +---rw te-bandwidth
            +---rw (technology)?
                +---:(generic)
                    +---rw generic? te-bandwidth
        +---rw link-protection?
            identityref
        +---rw setup-priority? uint8
        +---rw hold-priority? uint8
        +---rw signaling-type?
            identityref
        +---rw path-metric-bounds
            +---rw path-metric-bound* [metric-type]
                +---rw metric-type identityref
                +---rw upper-bound? uint64
        +---rw path-affinities-values
            +---rw path-affinities-value* [usage]
                +---rw usage identityref
                +---rw value? admin-groups
        +---rw path-affinity-names
            +---rw path-affinity-name* [usage]
                +---rw usage identityref
                +---rw affinity-name* [name]
                    +---rw name string
        +---rw path-srlgs-lists
            +---rw path-srlgs-list* [usage]
                +---rw usage identityref
                +---rw values* srlg
        +---rw path-srlgs-names
            +---rw path-srlgs-name* [usage]
                +---rw usage identityref
                +---rw names* string
        +---rw disjointness?
            te-path-disjointness

```



```

+---rw explicit-route-objects-always
+---rw route-object-exclude-always* [index]
+---rw index                                uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id          te-node-id
+---rw hop-type?        te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id        te-tp-id
+---rw hop-type?        te-hop-type
+---rw direction?
+---rw                        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?
+---rw                        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?
+---rw                        rt-types:generalized
- label
+---rw direction?
+---rw                        te-label-direction
+---rw route-object-include-exclude* [index]
+---rw explicit-route-usage?
+---rw identityref
+---rw index                                uint32
+---rw (type)?
+---:(numbered-node-hop)
+---rw numbered-node-hop
+---rw node-id          te-node-id
+---rw hop-type?        te-hop-type
+---:(numbered-link-hop)
+---rw numbered-link-hop
+---rw link-tp-id        te-tp-id
+---rw hop-type?        te-hop-type

```

				<pre> +---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 </pre>
-label				<pre> +---rw direction? te-label-direction +---:(srlg) +---rw srlg +---rw srlg? uint32 +---rw path-in-segment! +---rw label-restrictions +---rw label-restriction* [index] +---rw restriction? enumeration +---rw index uint32 +---rw label-start +---rw te-label +---rw (technology)? +---:(generic) +---rw generic? rt-types:generalized-la </pre>
bel				<pre> +---rw direction? te-label-direction +---rw label-end +---rw te-label +---rw (technology)? +---:(generic) +---rw generic? rt-types:generalized-la </pre>
bel				<pre> +---rw direction? </pre>

					<pre> te-label-direction +--rw label-step +--rw (technology)? +--:(generic) +--rw generic? int32 +--rw range-bitmap? yang:hex-string +--rw path-out-segment! +--rw label-restrictions +--rw label-restriction* [index] +--rw restriction? enumeration +--rw index uint32 +--rw label-start +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel					
					<pre> +--rw direction? te-label-direction +--rw label-end +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel					
					<pre> +--rw direction? te-label-direction +--rw label-step +--rw (technology)? +--:(generic) +--rw generic? int32 +--rw range-bitmap? yang:hex-string +--ro computed-paths-properties +--ro computed-path-properties* [k-index] +--ro k-index uint8 +--ro path-properties +--ro path-metric* [metric-type] +--ro metric-type identityref +--ro accumulative-value? uint64 +--ro path-affinities-values +--ro path-affinities-value* [usage] +--ro usage identityref +--ro value? admin-groups +--ro path-affinity-names +--ro path-affinity-name* [usage] </pre>

```

+---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
+---ro (type)?
+---:(numbered-node-hop)
+---ro numbered-node-hop
+---ro node-id
+---ro hop-type?
+---ro te-hop-type
+---:(numbered-link-hop)
+---ro numbered-link-hop
+---ro link-tp-id
+---ro hop-type?
+---ro te-hop-type
+---ro direction?
+---ro te-link-direction
+---:(unnumbered-link-hop)
+---ro unnumbered-link-hop
+---ro link-tp-id
+---ro node-id
+---ro hop-type?
+---ro te-hop-type
+---ro direction?
+---ro te-link-direction
+---:(as-number)
+---ro as-number-hop
+---ro as-number
+---ro hop-type?
+---ro te-hop-type
+---:(label)
+---ro label-hop

```

```

+--ro te-label
+--ro (technology)?
+--:(generic)
+--ro generic?
rt-types:ge

neralized-label

+--ro direction?
te-label-directio

n

+--ro te-bandwidth
+--ro (technology)?
+--:(generic)
+--ro generic? te-bandwidth
+--ro disjointness-type?
te-types:te-path-disjointness
+--ro computed-path-error-infos
+--ro computed-path-error-info* []
+--ro error-description? string
+--ro error-timestamp?
| yang:date-and-time
+--ro error-reason? identityref
+--ro lsp-provisioning-error-infos
+--ro lsp-provisioning-error-info* []
+--ro error-description? string
+--ro error-timestamp?
| yang:date-and-time
+--ro error-node-id?
| te-types:te-node-id
+--ro error-link-id?
| te-types:te-tp-id
+--ro lsp-id? uint16
+--ro lsps
+--ro lsp* [node lsp-id]
+--ro tunnel-name?
| -> /te/lsps/lsp/tunnel-name
+--ro node -> /te/lsps/lsp/node
+--ro lsp-id -> /te/lsps/lsp/lsp-id
+--rw candidate-secondary-reverse-paths
+--rw candidate-secondary-reverse-path*
[secondary-path]
+--rw secondary-path leafref
+--rw candidate-secondary-paths
+--rw candidate-secondary-path* [secondary-path]
+--rw secondary-path leafref
+--ro active? boolean
+--rw secondary-paths
+--rw secondary-path* [name]
+--rw name string

```

```

+--rw path-computation-method?          identityref
+--rw path-computation-server
|   +--rw id?          te-gen-node-id
|   +--rw type?        enumeration
+--rw compute-only?                    empty
+--rw use-path-computation?            boolean
+--rw lockdown?                        empty
+--ro path-scope?                      identityref
+--rw preference?                      uint8
+--rw association-objects
|   +--rw association-object* [association-key]
|   |   +--rw association-key      string
|   |   +--rw type?                identityref
|   |   +--rw id?                  uint16
|   |   +--rw source
|   |   |   +--rw id?          te-gen-node-id
|   |   |   +--rw type?        enumeration
|   +--rw association-object-extended*
|   |   [association-key]
|   |   +--rw association-key      string
|   |   +--rw type?                identityref
|   |   +--rw id?                  uint16
|   |   +--rw source
|   |   |   +--rw id?          te-gen-node-id
|   |   |   +--rw type?        enumeration
|   |   +--rw global-source?      uint32
|   |   +--rw extended-id?        yang:hex-string
+--rw optimizations
|   +--rw (algorithm)?
|   |   +--:(metric) {path-optimization-metric}?
|   |   |   +--rw optimization-metric* [metric-type]
|   |   |   |   +--rw metric-type
|   |   |   |   |   identityref
|   |   |   |   +--rw weight?
|   |   |   |   |   uint8
|   |   |   +--rw explicit-route-exclude-objects
|   |   |   |   +--rw route-object-exclude-object*
|   |   |   |   |   [index]
|   |   |   |   |   +--rw index
|   |   |   |   |   |   uint32
|   |   |   |   +--rw (type)?
|   |   |   |   |   +--:(numbered-node-hop)
|   |   |   |   |   |   +--rw numbered-node-hop
|   |   |   |   |   |   |   +--rw node-id
|   |   |   |   |   |   |   |   te-node-id
|   |   |   |   |   |   |   +--rw hop-type?
|   |   |   |   |   |   |   |   te-hop-type
|   |   |   |   +--:(numbered-link-hop)

```

```

+--rw numbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
|   +--rw (technology)?
|   |   +--:(generic)
|   |   +--rw generic?
|   |   rt-types:ge
neralized-label
n
+--rw direction?
|   te-label-directio
+--:(srlg)
+--rw srlg
|   +--rw srlg?   uint32
+--rw explicit-route-include-objects
+--rw route-object-include-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
|   +--rw node-id
|   |   te-node-id
+--rw hop-type?

```

						te-hop-type
					+	---:(numbered-link-hop)
						+---rw numbered-link-hop
						+---rw link-tp-id
						te-tp-id
						+---rw hop-type?
						te-hop-type
						+---rw direction?
						te-link-direction
					+	---:(unnumbered-link-hop)
						+---rw unnumbered-link-hop
						+---rw link-tp-id
						te-tp-id
						+---rw node-id
						te-node-id
						+---rw hop-type?
						te-hop-type
						+---rw direction?
						te-link-direction
					+	---:(as-number)
						+---rw as-number-hop
						+---rw as-number
						inet:as-number
						+---rw hop-type?
						te-hop-type
					+	---:(label)
						+---rw label-hop
						+---rw te-label
						+---rw (technology)?
						+---:(generic)
						+---rw generic?
						rt-types:ge
neralized-label						
						+---rw direction?
						te-label-directio
n						
						+---rw tiebreakers
						+---rw tiebreaker* [tiebreaker-type]
						+---rw tiebreaker-type identityref
					+	---:(objective-function)
						{path-optimization-objective-function}?
						+---rw objective-function
						+---rw objective-function-type?
						identityref
					+	---rw named-path-constraint? leafref
						{te-types:named-path-constraints}?
					+	---rw te-bandwidth
						+---rw (technology)?


```

    +---:(generic)
      +---rw generic?    te-bandwidth
+---rw link-protection?          identityref
+---rw setup-priority?          uint8
+---rw hold-priority?          uint8
+---rw signaling-type?          identityref
+---rw path-metric-bounds
  +---rw path-metric-bound* [metric-type]
  +---rw metric-type          identityref
  +---rw upper-bound?          uint64
+---rw path-affinities-values
  +---rw path-affinities-value* [usage]
  +---rw usage                  identityref
  +---rw value?                 admin-groups
+---rw path-affinity-names
  +---rw path-affinity-name* [usage]
  +---rw usage                  identityref
  +---rw affinity-name* [name]
  +---rw name                    string
+---rw path-srlgs-lists
  +---rw path-srlgs-list* [usage]
  +---rw usage                  identityref
  +---rw values*                srlg
+---rw path-srlgs-names
  +---rw path-srlgs-name* [usage]
  +---rw usage                  identityref
  +---rw names*                 string
+---rw disjointness?
  +---rw te-path-disjointness
+---rw explicit-route-objects-always
  +---rw route-object-exclude-always* [index]
  +---rw index                  uint32
  +---rw (type)?
    +---:(numbered-node-hop)
      +---rw numbered-node-hop
      +---rw node-id            te-node-id
      +---rw hop-type?          te-hop-type
    +---:(numbered-link-hop)
      +---rw numbered-link-hop
      +---rw link-tp-id         te-tp-id
      +---rw hop-type?          te-hop-type
      +---rw direction?         te-link-direction
    +---:(unnumbered-link-hop)
      +---rw unnumbered-link-hop
      +---rw link-tp-id         te-tp-id
      +---rw node-id            te-node-id
      +---rw hop-type?          te-hop-type
      +---rw direction?         te-link-direction

```

				<pre> +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--rw route-object-include-exclude* [index] +--rw explicit-route-usage? identityref +--rw index uint32 +--rw (type)? +--:(numbered-node-hop) +--rw numbered-node-hop +--rw node-id te-node-id +--rw hop-type? te-hop-type +--:(numbered-link-hop) +--rw numbered-link-hop +--rw link-tp-id te-tp-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(unnumbered-link-hop) +--rw unnumbered-link-hop +--rw link-tp-id te-tp-id +--rw node-id te-node-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--:(srlg) </pre>

```

        +---rw srlg
            +---rw srlg?   uint32
+---rw path-in-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-step
                +---rw (technology)?
                    +---:(generic)
                        +---rw generic?   int32
            +---rw range-bitmap?   yang:hex-string
+---rw path-out-segment!
    +---rw label-restrictions
        +---rw label-restriction* [index]
            +---rw restriction?   enumeration
            +---rw index          uint32
            +---rw label-start
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?
                        te-label-direction
            +---rw label-end
                +---rw te-label
                    +---rw (technology)?
                        +---:(generic)
                            +---rw generic?
                                rt-types:generalized-label
                    +---rw direction?

```

```

|                                     te-label-direction
|   +---rw label-step
|   |   +---rw (technology)?
|   |   |   +---:(generic)
|   |   |   +---rw generic?    int32
|   +---rw range-bitmap?    yang:hex-string
+---rw protection
|   +---rw enable?                                boolean
|   +---rw protection-type?                        identityref
|   +---rw protection-reversion-disable?          boolean
|   +---rw hold-off-time?                          uint32
|   +---rw wait-to-revert?                        uint16
|   +---rw aps-signal-id?                          uint8
+---rw restoration
|   +---rw enable?                                boolean
|   +---rw restoration-type?
|   |   identityref
|   +---rw restoration-scheme?
|   |   identityref
|   +---rw restoration-reversion-disable?          boolean
|   +---rw hold-off-time?                          uint32
|   +---rw wait-to-restore?                        uint16
|   +---rw wait-to-revert?                        uint16
+---ro computed-paths-properties
|   +---ro computed-path-properties* [k-index]
|   |   +---ro k-index                          uint8
|   |   +---ro path-properties
|   |   |   +---ro path-metric* [metric-type]
|   |   |   |   +---ro metric-type              identityref
|   |   |   |   +---ro accumulative-value?      uint64
|   |   +---ro path-affinities-values
|   |   |   +---ro path-affinities-value* [usage]
|   |   |   |   +---ro usage                    identityref
|   |   |   |   +---ro value?                  admin-groups
|   |   +---ro path-affinity-names
|   |   |   +---ro path-affinity-name* [usage]
|   |   |   |   +---ro usage                    identityref
|   |   |   |   +---ro affinity-name* [name]
|   |   |   |   |   +---ro name                string
|   |   +---ro path-srlgs-lists
|   |   |   +---ro path-srlgs-list* [usage]
|   |   |   |   +---ro usage                    identityref
|   |   |   |   +---ro values*                srlg
|   |   +---ro path-srlgs-names
|   |   |   +---ro path-srlgs-name* [usage]
|   |   |   |   +---ro usage                    identityref
|   |   |   |   +---ro names*                string
|   +---ro path-route-objects

```

				+--ro path-route-object* [index]
				+--ro index
				uint32
				+--ro (type)?
				+--:(numbered-node-hop)
				+--ro numbered-node-hop
				+--ro node-id te-node-id
				+--ro hop-type?
				te-hop-type
				+--:(numbered-link-hop)
				+--ro numbered-link-hop
				+--ro link-tp-id te-tp-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(unnumbered-link-hop)
				+--ro unnumbered-link-hop
				+--ro link-tp-id te-tp-id
				+--ro node-id
				te-node-id
				+--ro hop-type?
				te-hop-type
				+--ro direction?
				te-link-direction
				+--:(as-number)
				+--ro as-number-hop
				+--ro as-number
				inet:as-number
				+--ro hop-type?
				te-hop-type
				+--:(label)
				+--ro label-hop
				+--ro te-label
				+--ro (technology)?
				+--:(generic)
				+--ro generic?
				rt-types:gener
				+--ro direction?
				te-label-direction
				+--ro te-bandwidth
				+--ro (technology)?
				+--:(generic)
				+--ro generic? te-bandwidth
				+--ro disjointness-type?
				te-types:te-path-disjointness
				+--ro computed-path-error-infos

alized-label

```

    +--ro computed-path-error-info* []
      +--ro error-description?   string
      +--ro error-timestamp?     yang:date-and-time
      +--ro error-reason?        identityref
+--ro lsp-provisioning-error-infos
  +--ro lsp-provisioning-error-info* []
    +--ro error-description?   string
    +--ro error-timestamp?     yang:date-and-time
    +--ro error-node-id?       te-types:te-node-id
    +--ro error-link-id?       te-types:te-tp-id
    +--ro lsp-id?              uint16
+--ro lsps
  +--ro lsp* [node lsp-id]
    +--ro tunnel-name?
      |      -> /te/lsps/lsp/tunnel-name
    +--ro node                -> /te/lsps/lsp/node
    +--ro lsp-id              -> /te/lsps/lsp/lsp-id
+--rw secondary-reverse-paths
  +--rw secondary-reverse-path* [name]
    +--rw name                  string
    +--rw path-computation-method? identityref
    +--rw path-computation-server
      +--rw id?                 te-gen-node-id
      +--rw type?               enumeration
    +--rw compute-only?         empty
    +--rw use-path-computation? boolean
    +--rw lockdown?             empty
    +--ro path-scope?           identityref
    +--rw preference?           uint8
    +--rw association-objects
      +--rw association-object* [association-key]
        +--rw association-key   string
        +--rw type?             identityref
        +--rw id?               uint16
        +--rw source
          +--rw id?             te-gen-node-id
          +--rw type?           enumeration
      +--rw association-object-extended*
        [association-key]
        +--rw association-key   string
        +--rw type?             identityref
        +--rw id?               uint16
        +--rw source
          +--rw id?             te-gen-node-id
          +--rw type?           enumeration
        +--rw global-source?    uint32
        +--rw extended-id?      yang:hex-string
+--rw optimizations

```

```

+--rw (algorithm)?
+--:(metric) {path-optimization-metric}?
+--rw optimization-metric* [metric-type]
+--rw metric-type
|   identityref
+--rw weight?
|   uint8
+--rw explicit-route-exclude-objects
+--rw route-object-exclude-object*
|   [index]
+--rw index
|   uint32
+--rw (type)?
+--:(numbered-node-hop)
+--rw numbered-node-hop
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--:(numbered-link-hop)
+--rw numbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(unnumbered-link-hop)
+--rw unnumbered-link-hop
+--rw link-tp-id
|   te-tp-id
+--rw node-id
|   te-node-id
+--rw hop-type?
|   te-hop-type
+--rw direction?
|   te-link-direction
+--:(as-number)
+--rw as-number-hop
+--rw as-number
|   inet:as-number
+--rw hop-type?
|   te-hop-type
+--:(label)
+--rw label-hop
+--rw te-label
+--rw (technology)?
|   +--:(generic)

```

```

+---rw generic?
      rt-types:generic
n
+---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:ge

neralized-label

+---rw direction?
te-label-directio

n
+---rw tiebreakers
+---rw tiebreaker* [tiebreaker-type]
+---rw tiebreaker-type identityref
+---:(objective-function)
{path-optimization-objective-function}?
+---rw objective-function
+---rw objective-function-type?
identityref
+---rw named-path-constraint? leafref
{te-types:named-path-constraints}?
+---rw te-bandwidth
+---rw (technology)?
+---:(generic)
+---rw generic? te-bandwidth
+---rw link-protection? identityref
+---rw setup-priority? uint8
+---rw hold-priority? uint8
+---rw signaling-type? identityref
+---rw path-metric-bounds
+---rw path-metric-bound* [metric-type]
+---rw metric-type identityref
+---rw upper-bound? uint64
+---rw path-affinities-values
+---rw path-affinities-value* [usage]
+---rw usage identityref
+---rw value? admin-groups
+---rw path-affinity-names
+---rw path-affinity-name* [usage]
+---rw usage identityref
+---rw affinity-name* [name]
+---rw name string
+---rw path-srlgs-lists
+---rw path-srlgs-list* [usage]
+---rw usage identityref
+---rw values* srlg
+---rw path-srlgs-names
+---rw path-srlgs-name* [usage]
+---rw usage identityref
+---rw names* string
+---rw disjointness?

```

				te-path-disjointness
			+--rw	explicit-route-objects-always
			+--rw	route-object-exclude-always* [index]
			+--rw	index uint32
			+--rw	(type)?
			+--:	(numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--:	(numbered-link-hop)
			+--rw	numbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:	(unnumbered-link-hop)
			+--rw	unnumbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:	(as-number)
			+--rw	as-number-hop
			+--rw	as-number inet:as-number
			+--rw	hop-type? te-hop-type
			+--:	(label)
			+--rw	label-hop
			+--rw	te-label
			+--rw	(technology)?
			+--:	(generic)
			+--rw	generic?
				rt-types:generalized-la
			+--rw	direction?
				te-label-direction
			+--rw	route-object-include-exclude* [index]
			+--rw	explicit-route-usage? identityref
			+--rw	index uint32
			+--rw	(type)?
			+--:	(numbered-node-hop)
			+--rw	numbered-node-hop
			+--rw	node-id te-node-id
			+--rw	hop-type? te-hop-type
			+--:	(numbered-link-hop)
			+--rw	numbered-link-hop
			+--rw	link-tp-id te-tp-id
			+--rw	hop-type? te-hop-type
			+--rw	direction? te-link-direction
			+--:	(unnumbered-link-hop)

				<pre> +--rw unnumbered-link-hop +--rw link-tp-id te-tp-id +--rw node-id te-node-id +--rw hop-type? te-hop-type +--rw direction? te-link-direction +--:(as-number) +--rw as-number-hop +--rw as-number inet:as-number +--rw hop-type? te-hop-type +--:(label) +--rw label-hop +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-la </pre>
bel				<pre> +--rw direction? te-label-direction +--:(srlg) +--rw srlg +--rw srlg? uint32 +--rw path-in-segment! +--rw label-restrictions +--rw label-restriction* [index] +--rw restriction? enumeration +--rw index uint32 +--rw label-start +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-end +--rw te-label +--rw (technology)? +--:(generic) +--rw generic? rt-types:generalized-label +--rw direction? te-label-direction +--rw label-step +--rw (technology)? +--:(generic) +--rw generic? int32 +--rw range-bitmap? yang:hex-string </pre>

```

+--rw path-out-segment!
  +--rw label-restrictions
    +--rw label-restriction* [index]
      +--rw restriction?      enumeration
      +--rw index             uint32
      +--rw label-start
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-end
        +--rw te-label
          +--rw (technology)?
            +--:(generic)
              +--rw generic?
                rt-types:generalized-label
          +--rw direction?
            te-label-direction
      +--rw label-step
        +--rw (technology)?
          +--:(generic)
            +--rw generic?    int32
        +--rw range-bitmap?   yang:hex-string
+--rw protection
  +--rw enable?                boolean
  +--rw protection-type?       identityref
  +--rw protection-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-revert?        uint16
  +--rw aps-signal-id?         uint8
+--rw restoration
  +--rw enable?                boolean
  +--rw restoration-type?
    | identityref
  +--rw restoration-scheme?
    | identityref
  +--rw restoration-reversion-disable? boolean
  +--rw hold-off-time?         uint32
  +--rw wait-to-restore?       uint16
  +--rw wait-to-revert?        uint16
+--ro computed-paths-properties
  +--ro computed-path-properties* [k-index]
    +--ro k-index              uint8
  +--ro path-properties
    +--ro path-metric* [metric-type]

```

```

|   +---ro metric-type          identityref
|   +---ro accumulative-value?  uint64
+---ro path-affinities-values
|   +---ro path-affinities-value* [usage]
|   +---ro usage                identityref
|   +---ro value?               admin-groups
+---ro path-affinity-names
|   +---ro path-affinity-name* [usage]
|   +---ro usage                identityref
|   +---ro affinity-name* [name]
|   +---ro name                 string
+---ro path-srlgs-lists
|   +---ro path-srlgs-list* [usage]
|   +---ro usage                identityref
|   +---ro values*             srlg
+---ro path-srlgs-names
|   +---ro path-srlgs-name* [usage]
|   +---ro usage                identityref
|   +---ro names*              string
+---ro path-route-objects
|   +---ro path-route-object* [index]
|   +---ro index
|   |   uint32
|   +---ro (type)?
|   |   +---:(numbered-node-hop)
|   |   |   +---ro numbered-node-hop
|   |   |   |   +---ro node-id      te-node-id
|   |   |   |   +---ro hop-type?
|   |   |   |   |   te-hop-type
|   |   |   +---:(numbered-link-hop)
|   |   |   |   +---ro numbered-link-hop
|   |   |   |   |   +---ro link-tp-id  te-tp-id
|   |   |   |   |   +---ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +---ro direction?
|   |   |   |   |   |   te-link-direction
|   |   |   +---:(unnumbered-link-hop)
|   |   |   |   +---ro unnumbered-link-hop
|   |   |   |   |   +---ro link-tp-id  te-tp-id
|   |   |   |   |   +---ro node-id
|   |   |   |   |   |   te-node-id
|   |   |   |   |   +---ro hop-type?
|   |   |   |   |   |   te-hop-type
|   |   |   |   |   +---ro direction?
|   |   |   |   |   |   te-link-direction
|   |   +---:(as-number)
|   |   |   +---ro as-number-hop
|   |   |   +---ro as-number

```

```

|                                     |      inet:as-number
|                                     |      +---ro hop-type?
|                                     |          te-hop-type
|                                     |      +---:(label)
|                                     |          +---ro label-hop
|                                     |              +---ro te-label
|                                     |                  +---ro (technology)?
|                                     |                      +---:(generic)
|                                     |                          +---ro generic?
|                                     |                              rt-types:gener
alized-label
|                                     |      +---ro direction?
|                                     |          te-label-direction
|                                     |      +---ro te-bandwidth
|                                     |          +---ro (technology)?
|                                     |              +---:(generic)
|                                     |                  +---ro generic?    te-bandwidth
|                                     |      +---ro disjointness-type?
|                                     |          te-types:te-path-disjointness
+---ro computed-path-error-infos
|   +---ro computed-path-error-info* []
|       +---ro error-description?    string
|       +---ro error-timestamp?     yang:date-and-time
|       +---ro error-reason?        identityref
+---ro lsp-provisioning-error-infos
|   +---ro lsp-provisioning-error-info* []
|       +---ro error-description?    string
|       +---ro error-timestamp?     yang:date-and-time
|       +---ro error-node-id?       te-types:te-node-id
|       +---ro error-link-id?       te-types:te-tp-id
|       +---ro lsp-id?               uint16
+---ro lsps
|   +---ro lsp* [node lsp-id]
|       +---ro tunnel-name?
|           |         -> /te/lsps/lsp/tunnel-name
|       +---ro node         -> /te/lsps/lsp/node
|       +---ro lsp-id       -> /te/lsps/lsp/lsp-id
+---x tunnel-action
|   +---w input
|       |   +---w action-type?    identityref
|   +---ro output
|       +---ro action-result?    identityref
+---x protection-external-commands
|   +---w input
|       +---w protection-external-command?
|           |   identityref
|   +---w protection-group-ingress-node-id?
|       |   te-types:te-node-id

```

```

|         +---w protection-group-egress-node-id?
|         |         te-types:te-node-id
|         +---w path-ref?                                path-ref
|         +---w traffic-type?
|         |         enumeration
|         +---w extra-traffic-tunnel-ref?                tunnel-ref
+---ro lsp
+---ro lsp* [tunnel-name lsp-id node]
+---ro tunnel-name                                string
+---ro lsp-id                                    uint16
+---ro node
|         te-types:te-node-id
+---ro source?
|         te-types:te-node-id
+---ro destination?
|         te-types:te-node-id
+---ro tunnel-id?                                uint16
+---ro extended-tunnel-id?                       yang:dotted-quad
+---ro operational-state?                       identityref
+---ro signaling-type?                         identityref
+---ro origin-type?                           enumeration
+---ro lsp-resource-status?                   enumeration
+---ro lockout-of-normal?                     boolean
+---ro freeze?                               boolean
+---ro lsp-protection-role?                   enumeration
+---ro lsp-protection-state?                 identityref
+---ro protection-group-ingress-node-id?
|         te-types:te-node-id
+---ro protection-group-egress-node-id?
|         te-types:te-node-id
+---ro lsp-record-route-information
+---ro lsp-record-route-information* [index]
+---ro index                                    uint32
+---ro (type)?
+---:(numbered-node-hop)
|         +---ro numbered-node-hop
|         |         +---ro node-id      te-node-id
|         |         +---ro flags*      path-attribute-flags
+---:(numbered-link-hop)
|         +---ro numbered-link-hop
|         |         +---ro link-tp-id   te-tp-id
|         |         +---ro flags*      path-attribute-flags
+---:(unnumbered-link-hop)
|         +---ro unnumbered-link-hop
|         |         +---ro link-tp-id   te-tp-id
|         |         +---ro node-id?    te-node-id
|         |         +---ro flags*      path-attribute-flags
+---:(label)

```

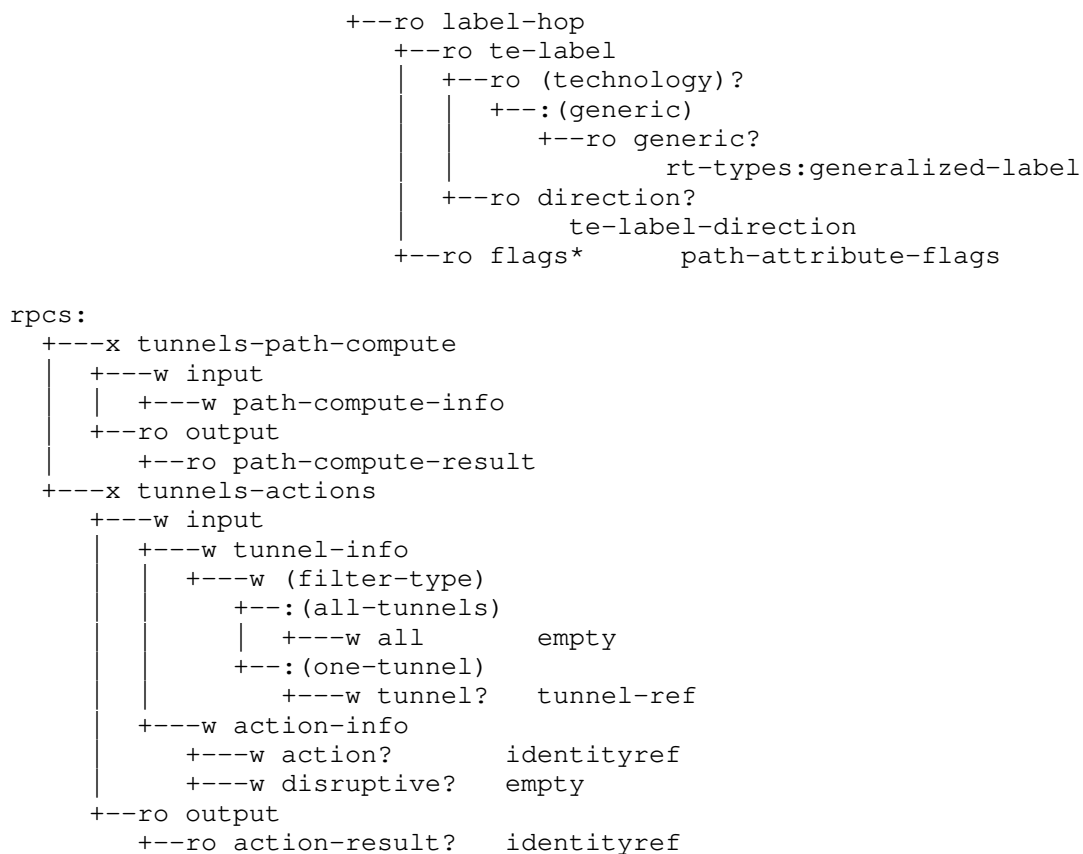


Figure 8: TE Tunnel generic model YANG tree diagram

5.3. YANG Module

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

- ```
* ietf-yang-types and ietf-inet-types defined in [RFC6991]
* ietf-te-types defined in [RFC8776]
```

This module references the following documents: [RFC6991], [RFC4875], [RFC7551], [RFC4206], [RFC4427], [RFC4872], [RFC3945], [RFC3209], [RFC6780], [RFC8800], and [RFC7308].



```
<CODE BEGINS> file "ietf-te@2021-10-22.yang"
module ietf-te {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te";

 /* Replace with IANA when assigned */

 prefix te;

 /* Import TE generic types */

 import ietf-te-types {
 prefix te-types;
 reference
 "RFC8776: Common YANG Data Types for Traffic Engineering.";
 }
 import ietf-inet-types {
 prefix inet;
 reference
 "RFC6991: Common YANG Data Types.";
 }
 import ietf-yang-types {
 prefix yang;
 reference
 "RFC6991: Common YANG Data Types.";
 }
}

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group.";
contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

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

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

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

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

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

```
Editor: Igor Bryskin
 <mailto:i_bryskin@yahoo.com>;

description
 "YANG data module for TE configuration, state, and RPCs.
 The model fully conforms to the Network Management
 Datastore Architecture (NMDA).

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

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

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

revision 2021-10-22 {
 description
 "Latest update to TE generic YANG module.";
 reference
 "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
 and Interfaces.";
}

identity path-computation-error-reason {
 description
 "Base identity for path computation error reasons.";
}

identity path-computation-error-no-topology {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no topology
 with the provided topology-identifier.";
}

identity path-computation-error-no-dependent-server {
 base path-computation-error-reason;
 description
 "Path computation has failed because one or more dependent
```

```
 path computation servers are unavailable.
 The dependent path computation server could be
 a Backward-Recursive Path Computation (BRPC) downstream
 PCE or a child PCE.";
 reference
 "RFC5441, RFC8685";
}

identity path-computation-error-pce-unavailable {
 base path-computation-error-reason;
 description
 "Path computation has failed because PCE is not available.";
 reference
 "RFC5440";
}

identity path-computation-error-no-inclusion-hop {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no
 node or link provided by one or more inclusion hops.";
 reference
 "RFC8685";
}

identity path-computation-error-destination-unknown-in-domain {
 base path-computation-error-reason;
 description
 "Path computation has failed because the destination node is
 unknown in indicated destination domain.";
 reference
 "RFC8685";
}

identity path-computation-error-no-resource {
 base path-computation-error-reason;
 description
 "Path computation has failed because there is no
 available resource in one or more domains.";
 reference
 "RFC8685";
}

identity path-computation-error-child-pce-unresponsive {
 base path-computation-error-reason;
 description
 "Path computation has failed because child PCE is not
 responsive.";
```

```
 reference
 "RFC8685";
 }

 identity path-computation-error-destination-domain-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because the destination domain
 was unknown.";
 reference
 "RFC8685";
 }

 identity path-computation-error-p2mp {
 base path-computation-error-reason;
 description
 "Path computation has failed because of P2MP reachability
 problem.";
 reference
 "RFC8306";
 }

 identity path-computation-error-no-gco-migration {
 base path-computation-error-reason;
 description
 "Path computation has failed because of no Global Concurrent
 Optimization (GCO) migration path found.";
 reference
 "RFC5557";
 }

 identity path-computation-error-no-gco-solution {
 base path-computation-error-reason;
 description
 "Path computation has failed because of no GCO solution
 found.";
 reference
 "RFC5557";
 }

 identity path-computation-error-path-not-found {
 base path-computation-error-reason;
 description
 "Path computation no path found error reason.";
 reference
 "RFC5440";
 }
 }
```

```
identity path-computation-error-pks-expansion {
 base path-computation-error-reason;
 description
 "Path computation has failed because of Path-Key Subobject
 (PKS) expansion failure.";
 reference
 "RFC5520";
}

identity path-computation-error-brpc-chain-unavailable {
 base path-computation-error-reason;
 description
 "Path computation has failed because PCE BRPC chain
 unavailable.";
 reference
 "RFC5441";
}

identity path-computation-error-source-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because source node is unknown.";
 reference
 "RFC5440";
}

identity path-computation-error-destination-unknown {
 base path-computation-error-reason;
 description
 "Path computation has failed because destination node is
 unknown.";
 reference
 "RFC5440";
}

identity path-computation-error-no-server {
 base path-computation-error-reason;
 description
 "Path computation has failed because path computation
 server is unavailable.";
 reference
 "RFC5440";
}

identity tunnel-actions-type {
 description
 "TE tunnel actions type.";
}
```

```
identity tunnel-action-reoptimize {
 base tunnel-actions-type;
 description
 "Reoptimize tunnel action type.";
}

identity tunnel-admin-auto {
 base te-types:tunnel-admin-state-type;
 description
 "Tunnel administrative auto state. The administrative status
 in state datastore transitions to 'tunnel-admin-up' when the
 tunnel used by the client layer, and to 'tunnel-admin-down'
 when it is not used by the client layer.";
}

identity association-type-diversity {
 base te-types:association-type;
 description
 "Association Type diversity used to associate LSPs whose paths
 are to be diverse from each other.";
 reference
 "RFC8800";
}

identity protocol-origin-type {
 description
 "Base identity for protocol origin type.";
}

identity protocol-origin-api {
 base protocol-origin-type;
 description
 "Protocol origin is via Application Programmable Interface
 (API).";
}

identity protocol-origin-pcep {
 base protocol-origin-type;
 description
 "Protocol origin is Path Computation Engine Protocol (PCEP).";
 reference "RFC5440";
}

identity protocol-origin-bgp {
 base protocol-origin-type;
 description
 "Protocol origin is Border Gateway Protocol (BGP).";
 reference "RFC5512";
}

typedef tunnel-ref {
```

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

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

 typedef te-gen-node-id {
 type union {
 type te-types:te-node-id;
 type inet:ip-address;
 }
 description
 "Generic type that identifies a node in a TE topology.";
 }

 /**
 * TE tunnel generic groupings
 */

 grouping te-generic-node-id {
 description
 "A reusable grouping for a TE generic node identifier.";
 leaf id {
 type te-gen-node-id;
 description
 "The identifier of the node. Can be represented as IP
 address or dotted quad address.";
 }
 leaf type {
 type enumeration {
```

```
 enum ip {
 description
 "IP address representation of the node identifier.";
 }
 enum dotted-quad {
 description
 "Dotted quad address representation of the node
 identifier.";
 }
 }
 description
 "Type of node identifier representation.";
}

grouping primary-path {
 description
 "The tunnel primary path properties.";
 uses path-common-properties;
 uses path-preference;
 uses k-requested-paths;
 uses path-compute-info;
 uses path-state;
}

grouping primary-reverse-path {
 description
 "The tunnel primary reverse path properties.";
 reference
 "RFC7551";
 uses path-common-properties;
 uses path-compute-info;
 uses path-state;
}

grouping secondary-path {
 description
 "The tunnel secondary path properties.";
 uses path-common-properties;
 uses path-preference;
 uses path-compute-info;
 uses protection-restoration-properties;
 uses path-state;
}

grouping secondary-reverse-path {
 description
 "The tunnel secondary reverse path properties.";
```



```
 uses path-common-properties;
 uses path-preference;
 uses path-compute-info;
 uses protection-restoration-properties;
 uses path-state;
}

grouping path-common-properties {
 description
 "Common path attributes.";
 leaf name {
 type string;
 description
 "TE path name.";
 }
 leaf path-computation-method {
 type identityref {
 base te-types:path-computation-method;
 }
 default "te-types:path-locally-computed";
 description
 "The method used for computing the path, either
 locally computed, queried from a server or not
 computed at all (explicitly configured).";
 }
 container path-computation-server {
 when "derived-from-or-self(..path-computation-method, "
 + "'te-types:path-externally-queried') " {
 description
 "The path-computation server when the path is
 externally queried.";
 }
 uses te-generic-node-id;
 description
 "Address of the external path computation
 server.";
 }
 leaf compute-only {
 type empty;
 description
 "When set, the path is computed and updated whenever
 the topology is updated. No resources are committed
 or reserved in the network.";
 }
 leaf use-path-computation {
 when "derived-from-or-self(..path-computation-method, "
 + "'te-types:path-locally-computed') " {
 type boolean;
 }
 }
}
```

```
 default "true";
 description
 "When 'true' indicates the path is dynamically computed
 and/or validated against the Traffic-Engineering Database
 (TED), and when 'false' indicates no validation against
 the TED is required.";
}
leaf lockdown {
 type empty;
 description
 "Indicates no reoptimization to be attempted for this path.";
}
leaf path-scope {
 type identityref {
 base te-types:path-scope-type;
 }
 default "te-types:path-scope-end-to-end";
 config false;
 description
 "Path scope if segment or an end-to-end path.";
}
}

/* This grouping will be re-used in path-computation rpc */

grouping path-compute-info {
 description
 "Attributes used for path computation request.";
 uses tunnel-associations-properties;
 uses te-types:generic-path-optimization;
 leaf named-path-constraint {
 if-feature "te-types:named-path-constraints";
 type leafref {
 path "/te:te/te:globals/te:named-path-constraints/"
 + "te:named-path-constraint/te:name";
 }
 description
 "Reference to a globally defined named path constraint set.";
 }
 uses path-constraints-common;
}

/* This grouping will be re-used in path-computation rpc */

grouping path-preference {
 description
 "The path preference.";
 leaf preference {
```

```
 type uint8 {
 range "1..255";
 }
 default "1";
 description
 "Specifies a preference for this path. The lower the number
 higher the preference.";
 }
}

/* This grouping will be re-used in path-computation rpc */

grouping k-requested-paths {
 description
 "The k-shortest paths requests.";
 leaf k-requested-paths {
 type uint8;
 default "1";
 description
 "The number of k-shortest-paths requested from the path
 computation server and returned sorted by its optimization
 objective. The value 0 all possible paths.";
 }
}

grouping path-properties {
 description
 "TE computed path properties grouping.";
 uses te-types:generic-path-properties {
 augment "path-properties" {
 description
 "additional path properties returned by path computation.";
 uses te-types:te-bandwidth;
 leaf disjointness-type {
 type te-types:te-path-disjointness;
 config false;
 description
 "The type of resource disjointness.
 When reported for a primary path, it represents the
 minimum level of disjointness of all the secondary
 paths.
 When reported for a secondary path, it represents the
 disjointness of the secondary path.";
 }
 }
 }
}
```

```
grouping path-state {
 description
 "TE per path state parameters.";
 uses path-computation-response;
 uses lsp-provisioning-error-info {
 augment "lsp-provisioning-error-infos/"
 + "lsp-provisioning-error-info" {
 description
 "Augmentation of LSP provisioning information under a
 specific path.";
 leaf lsp-id {
 type uint16;
 description
 "The LSP-ID for which path computation was performed.";
 }
 }
 }
}
container lsps {
 config false;
 description
 "The TE LSPs container.";
 list lsp {
 key "node lsp-id";
 description
 "List of LSPs associated with the tunnel.";
 leaf tunnel-name {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:tunnel-name";
 }
 description "TE tunnel name.";
 }
 leaf node {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:node";
 }
 description "The node where the LSP state resides on.";
 }
 leaf lsp-id {
 type leafref {
 path "/te:te/te:lsps/te:lsp/te:lsp-id";
 }
 description "The TE LSP identifier.";
 }
 }
}

/* This grouping will be re-used in path-computation rpc */
```

```
grouping path-computation-response {
 description
 "Attributes reported by path computation response.";
 container computed-paths-properties {
 config false;
 description
 "Computed path properties container.";
 list computed-path-properties {
 key "k-index";
 description
 "List of computed paths.";
 leaf k-index {
 type uint8;
 description
 "The k-th path returned from the computation server.
 A lower k value path is more optimal than higher k
 value path(s)";
 }
 uses path-properties {
 description
 "The TE path computed properties.";
 }
 }
 }
}
container computed-path-error-infos {
 config false;
 description
 "Path computation information container.";
 list computed-path-error-info {
 description
 "List of path computation info entries.";
 leaf error-description {
 type string;
 description
 "Textual representation of the error occurred during
 path computation.";
 }
 leaf error-timestamp {
 type yang:date-and-time;
 description
 "Timestamp of last path computation attempt.";
 }
 leaf error-reason {
 type identityref {
 base path-computation-error-reason;
 }
 description
 "Reason for the path computation error.";
 }
 }
}
```

```
 }
 }
}

grouping lsp-provisioning-error-info {
 description
 "Grouping for LSP provisioning error information.";
 container lsp-provisioning-error-infos {
 config false;
 description
 "LSP provisioning error information.";
 list lsp-provisioning-error-info {
 description
 "List of LSP provisioning error info entries.";
 leaf error-description {
 type string;
 description
 "Textual representation of the error occurred during
 path computation.";
 }
 leaf error-timestamp {
 type yang:date-and-time;
 description
 "Timestamp of when the reported error occurred.";
 }
 leaf error-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Node identifier of node where error occurred.";
 }
 leaf error-link-id {
 type te-types:te-tp-id;
 default "0";
 description
 "Link ID where the error occurred.";
 }
 }
 }
}

grouping protection-restoration-properties-state {
 description
 "Protection parameters grouping.";
 leaf lockout-of-normal {
 type boolean;
 default "false";
 }
}
```

```
description
 "When set to 'True', it represents a lockout of normal
 traffic external command. When set to 'False', it
 represents a clear lockout of normal traffic external
 command. The lockout of normal traffic command applies
 to this Tunnel.";
reference
 "RFC4427";
}
leaf freeze {
 type boolean;
 default "false";
 description
 "When set to 'True', it represents a freeze external command.
 When set to 'False', it represents a clear freeze external
 command. The freeze command applies to all the Tunnels which
 are sharing the protection resources with this Tunnel.";
 reference
 "RFC4427";
}
leaf lsp-protection-role {
 type enumeration {
 enum working {
 description
 "A working LSP must be a primary LSP whilst a protecting
 LSP can be either a primary or a secondary LSP. Also,
 known as protected LSPs when working LSPs are associated
 with protecting LSPs.";
 }
 enum protecting {
 description
 "A secondary LSP is an LSP that has been provisioned
 in the control plane only; e.g. resource allocation
 has not been committed at the data plane.";
 }
 }
 default "working";
 description
 "LSP role type.";
 reference
 "RFC4872, section 4.2.1";
}
leaf lsp-protection-state {
 type identityref {
 base te-types:lsp-protection-state;
 }
 default "te-types:normal";
 description
```

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

grouping protection-restoration-properties {
 description
 "Protection and restoration parameters.";
 container protection {
 description
 "Protection parameters.";
 leaf enable {
 type boolean;
 default "false";
 description
 "A flag to specify if LSP protection is enabled.";
 reference
 "RFC4427";
 }
 leaf protection-type {
```



```
 type identityref {
 base te-types:lsp-protection-type;
 }
 default "te-types:lsp-protection-unprotected";
 description
 "LSP protection type.";
 }
 leaf protection-reversion-disable {
 type boolean;
 default "false";
 description
 "Disable protection reversion to working path.";
 }
 leaf hold-off-time {
 type uint32;
 units "milli-seconds";
 default "0";
 description
 "The time between the declaration of an SF or SD condition
 and the initialization of the protection switching
 algorithm.";
 reference
 "RFC4427";
 }
 leaf wait-to-revert {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP reversion.";
 reference
 "RFC4427";
 }
 leaf aps-signal-id {
 type uint8 {
 range "1..255";
 }
 default "1";
 description
 "The APS signal number used to reference the traffic of
 this tunnel. The default value for normal traffic is 1.
 The default value for extra-traffic is 255. If not
 specified, non-default values can be assigned by the
 server, if and only if, the server controls both
 endpoints.";
 reference
 "RFC4427";
 }
}
```

```
container restoration {
 description
 "Restoration parameters.";
 leaf enable {
 type boolean;
 default "false";
 description
 "A flag to specify if LSP restoration is enabled.";
 reference
 "RFC4427";
 }
 leaf restoration-type {
 type identityref {
 base te-types:lsp-restoration-type;
 }
 default "te-types:lsp-restoration-restore-any";
 description
 "LSP restoration type.";
 }
 leaf restoration-scheme {
 type identityref {
 base te-types:restoration-scheme-type;
 }
 default "te-types:restoration-scheme-preconfigured";
 description
 "LSP restoration scheme.";
 }
 leaf restoration-reversion-disable {
 type boolean;
 default "false";
 description
 "Disable restoration reversion to working path.";
 }
 leaf hold-off-time {
 type uint32;
 units "milli-seconds";
 description
 "The time between the declaration of an SF or SD condition
 and the initialization of the protection switching
 algorithm.";
 reference
 "RFC4427";
 }
 leaf wait-to-restore {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP restoration.";
```

```
 reference
 "RFC4427";
 }
 leaf wait-to-revert {
 type uint16;
 units "seconds";
 description
 "Time to wait before attempting LSP reversion.";
 reference
 "RFC4427";
 }
 }
 }
}

grouping tunnel-associations-properties {
 description
 "TE tunnel association grouping.";
 container association-objects {
 description
 "TE tunnel associations.";
 list association-object {
 key "association-key";
 unique "type id source/id source/type";
 description
 "List of association base objects.";
 reference
 "RFC4872";
 leaf association-key {
 type string;
 description
 "Association key used to identify a specific
 association in the list";
 }
 leaf type {
 type identityref {
 base te-types:association-type;
 }
 description
 "Association type.";
 reference
 "RFC4872";
 }
 leaf id {
 type uint16;
 description
 "Association identifier.";
 reference
 "RFC4872";
 }
 }
 }
}
```

```
 }
 container source {
 uses te-generic-node-id;
 description
 "Association source.";
 reference
 "RFC4872";
 }
 }
 list association-object-extended {
 key "association-key";
 unique
 "type id source/id source/type global-source extended-id";
 description
 "List of extended association objects.";
 reference
 "RFC6780";
 leaf association-key {
 type string;
 description
 "Association key used to identify a specific
 association in the list";
 }
 leaf type {
 type identityref {
 base te-types:association-type;
 }
 description
 "Association type.";
 reference
 "RFC4872, RFC6780";
 }
 leaf id {
 type uint16;
 description
 "Association identifier.";
 reference
 "RFC4872, RFC6780";
 }
 }
 container source {
 uses te-generic-node-id;
 description
 "Association source.";
 reference
 "RFC4872, RFC6780";
 }
 leaf global-source {
 type uint32;
```

```
 description
 "Association global source.";
 reference
 "RFC6780";
 }
 leaf extended-id {
 type yang:hex-string;
 description
 "Association extended identifier.";
 reference
 "RFC6780";
 }
}
}

/* TE tunnel configuration/state grouping */
/* These grouping will be re-used in path-computation rpc */

grouping encoding-and-switching-type {
 description
 "Common grouping to define the LSP encoding and
 switching types";
 leaf encoding {
 type identityref {
 base te-types:lsp-encoding-types;
 }
 description
 "LSP encoding type.";
 reference
 "RFC3945";
 }
 leaf switching-type {
 type identityref {
 base te-types:switching-capabilities;
 }
 description
 "LSP switching type.";
 reference
 "RFC3945";
 }
}

grouping tunnel-common-attributes {
 description
 "Common grouping to define the TE tunnel parameters";
 leaf source {
 type te-types:te-node-id;
```

```
 description
 "TE tunnel source node ID.";
 }
 leaf destination {
 type te-types:te-node-id;
 description
 "TE tunnel destination node identifier.";
 }
 leaf src-tunnel-tp-id {
 type binary;
 description
 "TE tunnel source termination point identifier.";
 }
 leaf dst-tunnel-tp-id {
 type binary;
 description
 "TE tunnel destination termination point identifier.";
 }
 leaf bidirectional {
 type boolean;
 default "false";
 description
 "Indicates a bidirectional co-routed LSP.";
 }
}

grouping tunnel-hierarchy-properties {
 description
 "A grouping for TE tunnel hierarchy information.";
 container hierarchy {
 description
 "Container for TE hierarchy related information.";
 container dependency-tunnels {
 description
 "List of tunnels that this tunnel can be potentially
 dependent on.";
 list dependency-tunnel {
 key "name";
 description
 "A tunnel entry that this tunnel can potentially depend
 on.";
 leaf name {
 type leafref {
 path "/te:te/te:tunnels/te:tunnel/te:name";
 require-instance false;
 }
 description
 "Dependency tunnel name. The tunnel may not have been
```

```
 instantiated yet.";
 }
 uses encoding-and-switching-type;
}
}
container hierarchical-link {
 description
 "Identifies a hierarchical link (in client layer)
 that this tunnel is associated with.";
 reference
 "RFC4206";
 leaf local-te-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "The local TE node identifier.";
 }
 leaf local-te-link-tp-id {
 type te-types:te-tp-id;
 default "0";
 description
 "The local TE link termination point identifier.";
 }
 leaf remote-te-node-id {
 type te-types:te-node-id;
 default "0.0.0.0";
 description
 "Remote TE node identifier.";
 }
 uses te-types:te-topology-identifier {
 description
 "The topology identifier where the hierarchical link
 supported by this TE tunnel is instantiated.";
 }
}
}
}

grouping tunnel-properties {
 description
 "Top level grouping for tunnel properties.";
 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 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 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.";
 list primary-path {
 key "name";
 description
 "List of primary paths for this tunnel.";
 uses primary-path;
 container primary-reverse-path {
 description
 "The reverse primary path properties.";
 uses primary-reverse-path;
 container candidate-secondary-reverse-paths {
 description
 "The set of referenced candidate reverse secondary
 paths from the full set of secondary reverse paths
 which may be used for this primary path.";
 list candidate-secondary-reverse-path {
 key "secondary-path";
 ordered-by user;
 description
 "List of candidate secondary reverse path(s)";
 leaf secondary-path {
 type leafref {
 path "../.../.../.../.../..."
 }
 }
 }
 }
 }
 }
}
```

```

 + "te:secondary-reverse-paths/"
 + "te:secondary-reverse-path/te:name";
 }
 description
 "A reference to the secondary reverse path that
 should be utilised when the containing primary
 reverse path option is in use.";
 }
}
}
}
container candidate-secondary-paths {
 description
 "The set of candidate secondary paths which may be used
 for this primary path. When secondary paths are
 specified in the list the path of the secondary LSP in
 use must be restricted to those path options referenced.
 The priority of the secondary paths is specified within
 the list. Higher priority values are less preferred -
 that is to say that a path with priority 0 is the most
 preferred path. In the case that the list is empty, any
 secondary path option may be utilised when the current
 primary path is in use.";
 list candidate-secondary-path {
 key "secondary-path";
 ordered-by user;
 description
 "List of candidate secondary paths for this tunnel.";
 leaf secondary-path {
 type leafref {
 path "../../../../../te:secondary-paths/"
 + "te:secondary-path/te:name";
 }
 description
 "A reference to the secondary path that should be
 utilised when the containing primary path option is
 in use.";
 }
 leaf active {
 type boolean;
 config false;
 description
 "Indicates the current active path option that has
 been selected of the candidate secondary paths.";
 }
 }
}
}
}

```

```
 }
 container secondary-paths {
 description
 "The set of secondary paths.";
 list secondary-path {
 key "name";
 description
 "List of secondary paths for this tunnel.";
 uses secondary-path;
 }
 }
 container secondary-reverse-paths {
 description
 "The set of secondary reverse paths.";
 list secondary-reverse-path {
 key "name";
 description
 "List of secondary paths for this tunnel.";
 uses secondary-reverse-path;
 }
 }
 }
}

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

```
}

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

```
 enum extra-traffic {
 description
 "The manual-switch or forced-switch command applies
 to the extra traffic (the extra-traffic Tunnel
 sharing protection bandwidth with this Tunnel).";
 }
 }
 description
 "Indicates whether the manual-switch or forced-switch
 commands applies to the normal traffic, the null traffic
 or the extra-traffic.";
 reference
 "RFC4427";
}
leaf extra-traffic-tunnel-ref {
 type tunnel-ref;
 description
 "In case there are multiple extra-traffic tunnels sharing
 protection bandwidth with this Tunnel (m:n protection),
 represents which extra-traffic Tunnel the manual-switch
 or forced-switch to extra-traffic command applies to.";
}
}
}

/** End of TE tunnel groupings */
/**
 * LSP related generic groupings
 */

grouping lsp-record-route-information-state {
 description
 "LSP Recorded route information grouping.";
 container lsp-record-route-information {
 description
 "RSVP recorded route object information.";
 list lsp-record-route-information {
 when "../origin-type = 'ingress'" {
 description
 "Applicable on ingress LSPs only.";
 }
 }
 key "index";
 description
 "Record route list entry.";
 uses te-types:record-route-state;
 }
}
```

```
 }

 grouping lsp-s-grouping {
 description
 "LSPs state operational data grouping.";
 container lsp {
 config false;
 description
 "TE LSPs state container.";
 list lsp {
 key "tunnel-name lsp-id node";
 unique "source destination tunnel-id lsp-id "
 + "extended-tunnel-id";
 description
 "List of LSPs associated with the tunnel.";
 leaf tunnel-name {
 type string;
 description "The TE tunnel name.";
 }
 leaf lsp-id {
 type uint16;
 description
 "Identifier used in the SENDER_TEMPLATE and the
 FILTER_SPEC that can be changed to allow a sender to
 share resources with itself.";
 reference
 "RFC3209";
 }
 leaf node {
 type te-types:te-node-id;
 description
 "The node where the TE LSP state resides on.";
 }
 uses lsp-properties-state;
 uses lsp-record-route-information-state;
 }
 }
 }

 /*** End of TE LSP groupings ***/
 /**
 * TE global generic groupings
 */
 /* Global named admin-groups configuration data */

 grouping named-admin-groups-properties {
 description
 "Global named administrative groups configuration
```

```
 grouping.";
 leaf name {
 type string;
 description
 "A string name that uniquely identifies a TE
 interface named admin-group.";
 }
 leaf bit-position {
 type uint32;
 description
 "Bit position representing the administrative group.";
 reference
 "RFC3209 and RFC7308";
 }
}

grouping named-admin-groups {
 description
 "Global named administrative groups configuration
 grouping.";
 container named-admin-groups {
 description
 "TE named admin groups container.";
 list named-admin-group {
 if-feature "te-types:extended-admin-groups";
 if-feature "te-types:named-extended-admin-groups";
 key "name";
 description
 "List of named TE admin-groups.";
 uses named-admin-groups-properties;
 }
 }
}

/* Global named admin-srlgs configuration data */

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

```
 type string;
 description
 "A string name that uniquely identifies a TE
 interface named SRLG.";
 }
 leaf value {
 type te-types:srlg;
 description
 "An SRLG value.";
 }
 leaf cost {
 type uint32;
 description
 "SRLG associated cost. Used during path to append
 the path cost when traversing a link with this SRLG.";
 }
}
}
}

/* Global named paths constraints configuration data */

grouping path-constraints-common {
 description
 "Global named path constraints configuration
 grouping.";
 uses te-types:common-path-constraints-attributes {
 description
 "The constraints applicable to the path. This includes:
 - The path bandwidth constraint
 - The path link protection type constraint
 - The path setup/hold priority constraint
 - path signaling type constraint
 - path metric bounds constraint. The unit of path metric
 bound is interpreted in the context of the metric-type.
 For example for metric-type 'path-metric-loss', the bound
 is multiples of the basic unit 0.000003% as described
 in RFC7471 for OSPF, and RFC8570 for ISIS.
 - path affinity constraints
 - path SRLG constraints";
 }
 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;
}
}

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

/* TE globals container data */

grouping globals-grouping {
 description
```

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

/* TE tunnels container data */

grouping tunnels-grouping {
 description
 "Tunnels TE configuration data grouping.";
 container tunnels {
 description
 "Tunnels TE configuration data container.";
 list tunnel {
 key "name";
 description
 "The list of TE tunnels.";
 uses tunnel-properties;
 uses tunnel-actions;
 uses tunnel-protection-actions;
 }
 }
}

/* TE LSPs ephemeral state container data */

grouping lsp-properties-state {
 description
 "LSPs state operational data grouping.";
 leaf source {
 type te-types:te-node-id;
 description
 "Tunnel sender address extracted from
 SENDER_TEMPLATE object.";
 reference
 "RFC3209";
 }
 leaf destination {
 type te-types:te-node-id;
 description
 "The tunnel endpoint address extracted from SESSION object.";
 reference
 "RFC3209";
 }
}
```

```
 }
 leaf tunnel-id {
 type uint16;
 description
 "The tunnel identifier used in the SESSION that remains
 constant over the life of the tunnel.";
 reference
 "RFC3209";
 }
 leaf extended-tunnel-id {
 type yang:dotted-quad;
 description
 "The LSP Extended Tunnel ID.";
 reference
 "RFC3209";
 }
 leaf operational-state {
 type identityref {
 base te-types:lsp-state-type;
 }
 description
 "The LSP operational state.";
 }
 leaf signaling-type {
 type identityref {
 base te-types:path-signaling-type;
 }
 description
 "The signaling protocol used to set up this LSP.";
 }
 leaf origin-type {
 type enumeration {
 enum ingress {
 description
 "Origin ingress.";
 }
 enum egress {
 description
 "Origin egress.";
 }
 enum transit {
 description
 "Origin transit.";
 }
 }
 default "ingress";
 description
 "The origin of the LSP relative to the location of the local
```

```
 switch in the path.";
 }
 leaf lsp-resource-status {
 type enumeration {
 enum primary {
 description
 "A primary LSP is a fully established LSP for which the
 resource allocation has been committed at the data
 plane.";
 }
 enum secondary {
 description
 "A secondary LSP is an LSP that has been provisioned
 in the control plane only; e.g. resource allocation
 has not been committed at the data plane.";
 }
 }
 default "primary";
 description
 "LSP resource allocation state.";
 reference
 "RFC4872, section 4.2.1";
 }
 uses protection-restoration-properties-state;
}

/** End of TE global groupings */
/**
 * TE container
 */

container te {
 presence "Enable TE feature.";
 description
 "TE global container.";
 /* TE Global Data */
 uses globals-grouping;

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

 /* TE LSPs Data */
 uses lsps-grouping;
}

/* TE Tunnel RPCs/execution Data */

rpc tunnels-path-compute {
```

```
description
 "TE tunnels RPC nodes.";
input {
 container path-compute-info {
 /*
 * An external path compute module may augment this
 * target.
 */
 description
 "RPC input information.";
 }
}
output {
 container path-compute-result {
 /*
 * An external path compute module may augment this
 * target.
 */
 description
 "RPC output information.";
 }
}

rpc tunnels-actions {
 description
 "TE tunnels actions RPC";
 input {
 container tunnel-info {
 description
 "TE tunnel information.";
 choice filter-type {
 mandatory true;
 description
 "Filter choice.";
 case all-tunnels {
 leaf all {
 type empty;
 mandatory true;
 description
 "Apply action on all TE tunnels.";
 }
 }
 case one-tunnel {
 leaf tunnel {
 type tunnel-ref;
 description
 "Apply action on the specific TE tunnel.";
 }
 }
 }
 }
 }
}
```

```

 }
 }
}
container action-info {
 description
 "TE tunnel action information.";
 leaf action {
 type identityref {
 base tunnel-actions-type;
 }
 description
 "The action type.";
 }
 leaf disruptive {
 when "derived-from-or-self(..../action, "
 + "'te:tunnel-action-reoptimize')";
 type empty;
 description
 "Specifies whether or not the reoptimization action
 is allowed to be disruptive.";
 }
}
}
output {
 leaf action-result {
 type identityref {
 base te-types:te-action-result;
 }
 description
 "The result of the tunnel action operation.";
 }
}
}
}
<CODE ENDS>

```

Figure 9: 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 10. 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 10: TE interface state YANG subtree

## 6.2. Tree Diagram

Figure 11 shows the tree diagram of the device TE YANG model defined in modules 'ietf-te.yang'.

```

module: ietf-te-device
 augment /te:te:
 +--rw interfaces
 |
 | +--rw threshold-type? enumeration
 | +--rw delta-percentage? rt-types:percentage
 | +--rw threshold-specification? enumeration
 | +--rw up-thresholds* rt-types:percentage
 | +--rw down-thresholds* rt-types:percentage
 | +--rw up-down-thresholds* rt-types:percentage
 | +--rw interface* [interface]
 | |
 | | +--rw interface if:interface-ref
 | | +--rw te-metric?
 | | | te-types:te-metric
 | | +--rw (admin-group-type)?
 | | | +--:(value-admin-groups)
 | | | | +--rw (value-admin-group-type)?
 | | | | | +--:(admin-groups)
 | | | | | | +--rw admin-group?
 | | | | | | | te-types:admin-group
 | | | | | +--:(extended-admin-groups)
 | | | | | | {te-types:extended-admin-groups}?
 | | | | | | +--rw extended-admin-group?
 | | | | | | | te-types:extended-admin-group
 | | | | +--:(named-admin-groups)
 | | | | | +--rw named-admin-groups* [named-admin-group]
 | | | | | | {te-types:extended-admin-groups,te-types:named-
 | | | | | | extended-admin-groups}?
 | | | | | | +--rw named-admin-group leafref
 | | | +--rw (srlg-type)?
 | | | | +--:(value-srlgs)
 | | | | | +--rw values* [value]

```



```

 | +---rw value uint32
 | +---:(named-srlgs)
 | | +---rw named-srlgs* [named-srlg]
 | | | {te-types:named-srlg-groups}?
 | | +---rw named-srlg leafref
 +---rw threshold-type? enumeration
 +---rw delta-percentage?
 | rt-types:percentage
 +---rw threshold-specification? enumeration
 +---rw up-thresholds*
 | rt-types:percentage
 +---rw down-thresholds*
 | rt-types:percentage
 +---rw up-down-thresholds*
 | rt-types:percentage
 +---rw switching-capabilities* [switching-capability]
 | +---rw switching-capability identityref
 | +---rw encoding? identityref
 +---ro state
 | +---ro te-advertisements-state
 | | +---ro flood-interval? uint32
 | | +---ro last-flooded-time? uint32
 | | +---ro next-flooded-time? uint32
 | | +---ro last-flooded-trigger? enumeration
 | | +---ro advertised-level-areas* [level-area]
 | | +---ro level-area uint32
 +---rw performance-thresholds
augment /te:te/te:globals:
 +---rw lsp-install-interval? uint32
 +---rw lsp-cleanup-interval? uint32
 +---rw lsp-invalidation-interval? uint32
augment /te:te/te:tunnels/te:tunnel:
 +---rw path-invalidation-action? identityref
 +---rw lsp-install-interval? uint32
 +---rw lsp-cleanup-interval? uint32
 +---rw lsp-invalidation-interval? uint32
augment /te:te/te:lsps/te:lsp:
 +---ro lsp-timers
 | +---ro life-time? uint32
 | +---ro time-to-install? uint32
 | +---ro time-to-destroy? uint32
 +---ro downstream-info
 | +---ro nhop? te-types:te-tp-id
 | +---ro outgoing-interface? if:interface-ref
 | +---ro neighbor
 | | +---ro id? te-gen-node-id
 | | +---ro type? enumeration
 +---ro label? rt-types:generalized-label

```

```

+--ro upstream-info
 +--ro phop? te-types:te-tp-id
 +--ro neighbor
 | +--ro id? te-gen-node-id
 | +--ro type? enumeration
 +--ro label? rt-types:generalized-label

rpcs:
 +---x link-state-update
 +---w input
 +---w (filter-type)
 +---:(match-all)
 | +---w all empty
 +---:(match-one-interface)
 +---w interface? if:interface-ref

```

Figure 11: 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-yang-types and ietf-inet-types defined in [RFC6991]
- \* ietf-interfaces defined in [RFC8343]
- \* ietf-routing-types defined in [RFC8294]
- \* ietf-te-types defined in [RFC8776]
- \* ietf-te defined in this document

```

<CODE BEGINS> file "ietf-te-device@2021-10-22.yang"
module ietf-te-device {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-device";

 /* Replace with IANA when assigned */

 prefix te-dev;

 /* Import TE module */

 import ietf-te {
 prefix te;
 reference
 "draft-ietf-teas-yang-te: A YANG Data Model for Traffic

```

```
 Engineering Tunnels and Interfaces";
}

/* Import TE types */

import ietf-te-types {
 prefix te-types;
 reference
 "RFC8776: Common YANG Data Types for Traffic Engineering.";
}
import ietf-interfaces {
 prefix if;
 reference
 "RFC8343: A YANG Data Model for Interface Management";
}
import ietf-routing-types {
 prefix rt-types;
 reference
 "RFC8294: Common YANG Data Types for the Routing Area";
}

organization
 "IETF Traffic Engineering Architecture and Signaling (TEAS)
 Working Group";
contact
 "WG Web: <http://tools.ietf.org/wg/teas/>
 WG List: <mailto:teas@ietf.org>

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

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

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

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

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

 Editor: Igor Bryskin
 <mailto:i_bryskin@yahoo.com>";
description
 "YANG data module for TE device configurations,
 state, and RPCs. The model fully conforms to the
```

Network Management Datastore Architecture (NMDA).

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

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

```
// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.
// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.
```

```
revision 2021-10-22 {
 description
 "Latest update to TE device YANG module.";
 reference
 "RFCXXXX: A YANG Data Model for Traffic Engineering Tunnels
 and Interfaces";
}
```

```
/**
 * TE LSP device state grouping
 */
```

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

```
 type uint32;
 units "seconds";
 description
 "TE LSP installation delay time.";
 }
 leaf time-to-destroy {
 type uint32;
 units "seconds";
 description
 "TE LSP expiration delay time.";
 }
}
container downstream-info {
 when "../te:origin-type != 'egress'" {
 description
 "Downstream information of the LSP.";
 }
 description
 "downstream information.";
 leaf nhop {
 type te-types:te-tp-id;
 description
 "downstream next-hop address.";
 }
 leaf outgoing-interface {
 type if:interface-ref;
 description
 "downstream interface.";
 }
 container neighbor {
 uses te:te-generic-node-id;
 description
 "downstream neighbor address.";
 }
 leaf label {
 type rt-types:generalized-label;
 description
 "downstream label.";
 }
}
container upstream-info {
 when "../te:origin-type != 'ingress'" {
 description
 "Upstream information of the LSP.";
 }
 description
 "upstream information.";
 leaf phop {
```

```
 type te-types:te-tp-id;
 description
 "upstream next-hop or previous-hop address.";
 }
 container neighbor {
 uses te:te-generic-node-id;
 description
 "upstream neighbor address.";
 }
 leaf label {
 type rt-types:generalized-label;
 description
 "upstream label.";
 }
}

/**
 * Device general groupings.
 */

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

/**
 * TE global device groupings
 */
/* TE interface container data */
```

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

/**
 * TE interface device groupings
 */

grouping te-admin-groups-config {
 description
 "TE interface affinities grouping.";
 choice admin-group-type {
 description
 "TE interface administrative groups
 representation type.";
 case value-admin-groups {
 choice value-admin-group-type {
 description
 "choice of admin-groups.";
 case admin-groups {
 description
 "Administrative group/Resource
 class/Color.";
 leaf admin-group {
 type te-types:admin-group;
 description
 "TE interface administrative group.";
 }
 }
 }
 case extended-admin-groups {
```

```
 if-feature "te-types:extended-admin-groups";
 description
 "Extended administrative group/Resource
 class/Color.";
 leaf extended-admin-group {
 type te-types:extended-admin-group;
 description
 "TE interface extended administrative group.";
 }
 }
}
}
case named-admin-groups {
 list named-admin-groups {
 if-feature "te-types:extended-admin-groups";
 if-feature "te-types:named-extended-admin-groups";
 key "named-admin-group";
 description
 "A list of named admin-group entries.";
 leaf named-admin-group {
 type leafref {
 path "../..../te:globals/"
 + "te:named-admin-groups/te:named-admin-group/"
 + "te:name";
 }
 description
 "A named admin-group entry.";
 }
 }
}
}
}

/* TE interface SRLGs */

grouping te-srlgs-config {
 description
 "TE interface SRLG grouping.";
 choice srlg-type {
 description
 "Choice of SRLG configuration.";
 case value-srlgs {
 list values {
 key "value";
 description
 "List of SRLG values that
 this link is part of.";
 leaf value {
```



```
 type uint32 {
 range "0..4294967295";
 }
 description
 "Value of the SRLG";
 }
}
}
case named-srlgs {
 list named-srlgs {
 if-feature "te-types:named-srlg-groups";
 key "named-srlg";
 description
 "A list of named SRLG entries.";
 leaf named-srlg {
 type leafref {
 path "../..../te:globals/"
 + "te:named-srlgs/te:named-srlg/te:name";
 }
 description
 "A named SRLG entry.";
 }
 }
}
}
}
}

grouping te-igp-flooding-bandwidth-config {
 description
 "Configurable items for igp flooding bandwidth
 threshold configuration.";
 leaf threshold-type {
 type enumeration {
 enum delta {
 description
 "'delta' indicates that the local
 system should flood IGP updates when a
 change in reserved bandwidth >= the specified
 delta occurs on the interface.";
 }
 enum threshold-crossed {
 description
 "THRESHOLD-CROSSED indicates that
 the local system should trigger an update (and
 hence flood) the reserved bandwidth when the
 reserved bandwidth changes such that it crosses,
 or becomes equal to one of the threshold values.";
 }
 }
 }
}
```

```
}
description
 "The type of threshold that should be used to specify the
 values at which bandwidth is flooded. 'delta' indicates that
 the local system should flood IGP updates when a change in
 reserved bandwidth >= the specified delta occurs on the
 interface. Where 'threshold-crossed' is specified, the local
 system should trigger an update (and hence flood) the
 reserved bandwidth when the reserved bandwidth changes such
 that it crosses, or becomes equal to one of the threshold
 values."
}
leaf delta-percentage {
 when "../threshold-type = 'delta'" {
 description
 "The percentage delta can only be specified when the
 threshold type is specified to be a percentage delta of
 the reserved bandwidth."
 }
 type rt-types:percentage;
 description
 "The percentage of the maximum-reservable-bandwidth
 considered as the delta that results in an IGP update
 being flooded."
}
leaf threshold-specification {
 when "../threshold-type = 'threshold-crossed'" {
 description
 "The selection of whether mirrored or separate threshold
 values are to be used requires user specified thresholds
 to be set."
 }
 type enumeration {
 enum mirrored-up-down {
 description
 "mirrored-up-down indicates that a single set of
 threshold values should be used for both increasing
 and decreasing bandwidth when determining whether
 to trigger updated bandwidth values to be flooded
 in the IGP TE extensions."
 }
 enum separate-up-down {
 description
 "separate-up-down indicates that a separate
 threshold values should be used for the increasing
 and decreasing bandwidth when determining whether
 to trigger updated bandwidth values to be flooded
 in the IGP TE extensions."
 }
 }
}
```

```
 }
 }
 description
 "This value specifies whether a single set of threshold
 values should be used for both increasing and decreasing
 bandwidth when determining whether to trigger updated
 bandwidth values to be flooded in the IGP TE extensions.
 'mirrored-up-down' indicates that a single value (or set of
 values) should be used for both increasing and decreasing
 values, where 'separate-up-down' specifies that the
 increasing and decreasing values will be separately
 specified.";
 }
 leaf-list up-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'separate-up-down'" {
 description
 "A list of up-thresholds can only be specified when the
 bandwidth update is triggered based on crossing a
 threshold and separate up and down thresholds are
 required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth) at which bandwidth updates are to be
 triggered when the bandwidth is increasing.";
 }
 leaf-list down-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'separate-up-down'" {
 description
 "A list of down-thresholds can only be specified when the
 bandwidth update is triggered based on crossing a
 threshold and separate up and down thresholds are
 required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth) at which bandwidth updates are to be
 triggered when the bandwidth is decreasing.";
 }
 leaf-list up-down-thresholds {
 when "../threshold-type = 'threshold-crossed'"
 + "and ../threshold-specification = 'mirrored-up-down'" {
 description
 "A list of thresholds corresponding to both increasing
```

```
 and decreasing bandwidths can be specified only when an
 update is triggered based on crossing a threshold, and
 the same up and down thresholds are required.";
 }
 type rt-types:percentage;
 description
 "The thresholds (expressed as a percentage of the maximum
 reservable bandwidth of the interface) at which bandwidth
 updates are flooded - used both when the bandwidth is
 increasing and decreasing.";
}
}

/* TE interface metric */

grouping te-metric-config {
 description
 "TE interface metric grouping.";
 leaf te-metric {
 type te-types:te-metric;
 description
 "TE interface metric.";
 }
}

/* TE interface switching capabilities */

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

```
 }
 }

 grouping te-advertisements-state {
 description
 "TE interface advertisements state grouping.";
 container te-advertisements-state {
 description
 "TE interface advertisements state container.";
 leaf flood-interval {
 type uint32;
 description
 "The periodic flooding interval.";
 }
 leaf last-flooded-time {
 type uint32;
 units "seconds";
 description
 "Time elapsed since last flooding in seconds.";
 }
 leaf next-flooded-time {
 type uint32;
 units "seconds";
 description
 "Time remained for next flooding in seconds.";
 }
 leaf last-flooded-trigger {
 type enumeration {
 enum link-up {
 description
 "Link-up flooding trigger.";
 }
 enum link-down {
 description
 "Link-down flooding trigger.";
 }
 enum threshold-up {
 description
 "Bandwidth reservation up threshold.";
 }
 enum threshold-down {
 description
 "Bandwidth reservation down threshold.";
 }
 enum bandwidth-change {
 description
 "Bandwidth capacity change.";
 }
 }
 }
 }
 }
```

```
 enum user-initiated {
 description
 "Initiated by user.";
 }
 enum srlg-change {
 description
 "SRLG property change.";
 }
 enum periodic-timer {
 description
 "Periodic timer expired.";
 }
 }
 default "periodic-timer";
 description
 "Trigger for the last flood.";
}
list advertised-level-areas {
 key "level-area";
 description
 "List of level-areas that the TE interface is advertised
 in.";
 leaf level-area {
 type uint32;
 description
 "The IGP area or level where the TE interface link state
 is advertised in.";
 }
}
}
}

/* TE interface attributes grouping */

grouping te-attributes {
 description
 "TE attributes configuration grouping.";
 uses te-metric-config;
 uses te-admin-groups-config;
 uses te-srlgs-config;
 uses te-igp-flooding-bandwidth-config;
 uses te-switching-cap-config;
 container state {
 config false;
 description
 "State parameters for interface TE metric.";
 uses te-advertisements-state;
 }
}
```

```
}

grouping te-all-attributes {
 description
 "TE attributes configuration grouping for all
 interfaces.";
 uses te-igp-flooding-bandwidth-config;
}

/** End of TE interfaces device groupings */
/**
 * TE device augmentations
 */

augment "/te:te" {
 description
 "TE global container.";
 /* TE Interface Configuration Data */
 uses interfaces-grouping;
 container performance-thresholds {
 description
 "Performance parameters configurable thresholds.";
 }
}

/* TE globals device augmentation */

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

/* TE tunnels device configuration augmentation */

augment "/te:te/te:tunnels/te:tunnel" {
 description
 "Tunnel device dependent augmentation.";
 leaf path-invalidation-action {
 type identityref {
 base te-types:path-invalidation-action-type;
 }
 description
 "Tunnel path invalidation action.";
 }
 uses lsp-device-timers;
}
```

```

/* TE LSPs device state augmentation */

augment "/te:te/te:lsps/te:lsp" {
 description
 "TE LSP device dependent augmentation.";
 uses lsp-device-info;
}

/* TE interfaces RPCs/execution Data */

rpc link-state-update {
 description
 "Triggers a link state update for the specific interface.";
 input {
 choice filter-type {
 mandatory true;
 description
 "Filter choice.";
 case match-all {
 leaf all {
 type empty;
 mandatory true;
 description
 "Match all TE interfaces.";
 }
 }
 case match-one-interface {
 leaf interface {
 type if:interface-ref;
 description
 "Match a specific TE interface.";
 }
 }
 }
 }
}
}

<CODE ENDS>

```

Figure 12: 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. TE Generic and Helper YANG Modules

## 9. IANA Considerations

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

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

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

This document registers two YANG modules in the YANG Module Names registry [RFC6020].

Name: ietf-te  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te  
Prefix: te  
Reference: RFCXXXX

Name: ietf-te-device  
Namespace: urn:ietf:params:xml:ns:yang:ietf-te-device  
Prefix: te-device  
Reference: RFCXXXX

## 10. Security Considerations

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

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

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

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

`"/te/tunnels"`: This list specifies the 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/lspss"`: 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.

## 11. Acknowledgement

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

The authors would like to thank Tom Petch for reviewing and providing useful feedback about the document. The authors would also like to thank Loa Andersson, Lou Berger, Sergio Belotti, Italo Busi, Carlo Perocchio, Francesco Lazzeri, Aihua Guo, Dhruv Dhody, and Raqib Jones for providing useful feedback on this document.

## 12. Contributors

Himanshu Shah  
Ciena

Email: hshah@ciena.com

Xia Chen  
Huawei Technologies

Email: jescia.chenxia@huawei.com

Bin Wen  
Comcast

Email: Bin\_Wen@cable.comcast.com

## 13. 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 where RSVP-TE MPLS Tunnels can be setup. The loopbacks of each router are shown. The network in Figure 13 will be used in the examples described in the following sections.

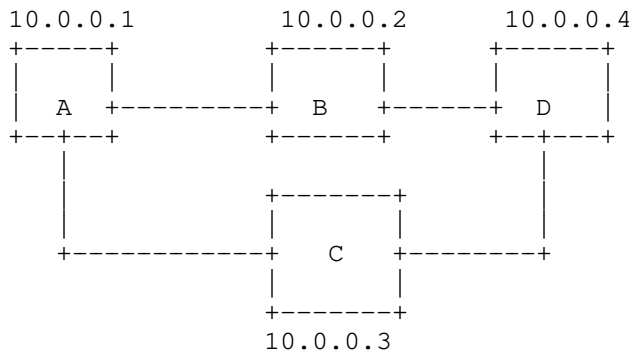


Figure 13: TE network used in data tree examples

### 13.1. Basic Tunnel Setup

This example uses the TE Tunnel YANG data model defined in this document to create an RSVP-TE signaled Tunnel 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": "10.0.0.1",
 "destination": "10.0.0.4",
 "bidirectional": "false",
 "signaling-type": "te-types:path-setup-rsvp"
 }
]
}

```

### 13.2. Global Named Path Constraints

This example uses the YANG data model to create a 'named path 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"
 }
 }
 }
}
```

### 13.3. Tunnel with Global Path Constraint

In this example, the previously created 'named path constraint' is applied to the TE Tunnel created in Section 13.1.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
 "ietf-te:ietf-tunnel": [
 {
 "name": "Example_LSP_Tunnel_A_4_1",
 "encoding": "te-types:lsp-encoding-packet",
 "description": "Simple_LSP_with_named_path",
 "admin-state": "te-types:tunnel-state-up",
 "source": "10.0.0.1",
 "destination": "10.0.0.4",
 "signaling-type": "path-setup-rsvp",
 "bidirectional": "false",
 "primary-paths": [
 {
 "primary-path": {
 "name": "Simple_LSP_1",
 "use-path-computation": "true",
 "named-path-constraint": "max-hop-3"
 }
 }
]
 }
]
}
```

#### 13.4. Tunnel with Per-tunnel Path Constraint

In this example, the a per tunnel path constraint is explicitly indicated under the TE Tunnel created in Section 13.1 to constrain the computed path for the tunnel.

```
POST /restconf/data/ietf-te:te/tunnels HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-data+json

{
 "ietf-te:tunnel": [
 {
 "name": "Example_LSP_Tunnel_A_4_2",
 "encoding": "te-types:lsp-encoding-packet",
 "admin-state": "te-types:tunnel-state-up",
 "source": "10.0.0.1",
 "destination": "10.0.0.4",
 "bidirectional": "false",
 "signaling-type": "te-types:path-setup-rsvp",
 "primary-paths": {
 "primary-path": [
 {
 "name": "path1",
 "path-metric-bounds": {
 "path-metric-bound": [
 {
 "metric-type": "te-types:path-metric-hop",
 "upper-bound": "3"
 }
]
 }
 }
]
 }
 }
]
}
```

### 13.5. Tunnel State

In this example, the 'GET' query is sent to return the state stored about the tunnel.

```
GET /restconf/data/ietf-te:te/tunnels/tunnel="Example_LSP_Tunnel_A_4_1"
/p2p-primary-paths/ HTTP/1.1
Host: example.com
Accept: application/yang-data+json
```

The request, with status code 200 would include, for example, the following json:

```

{
 "ietf-te: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": "10.0.0.2"
 }
 },
 {
 "index": "2",
 "numbered-node-hop": {
 "node-id": "10.0.0.4"
 }
 }
]
 }
 }
 }
]
 }
 }
],
 "lsp": {
 "lsp": [
 {
 "tunnel-name": "Example_LSP_Tunnel_A_4_1",
 "node": "10.0.0.1 ",
 "lsp-id": "25356"
 }
]
 }
]
}

```

## 14. References

### 14.1. 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>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <<https://www.rfc-editor.org/info/rfc3209>>.
- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, DOI 10.17487/RFC3945, October 2004, <<https://www.rfc-editor.org/info/rfc3945>>.
- [RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, DOI 10.17487/RFC4206, October 2005, <<https://www.rfc-editor.org/info/rfc4206>>.
- [RFC4872] Lang, J.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>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

- [RFC6107] Shiomoto, K., Ed. and A. Farrel, Ed., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, DOI 10.17487/RFC6107, February 2011, <<https://www.rfc-editor.org/info/rfc6107>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6780] Berger, L., Le Faucheur, F., and A. Narayanan, "RSVP ASSOCIATION Object Extensions", RFC 6780, DOI 10.17487/RFC6780, October 2012, <<https://www.rfc-editor.org/info/rfc6780>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7308] Osborne, E., "Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)", RFC 7308, DOI 10.17487/RFC7308, July 2014, <<https://www.rfc-editor.org/info/rfc7308>>.
- [RFC7551] Zhang, F., Ed., Jing, R., and R. Gandhi, Ed., "RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)", RFC 7551, DOI 10.17487/RFC7551, May 2015, <<https://www.rfc-editor.org/info/rfc7551>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <<https://www.rfc-editor.org/info/rfc8639>>.
- [RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <<https://www.rfc-editor.org/info/rfc8641>>.
- [RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <<https://www.rfc-editor.org/info/rfc8776>>.
- [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>>.

#### 14.2. Informative References

- [I-D.ietf-spring-segment-routing-policy]  
Filsfils, C., Talaulikar, K., Voyer, D., Bogdanov, A., and  
P. Mattes, "Segment Routing Policy Architecture", Work in  
Progress, Internet-Draft, draft-ietf-spring-segment-  
routing-policy-16, 28 January 2022,  
<[https://www.ietf.org/archive/id/draft-ietf-spring-  
segment-routing-policy-16.txt](https://www.ietf.org/archive/id/draft-ietf-spring-segment-routing-policy-16.txt)>.
- [I-D.ietf-teas-yang-rsvp]  
Beeram, V. P., Saad, T., Gandhi, R., Liu, X., and I.  
Bryskin, "A YANG Data Model for Resource Reservation  
Protocol (RSVP)", Work in Progress, Internet-Draft, draft-  
ietf-teas-yang-rsvp-17, 9 January 2022,  
<[https://www.ietf.org/archive/id/draft-ietf-teas-yang-  
rsvp-17.txt](https://www.ietf.org/archive/id/draft-ietf-teas-yang-rsvp-17.txt)>.
- [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>>.
- [RFC8800] Litkowski, S., Sivabalan, S., Barth, C., and M. Negi,  
"Path Computation Element Communication Protocol (PCEP)  
Extension for Label Switched Path (LSP) Diversity  
Constraint Signaling", RFC 8800, DOI 10.17487/RFC8800,  
July 2020, <<https://www.rfc-editor.org/info/rfc8800>>.
- [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>>.

#### Authors' Addresses

Tarek Saad  
Juniper Networks  
  
Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Rakesh Gandhi  
Cisco Systems Inc  
  
Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com

Oscar Gonzalez de Dios  
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 10, 2020

T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems Inc  
X. Liu  
Volta Networks  
V. Beeram  
Juniper Networks  
I. Bryskin  
Individual  
March 09, 2020

A YANG Data Model for MPLS Traffic Engineering Tunnels  
draft-ietf-teas-yang-te-mpls-03

Abstract

This document defines a YANG data model for the configuration and management of Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) tunnels, Label Switched Paths (LSPs) and interfaces. The model augments the TE generic YANG model for MPLS packet dataplane technology.

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

Status of This Memo

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

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

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

This Internet-Draft will expire on September 10, 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                            |    |
|--------------------------------------------|----|
| 1. Introduction . . . . .                  | 2  |
| 1.1. Terminology . . . . .                 | 3  |
| 1.2. Prefixes in Data Node Names . . . . . | 3  |
| 1.3. Acronyms and Abbreviations . . . . .  | 3  |
| 2. MPLS TE YANG Model . . . . .            | 3  |
| 2.1. Module(s) Relationship . . . . .      | 4  |
| 2.2. Model Tree Diagram . . . . .          | 4  |
| 2.3. MPLS TE YANG Module . . . . .         | 8  |
| 3. IANA Considerations . . . . .           | 18 |
| 4. Security Considerations . . . . .       | 18 |
| 5. Contributors . . . . .                  | 19 |
| 6. Normative References . . . . .          | 19 |
| Authors' Addresses . . . . .               | 21 |

## 1. Introduction

YANG [RFC6020] and [RFC7950] is a data modeling language used 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 the YANG data model for configuration and management of MPLS TE tunnels, LSPs, and interfaces. Other YANG module(s) that model the establishment of MPLS LSP(s) via signaling protocols such as RSVP-TE ([RFC3209], [RFC3473]) are described in separate document(s).

### 1.1. Terminology

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

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

### 1.2. Prefixes in Data Node Names

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

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

Table 1: Prefixes and corresponding YANG modules

### 1.3. Acronyms and Abbreviations

MPLS: Multiprotocol Label Switching LSP: Label Switched Path LSR: Label Switching Router LER: Label Edge Router TE: Traffic Engineering

## 2. MPLS TE YANG Model

The MPLS TE YANG model covers the configuration, state, RPC and notifications data pertaining to MPLS TE interfaces, tunnels and LSPs parameters. The data specific to the signaling protocol used to establish MPLS LSP(s) is outside the scope of this document and is covered in other documents, e.g. in [I-D.ietf-teas-yang-rsvp] and [I-D.ietf-teas-yang-rsvp-te].



## 2.1. Module(s) Relationship

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

- o ietf-te and ietf-te-device defined in [I-D.ietf-teas-yang-te]
- o ietf-te-types and ietf-te-packet-types defined in [I-D.ietf-teas-yang-te-types]
- o ietf-routing-types defined in [RFC8294]
- o ietf-mpls-static defined in [I-D.ietf-mpls-static-yang]

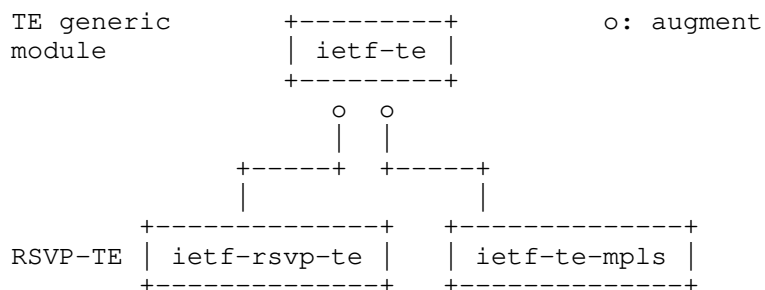


Figure 1: Relationship of MPLS TE module with TE generic and RSVP-TE YANG modules

The MPLS TE YANG module "ietf-te-mpls" augments the "ietf-te" TE generic YANG module as shown in Figure 1.

## 2.2. Model Tree Diagram

Figure 2 shows the tree diagram of the MPLS TE YANG model that is defined in ietf-te-mpls.yang.

```

module: ietf-te-mpls
 augment /te:te/te-dev:performance-thresholds:
 +--rw throttle
 +--rw one-way-delay-offset? uint32
 +--rw measure-interval? uint32
 +--rw advertisement-interval? uint32
 +--rw suppression-interval? uint32
 +--rw threshold-out
 +--rw one-way-delay? uint32
 +--rw one-way-residual-bandwidth?
 | rt-types:bandwidth-ieee-float32
 +--rw one-way-available-bandwidth?

```

```

 | rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw two-way-delay? uint32
+---rw one-way-min-delay? uint32
+---rw one-way-max-delay? uint32
+---rw one-way-delay-variation? uint32
+---rw one-way-packet-loss? decimal64
+---rw two-way-min-delay? uint32
+---rw two-way-max-delay? uint32
+---rw two-way-delay-variation? uint32
+---rw two-way-packet-loss? decimal64
+---rw threshold-in
 +---rw one-way-delay? uint32
 +---rw one-way-residual-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw two-way-delay? uint32
+---rw one-way-min-delay? uint32
+---rw one-way-max-delay? uint32
+---rw one-way-delay-variation? uint32
+---rw one-way-packet-loss? decimal64
+---rw two-way-min-delay? uint32
+---rw two-way-max-delay? uint32
+---rw two-way-delay-variation? uint32
+---rw two-way-packet-loss? decimal64
+---rw threshold-accelerated-advertisement
 +---rw one-way-delay? uint32
 +---rw one-way-residual-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw one-way-available-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw one-way-utilized-bandwidth?
 | rt-types:bandwidth-ieee-float32
+---rw two-way-delay? uint32
+---rw one-way-min-delay? uint32
+---rw one-way-max-delay? uint32
+---rw one-way-delay-variation? uint32
+---rw one-way-packet-loss? decimal64
+---rw two-way-min-delay? uint32
+---rw two-way-max-delay? uint32
+---rw two-way-delay-variation? uint32
+---rw two-way-packet-loss? decimal64
augment /te:te/te:tunnels/te:tunnel:
+---rw tunnel-igp-shortcut

```

```

 | +---rw shortcut-eligible? boolean
 | +---rw metric-type? identityref
 | +---rw metric? int32
 | +---rw routing-afs* inet:ip-version
+---rw forwarding
 | +---rw binding-label? rt-types:mpls-label
 | +---rw load-share? uint32
 | +---rw policy-class? uint8
+---rw bandwidth-mpls
 | +---rw specification-type?
 | | te-packet-types:te-bandwidth-requested-type
 | +---rw set-bandwidth? te-packet-types:bandwidth-kbps
 | +---rw class-type? te-types:te-ds-class
 | +---ro state
 | | +---ro signaled-bandwidth? te-packet-types:bandwidth-kbps
+---rw auto-bandwidth
 | +---rw enabled? boolean
 | +---rw min-bw? te-packet-types:bandwidth-kbps
 | +---rw max-bw? te-packet-types:bandwidth-kbps
 | +---rw adjust-interval? uint32
 | +---rw adjust-threshold? rt-types:percentage
 | +---rw overflow
 | | +---rw enabled? boolean
 | | +---rw overflow-threshold? rt-types:percentage
 | | +---rw trigger-event-count? uint16
 | +---rw underflow
 | | +---rw enabled? boolean
 | | +---rw underflow-threshold? rt-types:percentage
 | | +---rw trigger-event-count? uint16
augment /te:te/te:tunnels/te:tunnel/te:primary-paths/te:primary-path:
 +---rw static-lsp-name? mpls-static:static-lsp-ref
augment /te:te/te:tunnels/te:tunnel/te:secondary-paths
 /te:secondary-path:
 +---rw static-lsp-name? mpls-static:static-lsp-ref
augment /te:te/te:globals/te:named-path-constraints
 /te:named-path-constraint:
 +---rw bandwidth
 | +---rw specification-type?
 | | te-packet-types:te-bandwidth-requested-type
 | +---rw set-bandwidth? te-packet-types:bandwidth-kbps
 | +---rw class-type? te-types:te-ds-class
 | +---ro state
 | | +---ro signaled-bandwidth? te-packet-types:bandwidth-kbps
augment /te:te/te:tunnels/te:tunnel/te:primary-paths/te:primary-path
 /te:lsp/te:lsp:
 +---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 one-way-min-delay? uint32
+---ro one-way-min-delay-normality?
| te-types:performance-metrics-normality
+---ro one-way-max-delay? uint32
+---ro one-way-max-delay-normality?
| te-types:performance-metrics-normality
+---ro one-way-delay-variation? uint32
+---ro one-way-delay-variation-normality?
| te-types:performance-metrics-normality
+---ro one-way-packet-loss? decimal64
+---ro one-way-packet-loss-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 two-way-min-delay? uint32
+---ro two-way-min-delay-normality?
| te-types:performance-metrics-normality
+---ro two-way-max-delay? uint32
+---ro two-way-max-delay-normality?
| te-types:performance-metrics-normality
+---ro two-way-delay-variation? uint32
+---ro two-way-delay-variation-normality?
| te-types:performance-metrics-normality
+---ro two-way-packet-loss? decimal64
+---ro two-way-packet-loss-normality?
| te-types:performance-metrics-normality

```

Figure 2: MPLS TE model configuration and state tree

### 2.3. MPLS TE YANG Module

```
<CODE BEGINS> file "ietf-te-mpls@2020-03-09.yang"
module ietf-te-mpls {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-te-mpls";

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

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

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

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

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

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

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

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

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

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

description
 "YANG data module for MPLS TE configurations,
 state, RPC and notifications. 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 "2020-03-09" {
 description "Latest update to MPLS TE YANG module.";
 reference
 "RFCXXXX: A YANG Data Model for MPLS-TE Tunnels and LSP(s)";
}

/* MPLS TE Identities */
identity tunnel-action-resetup {
 base te:tunnel-actions-type;
 description "Resetup tunnel action type";
}

/* MPLS TE tunnel properties*/
grouping tunnel-igp-shortcut-config {
 description "TE tunnel IGP shortcut configs";
 leaf shortcut-eligible {
 type boolean;
 default "true";
 description
 "Whether this LSP is considered to be eligible for us as a
 shortcut in the IGP. In the case that this leaf is set to
 true, the IGP SPF calculation uses the metric specified to
 determine whether traffic should be carried over this LSP";
 }
 leaf metric-type {
 type identityref {
 base te-types:lsp-metric-type;
 }
 default te-types:lsp-metric-inherited;
 description
 "The type of metric specification that should be used to set
 the LSP(s) metric";
 }
 leaf metric {
 type int32;
 description
 "The value of the metric that should be specified. The value
 supplied in this leaf is used in conjunction with the metric
 type to determine the value of the metric used by the system.
 Where the metric-type is set to lsp-metric-absolute - the
 value of this leaf is used directly; where it is set to
 lsp-metric-relative, the relevant (positive or negative)
 offset is used to formulate the metric; where metric-type
 is lsp-metric-inherited, the value of this leaf is not
 utilized";
 }
}
```

```
 }
 leaf-list routing-afs {
 type inet:ip-version;
 description
 "Address families";
 }
 }

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

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

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



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

 leaf enabled {
 type boolean;
 default false;
 description
 "Enables MPLS auto-bandwidth on the
 LSP";
 }

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

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

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

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

grouping te-lsp-overflow-config {
 description
 "configuration for MPLS LSP bandwidth
```

```
 overflow adjustment";

 leaf enabled {
 type boolean;
 default false;
 description
 "Enables MPLS LSP bandwidth overflow
 adjustment on the LSP";
 }

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

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

grouping te-lsp-underflow-config {
 description
 "configuration for MPLS LSP bandwidth
 underflow adjustment";

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

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

 leaf trigger-event-count {
 type uint16;
 }
}
```

```
 description
 "number of consecutive underflow sample
 events needed to trigger an underflow adjustment";
 }
 }
 grouping te-tunnel-bandwidth-config {
 description
 "Configuration parameters related to bandwidth for a tunnel";

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

 leaf set-bandwidth {
 when "../specification-type = 'specified'" {
 description
 "The bandwidth value when bandwidth is explicitly
 specified";
 }
 type te-packet-types:bandwidth-kbps;
 description
 "set bandwidth explicitly, e.g., using
 offline calculation";
 }

 leaf class-type {
 type te-types:te-ds-class;
 description
 "The Class-Type of traffic transported by the LSP.";
 reference "RFC4124: section-4.3.1";
 }
 }
}

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

 leaf signaled-bandwidth {
 type te-packet-types:bandwidth-kbps;
 description
 "The currently signaled bandwidth of the LSP. In the case where
 the bandwidth is specified explicitly, then this will match the
 value of the set-bandwidth leaf; in cases where the bandwidth is
 dynamically computed by the system, the current value of the
 bandwidth should be reflected.";
 }
}
```

```
 }
 }

 grouping tunnel-bandwidth_top {
 description
 "Top level grouping for specifying bandwidth for a tunnel";

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

 uses te-tunnel-bandwidth-config;

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

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

 uses te-lsp-auto-bandwidth-config;

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

 uses te-lsp-overflow-config;
 }

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

 uses te-lsp-underflow-config;
 }
 }
 }
 }
```

```
 }
 }

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

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

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

 /**
 * MPLS TE augmentations
 */
 augment "/te:te/te-dev:performance-thresholds" {
 uses te-packet-types:performance-metrics-throttle-container-packet;
 description
 "Performance parameters configurable thresholds";
 }

 /* MPLS TE interface augmentations */

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

 /* MPLS TE LSPs augmentations */
 augment "/te:te/te:tunnels/te:tunnel/" +
 "te:primary-paths/te:primary-path" {
 when "/te:te/te:tunnels/te:tunnel" +
```

```
 "/te:primary-paths/te:primary-path" +
 "/te:signaling-type = 'te-types:path-setup-static'" {
 description
 "When the path is statically provisioned";
 }
 description "MPLS TE LSP augmentation";
 leaf static-lsp-name {
 type mpls-static:static-lsp-ref;
 description "Static LSP name";
 }
 }

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

augment "/te:te/te:globals/te:named-path-constraints/" +
 "te:named-path-constraint" {
 description "foo";
 uses te-path-bandwidth_top;
 }

augment "/te:te/te:tunnels/te:tunnel/te:primary-paths" +
 "/te:primary-path/te:lsps/te:lsp" {
 description
 "MPLS TE generic data augmentation pertaining to specific TE
 LSP";
 uses te-packet-types:performance-metrics-attributes-packet;
 }
}
<CODE ENDS>
```

Figure 3: TE generic 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-te-mpls  
XML: N/A, the requested URI is an XML namespace.

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

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

### 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.

A number of data nodes defined in this YANG module 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 MPLS network operations. Following are the subtrees and data nodes and their sensitivity/vulnerability:

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

"/te/globals": The augmentation to this target specifies configuration applicable to the to all or one TE device. Unauthorized access to this list could cause the device to ignore packets it should receive and process.

## 5. Contributors

Himanshu Shah  
Ciena  
Email: hshah@ciena.com

## 6. Normative References

- [I-D.ietf-mpls-static-yang]  
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin,  
"A YANG Data Model for MPLS Static LSPs", draft-ietf-mpls-  
static-yang-10 (work in progress), September 2019.
- [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-12 (work in progress),  
January 2020.
- [I-D.ietf-teas-yang-rsvp-te]  
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I.,  
and H. Shah, "A YANG Data Model for RSVP-TE Protocol",  
draft-ietf-teas-yang-rsvp-te-07 (work in progress), July  
2019.
- [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 and  
Interfaces", draft-ietf-teas-yang-te-22 (work in  
progress), November 2019.
- [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>>.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,  
and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP  
Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001,  
<<https://www.rfc-editor.org/info/rfc3209>>.



- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, DOI 10.17487/RFC3473, January 2003, <<https://www.rfc-editor.org/info/rfc3473>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.

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

#### Authors' Addresses

Tarek Saad  
Juniper Networks

Email: [tsaad@juniper.net](mailto:tsaad@juniper.net)

Rakesh Gandhi  
Cisco Systems Inc

Email: [rgandhi@cisco.com](mailto:rgandhi@cisco.com)

Xufeng Liu  
Volta Networks

Email: [xufeng.liu.ietf@gmail.com](mailto:xufeng.liu.ietf@gmail.com)

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

Igor Bryskin  
Individual

Email: [i\\_bryskin@yahoo.com](mailto:i_bryskin@yahoo.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

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on November 8, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## 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.....                       | 3  |
| 1.1. Terminology.....                      | 4  |
| 1.2. Tree diagram.....                     | 5  |
| 1.3. Prefixes in Data Node Names.....      | 5  |
| 2. Use-Cases.....                          | 5  |
| 3. Design of the Data Models.....          | 7  |
| 3.1. TE KPI Telemetry Model.....           | 7  |
| 3.2. VN KPI Telemetry Model.....           | 8  |
| 4. Autonomic Scaling Intent Mechanism..... | 9  |
| 5. Notification.....                       | 11 |
| 5.1. YANG Push Subscription Examples.....  | 11 |
| 6. YANG Data Tree.....                     | 13 |
| 7. Yang Data Model.....                    | 15 |
| 7.1. ietf-te-kpi-telemetry model.....      | 15 |
| 7.2. ietf-vn-kpi-telemetry model.....      | 21 |
| 8. Security Considerations.....            | 25 |
| 9. IANA Considerations.....                | 26 |

|                                   |    |
|-----------------------------------|----|
| 10. Acknowledgements.....         | 27 |
| 11. References.....               | 27 |
| 11.1. Normative References.....   | 27 |
| 11.2. Informative References..... | 28 |
| 12. Contributors.....             | 29 |
| Authors' Addresses.....           | 29 |

## 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 interchangeable 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.

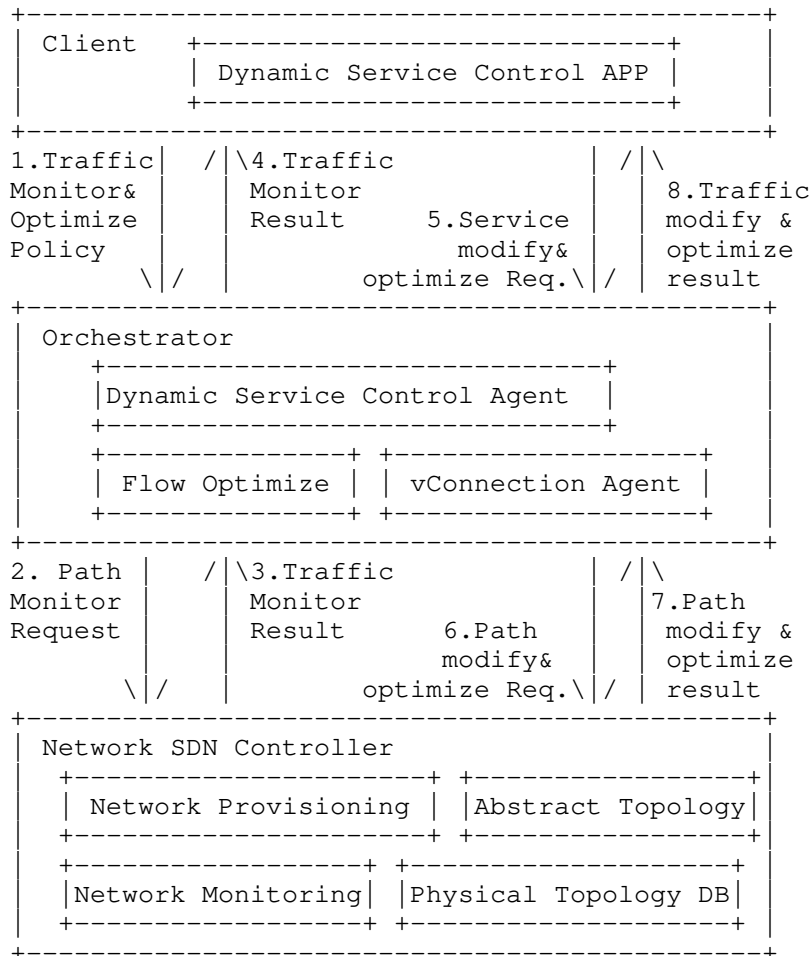


Figure 1 Workflows for dynamic service control based on traffic monitoring

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:

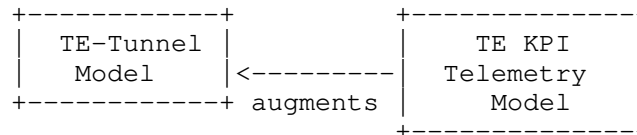
- (i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & 7.1 for details).
- (ii) VN KPI Telemetry Model which provides the VN level of the aggregated performance 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

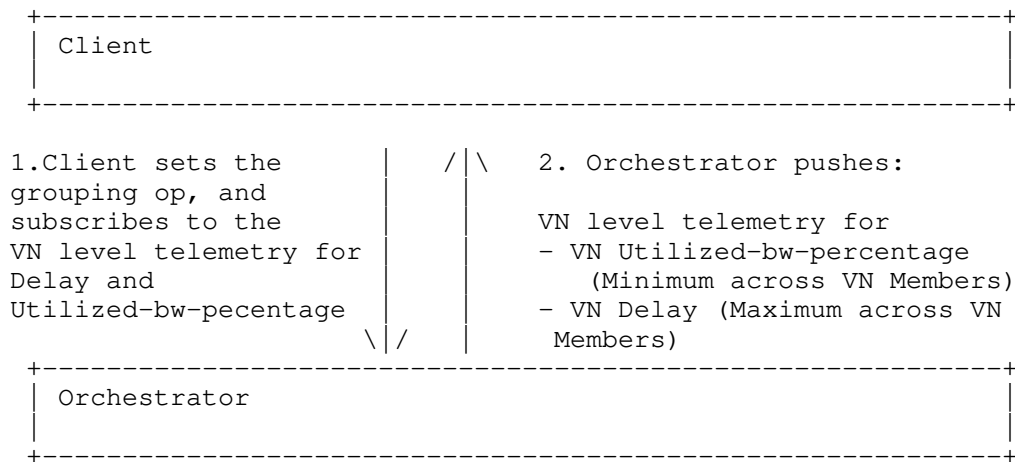
will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.



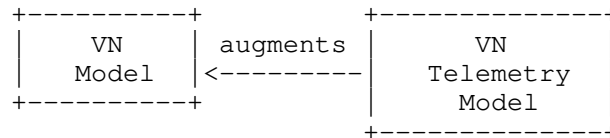
### 3.2. VN KPI Telemetry Model

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:



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.



#### 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 analyze 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.
 Defaults 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 analyze 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  
Category: Informational

Haomian Zheng  
Xianlong Luo  
Huawei Technologies  
Yang Zhao  
China Mobile  
Yunbin Xu  
CAICT  
Sergio Belotti  
Dieter Beller  
Nokia  
February 15, 2019

Expires: August 19, 2019

## Interworking of GMPLS Control and Centralized Controller System

draft-zheng-teas-gmpls-controller-inter-work-03

### Abstract

Generalized Multi-Protocol Label Switching (GMPLS) control allows each network element (NE) to perform local resource discovery, routing and signaling in a distributed manner.

On the other hand, with the development of software-defined transport networking technology, a set of NEs can be controlled via centralized controller hierarchies to address the issue from multi-domain, multi-vendor and multi-technology. An example of such centralized architecture is ACTN controller hierarchy described in RFC 8453.

Instead of competing with each other, both the distributed and the centralized control plane have their own advantages, and should be complementary in the system. This document describes how the GMPLS distributed control plane can interwork with a centralized controller system in a transport network.

### Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 19, 2019.

#### Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Conventions used in this document

#### Table of Contents

|                                                                  |   |
|------------------------------------------------------------------|---|
| 1. Introduction .....                                            | 3 |
| 2. Overview .....                                                | 4 |
| 2.1. Overview of GMPLS Control Plane.....                        | 4 |
| 2.2. Overview of Centralized Controller System .....             | 4 |
| 2.3. GMPLS Control Interwork with Centralized Controller System. | 4 |
| 3. Link Management Protocol .....                                | 6 |
| 4. Routing Options .....                                         | 6 |
| 4.1. OSPF-TE .....                                               | 6 |
| 4.2. ISIS-TE .....                                               | 6 |
| 4.3. Netconf/RESTconf .....                                      | 7 |
| 5. Path Computation .....                                        | 7 |
| 5.1. Constraint-based Path Computing in GMPLS Control.....       | 7 |
| 5.2. Path Computation Element (PCE) .....                        | 7 |
| 6. Signaling Options .....                                       | 8 |
| 6.1. RSVP-TE .....                                               | 8 |
| 7. Interworking Scenarios.....                                   | 8 |



|                                                    |    |
|----------------------------------------------------|----|
| 7.1. Topology Collection & Synchronization .....   | 8  |
| 7.2. Multi-domain/layer Service Provisioning ..... | 9  |
| 7.3. Recovery .....                                | 9  |
| 7.4. Controller Reliability .....                  | 10 |
| 8. Manageability Considerations.....               | 10 |
| 9. Security Considerations .....                   | 10 |
| 10. IANA Considerations.....                       | 10 |
| 11. References .....                               | 10 |
| 11.1. Normative References.....                    | 10 |
| 11.2. Informative References.....                  | 12 |
| 12. Authors' Addresses .....                       | 14 |

## 1. Introduction

Generalized Multi-Protocol Label Switching (GMPLS) [RFC3945] extends MPLS to support different classes of interfaces and switching capabilities such as Time-Division Multiplex Capable (TDM), Lambda Switch Capable (LSC), and Fiber-Switch Capable (FSC). Each network element (NE) running a GMPLS control plane collects network information from other NEs and supports service provisioning through signaling in a distributed manner. More generic description for Traffic-engineering networking information exchange can be found in [RFC7926].

On the other hand, Software-Defined Networking (SDN) technologies have been introduced to control the transport network in a centralized manner. Central controllers can collect network information from each node and provision services to corresponding nodes. One of the examples is the Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453], which defines a hierarchical architecture with Provisioning Network Controller(PNC), Multi-domain Service Coordinator(MDSC) and Customer Network Controller(CNC) as central controllers for different network abstraction levels. A Path Computation Element (PCE) based approach has been proposed as Application-Based Network Operations (ABNO) in [RFC7491].

In such centralized controller architectures, GMPLS can be applied for the NE-level control. A central controller may support GMPLS enabled domains and may interact with a GMPLS enabled domain where the GMPLS control plane does the service provisioning from ingress to egress. In this case the centralized controller sends the request to the ingress node and does not have to configure all NEs along the path through the domain from ingress to egress thus leveraging the GMPLS control plane. This document describes how GMPLS control interworks with centralized controller system in transport network.



## 2. Overview

In this section, overviews of GMPLS control plane and centralized controller system are discussed as well as the interactions between the GMPLS control plane and centralized controllers.

### 2.1. Overview of GMPLS Control Plane

GMPLS separates the control plane and the data plane to support time-division, wavelength, and spatial switching, which are significant in transport networks. For the NE level control in GMPLS, each node runs a GMPLS control plane instance. Functionalities such as service provisioning, protection, and restoration can be performed via GMPLS communication among multiple NEs. At the same time, the controller can also collect node and link resources in the network to construct the network topology and compute routing paths for serving service requests.

Several protocols have been designed for GMPLS control [RFC3945] including link management [RFC4204], signaling [RFC3471], and routing [RFC4202] protocols. The controllers applying these protocols communicate with each other to exchange resource information and establish Label Switched Paths (LSPs). In this way, controllers in different nodes in the network have the same view of the network topology and provision services based on local policies.

### 2.2. Overview of Centralized Controller System

With the development of SDN technologies, a centralized controller architecture has been introduced to transport networks. One example architecture can be found in ACTN [RFC8453]. In such systems, a controller is aware of the network topology and is responsible for provisioning incoming service requests.

Multiple hierarchies of controllers are designed at different levels implementing different functions. This kind of architecture enables multi-vendor, multi-domain, and multi-technology control. For example, an higher-level controller coordinates several lower-level controllers controlling different domains, for topology collection and service provisioning. Vendor-specific features can be abstracted between controllers, and standard API (e.g., generated from RESTconf/YANG) is used.

### 2.3. GMPLS Control Interwork with Centralized Controller System

Besides the GMPLS and the interactions among the controller hierarchies, it is also necessary for the controllers to communicate with the network elements. Within each domain, GMPLS control can be applied to each NE. The bottom-level central controller can act as a NE to collect network information and initiate LSP. Figure 1 shows





an example of GMPLS interworking with centralized controllers (ACTN terminologies are used in the figure).

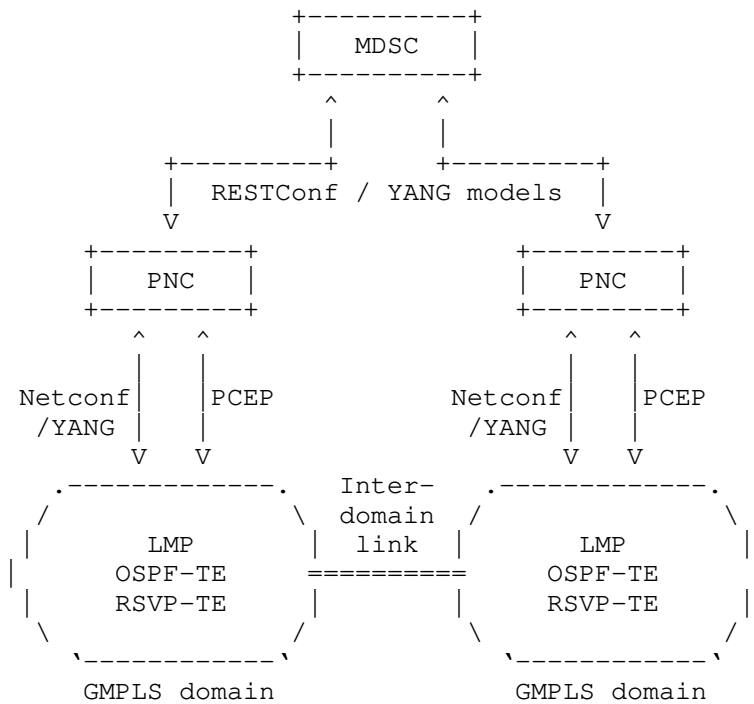


Figure 1: Example of GMPLS interworks with Controllers

In Figure 1, each domain has the GMPLS control plane enabled at the physical network level. The PNC can exploit GMPLS capability implemented in the domain to listen to the IGP routing protocol messages (OSPF LSAs for example) that the GMPLS control plane instances are disseminating into the network and thus learn the network topology. For path computation in the domain with PNC implementing a PCE, PCCs (e.g. NEs, other controller/PCE) use PCEP to ask the PNC for a path and get replies. The MDSC communicates with PNCs using for example REST/RESTConf based on YANG data models. As a PNC has learned its domain topology, it can report the topology to the MDSC. When a service arrives, the MDSC computes the path and coordinates PNCs to establish the corresponding LSP segment.

Alternatively, the NETCONF protocol can be used to retrieve topology information utilizing the e.g. [TE-topo] Yang model and the technology-specific YANG model augmentations required for the



specific network technology. The PNC can retrieve topology information from any NE (the GMPLS control plane instance of each NE in the domain has the same topological view), construct the topology of the domain and export an abstracted view to the MDSC. Based on the topology retrieved from multiple PNCs, the MDSC can create topology graph of the multi-domain network, and can use it for path computation. To setup a service, the MDSC can exploit e.g. [TE-Tunnel] Yang model together with the technology-specific YANG model augmentations.

### 3. Link Management Protocol

Link management protocol (LMP) [RFC4204] runs between a pair of nodes and is used to manage TE links. In addition to the setup and maintenance of control channels, LMP can be used to verify the data link connectivity and correlate the link property. In this way, link resources, which are fundamental resources in the network, are discovered by both ends of the link.

### 4. Routing Options

In GMPLS control, link state information is flooded within the network as defined in [RFC4202]. Each node in the network can build the network topology according to the flooded link state information. Routing protocols such as OSPF-TE [RFC4203] and ISIS-TE [RFC5307] have been extended to support different interfaces in GMPLS.

In centralized controller system, central controller can be placed at the GMPLS network and passively receive the information flooded in the network. In this way, the central controller can construct and update the network topology.

#### 4.1. OSPF-TE

OSPF-TE is introduced for TE networks in [RFC3630]. OSPF extensions have been defined in [RFC4203] to enable the capability of link state information for GMPLS network. Based on this work, OSPF protocol has been extended to support technology-specific routing. The routing protocol for OTN, WSON and optical flexi-grid network are defined in [RFC7138], [RFC7688] and [RFC8363], respectively.

#### 4.2. ISIS-TE

ISIS-TE is introduced for TE networks in [RFC5305] and is extended to support GMPLS routing functions [RFC5307], and has been updated to [RFC7074] to support the latest GMPLS switching capability and Types fields.



#### 4.3. Netconf/RESTconf

Netconf [RFC6241] and RESTconf [RFC8040] protocols are originally used for network configuration. Besides, these protocols can also be used for topology retrieval by using topology-related YANG models, such as [RFC8345] and [TE-topo]. These protocols provide a powerful mechanism for notification that permits to notify the client about topology changes.

#### 5. Path Computation

Once a controller learns the network topology, it can utilize the available resources to serve service requests by performing path computation. Due to abstraction, the controllers may not have sufficient information to compute the optimal path. In this case, the controller can interact with other controllers by sending Yang Path Computation requests [PAT-COMP] to compute a set of potential optimal paths and then, based on its own constraints, policy and specific knowledge (e.g. cost of access link) can choose the more feasible path for service e2e path setup.

Path computation is one of the key objectives in various types of controllers. In the given architecture, it is possible for different components that have the capability to compute the path.

##### 5.1. Constraint-based Path Computing in GMPLS Control

In GMPLS control, a routing path is computed by the ingress node [RFC3473] and is based on the ingress node TED. Constraint-based path computation is performed according to the local policy of the ingress node.

##### 5.2. Path Computation Element (PCE)

PCE has been introduced in [RFC4655] as a functional component that provides services to compute path in a network. In [RFC5440], the path computation is accomplished by using the Traffic Engineering Database (TED), which maintains the link resources in the network. The emergence of PCE efficiently improve the quality of network planning and offline computation, but there is a risk that the computed path may be infeasible if there is a diversity requirement, because stateless PCE has no knowledge about the former computed paths.

To address this issue, stateful PCE has been proposed in [RFC8231]. Besides the TED, an additional LSP Database (LSP-DB) is introduced to archive each LSP computed by the PCE. In this way, PCE can easily figure out the relationship between the computing path and former computed paths. In this approach, PCE provides computed paths to



PCC, and then PCC decides which path is deployed and when to be established.

In PCE Initiation [RFC8281], PCE is allowed to trigger the PCC to setup, maintenance, and teardown of the PCE-initiated LSP under the stateful PCE model. This would allow a dynamic network that is centrally controlled and deployed.

In centralized controller system, the PCE can be implemented in a central controller, and the central controller performs path computation according to its local policies. On the other hand, the PCE can also be placed outside of the central controller. In this case, the central controller acts as a PCC to request path computation to the PCE through PCEP. One of the reference architecture can be found at [RFC7491].

## 6. Signaling Options

Signaling mechanisms are used to setup LSPs in GMPLS control. Messages are sent hop by hop between the ingress node and the egress node of the LSP to allocate labels. Once the labels are allocated along the path, the LSP setup is accomplished. Signaling protocols such as RSVP-TE [RFC3473] have been extended to support different interfaces in GMPLS.

### 6.1. RSVP-TE

RSVP-TE is introduced in [RFC3209] and extended to support GMPLS signaling in [RFC3473]. Several label formats are defined for a generalized label request, a generalized label, suggested label and label sets. Based on [RFC3473], RSVP-TE has been extended to support technology-specific signaling. The RSVP-TE extensions for OTN, WSON, optical flexi-grid network are defined in [RFC7139], [RFC7689], and [RFC7792], respectively.

## 7. Interworking Scenarios

### 7.1. Topology Collection & Synchronization

Topology information is necessary on both network elements and controllers. The topology on network element is usually raw information, while the topology on the controller can be either raw or abstracted. Three different abstraction methods have been described in [RFC8453], and different controllers can select the corresponding method depending on application.

When there are changes in the network topology, the impacted network element(s) need to report changes to all the other network elements, together with the controller, to sync up the topology information. The inter-NE synchronization can be achieved via protocols mentioned





in section 3 and 4. The topology synchronization between NEs and controllers can either be achieved by routing protocols OSPF-TE/PCEP-LS in [PCEP-LS] or Netconf protocol notifications with YANG model.

## 7.2. Multi-domain/layer Service Provisioning

Based on the topology information on controllers and network elements, service provisioning can be deployed. Plenty of methods have been specified for single domain service provisioning, such as using PCEP and RSVP-TE.

Multi-domain/layer service provisioning would request coordination among the controller hierarchies. Given the service request, the end-to-end delivery procedure may include interactions at any level (i.e. interface) in the hierarchy of the controllers (e.g. MPI and SBI for ACTN). The computation for a cross-domain/layer path is usually completed by controllers who have a global view of the topologies. Then the configuration is decomposed into lower layer controllers, to configure the network elements to set up the path.

A combination of the centralized and distributed protocols may be necessary for the interaction between network elements and controller. A typical example would be the PCE Initiation scenario, in which a PCE message (PCInitiate) is sent from the controller to the first-end node, and then trigger a RSVP procedure along the path. Similarly, the interaction between the controller and the ingress node of a domain can be achieved by Netconf protocol with corresponding YANG models, and then completed by running RSVP among the network elements.

## 7.3. Recovery

The GMPLS recovery functions are described in [RFC4426]. Two models, span protection and end-to-end protection and restoration, are discussed with different protection schemes and message exchange requirements. Related RSVP-TE extensions to support end-to-end recovery is described in [RFC4872]. The extensions in [RFC4872] include protection, restoration, preemption, and rerouting mechanisms for an end-to-end LSP. Besides end-to-end recovery, a GMPLS segment recovery mechanism is defined in [RFC4873]. By introducing secondary record route objects, LSP segment can be switched to another path like fast reroute [RFC4090].

For the recovery with controllers, timely interaction between controller and network elements are required. Usually the re-routing can be decomposed into path computation and delivery, the controller can take some advantage in the path computation due to the global topology view. And the delivery can be achieved by the procedure described in section 7.2.



#### 7.4. Controller Reliability

Given the important role in the network, the reliability of controller is critical. Once a controller is shut down, the network should operate as well. It can be either achieved by controller back up or functionality back up. There are several of controller backup or federation mechanisms in the literature. It is also more reliable to have some function back up in the network element, to guarantee the performance in the network.

#### 8. Manageability Considerations

Each entity in the network, including both controllers and network elements, should be managed properly as it will interact with other entities. The manageability considerations in controller hierarchies and network elements still apply respectively. For the protocols applied in the network, manageability is also requested.

The responsibility of each entity should be clarified. The control of function and policy among different controllers should be consistent via proper negotiation process.

#### 9. Security Considerations

This document provides the interwork between the GMPLS and controller hierarchies. The security requirements in both system still applies respectively. Protocols referenced in this document also have various security considerations, which is also expected to be satisfied.

Other considerations on the interface between the controller and the network element are also important. Such security includes the functions to authenticate and authorize the control access to the controller from multiple network elements. Security mechanisms on the controller are also required to safeguard the underlying network elements against attacks on the control plane and/or unauthorized usage of data transport resources.

#### 10. IANA Considerations

This document requires no IANA actions.

#### 11. References

##### 11.1. Normative References

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



- [RFC3473] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.
- [RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.
- [RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, October 2005.
- [RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [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, May 2007.
- [RFC4873] Berger, L., Bryskin, I., Papadimitriou, D., and A. Farrel, "GMPLS Segment Recovery", RFC 4873, May 2007.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, October 2008.
- [RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, October 2008.
- [RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder J., Bierman A., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC7074] Berger, L. and J. Meuric, "Revised Definition of the GMPLS Switching Capability and Type Fields", RFC 7074, November 2013.
- [RFC7491] King, D., Farrel, A., "A PCE-Based Architecture for Application-Based Network Operations", RFC7491, March 2015.



- [RFC7926] Farrel, A., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D. and Zhang, X., "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", RFC7926, July 2016.
- [RFC8040] Bierman, A., Bjorklund, M., Watsen, K., "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8453] Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

## 11.2. Informative References

- [RFC3471] Berger, L., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.
- [RFC4204] Lang, J., Ed., "Link Management Protocol (LMP)", RFC 4204, October 2005.
- [RFC4426] Lang, J., Ed., Rajagopalan, B., Ed., and D. Papadimitriou, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification", RFC 4426, March 2006.
- [RFC7138] Ceccarelli, D., Ed., Zhang, F., Belotti, S., Rao, R., and J. Drake, "Traffic Engineering Extensions to OSPF for GMPLS Control of Evolving G.709 Optical Transport Networks", RFC 7138, March 2014.
- [RFC7139] Zhang, F., Ed., Zhang, G., Belotti, S., Ceccarelli, D., and K. Pithewan, "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.





- [RFC7688] Lee, Y., Ed. and G. Bernstein, Ed., "GMPLS OSPF Enhancement for Signal and Network Element Compatibility for Wavelength Switched Optical Networks", RFC 7688, November 2015.
- [RFC7689] Bernstein, G., Ed., Xu, S., Lee, Y., Ed., Martinelli, G., and H. Harai, "Signaling Extensions for Wavelength Switched Optical Networks", RFC 7689, November 2015.
- [RFC7792] Zhang, F., Zhang, X., Farrel, A., Gonzalez de Dios, O., and D. Ceccarelli, "RSVP-TE Signaling Extensions in Support of Flexi-Grid Dense Wavelength Division Multiplexing (DWDM) Networks", RFC 7792, March 2016.
- [RFC8231] Crabbe, E., Minei, I., Medved, J., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE", RFC 8231, September 2017.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", RFC 8281, October 2017.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., Liu, X., "A YANG Data Model for Network Topologies", RFC 8345, March 2018.
- [RFC8363] Zhang, X., Zheng, H., Casellas, R., Dios, O., and D. Ceccarelli, "GMPLS OSPF-TE Extensions in support of Flexi-grid DWDM networks", RFC8363, February 2017.
- [TE-topo] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., Gonzalez De Dios, O., "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-19, work in progress.
- [PAT-COMP] Busi, I., Belotti, S., Lopez, V., Gonzalez de Dios, O., Sharma, A., Shi, Y., Vilalta, R., Setheraman, K., "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-04, work in progress.
- [PCEP-LS] Dhody, D., Lee, Y., Ceccarelli, D., "PCEP Extensions for Distribution of Link-State and TE Information", draft-dhodylee-pce-pcep-ls, work in progress.
- [TE-Tunnel] Saad, T. et al., "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te, work in progress.



## 12. Authors' Addresses

Haomian Zheng  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base,  
Bantian, Longgang District,  
Shenzhen 518129 P.R.China  
Email: zhenghaomian@huawei.com

Xianlong Luo  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base,  
Bantian, Longgang District,  
Shenzhen 518129 P.R.China  
Email: luoxianlong@huawei.com

Yunbin Xu  
CAICT  
Email: xuyunbin@rit.cn

Yang Zhao  
China Mobile  
Email: zhaoyangyjy@chinamobile.com

Sergio Belotti  
Nokia  
Email: sergio.belotti@nokia.com

Dieter Beller  
Nokia  
Email: Dieter.Beller@nokia.com





TEAS WG  
Internet-Draft  
Updates: 4875 (if approved)  
Intended status: Standards Track  
Expires: August 2, 2019

Z. Zhang  
A. Deshmukh  
R. Singh  
Juniper Networks  
January 29, 2019

RSVP-TE P2MP Tunnels on RMR  
draft-zzhang-teas-rmr-rsvp-p2mp-02

Abstract

This document specifies the optimization in RSVP-TE P2MP tunnel signaling over Resilient MPLS Rings (RMR).

Requirements Language

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

Status of This Memo

This Internet-Draft 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 2, 2019.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                      |   |
|--------------------------------------|---|
| 1. Introduction . . . . .            | 2 |
| 2. Specification . . . . .           | 4 |
| 2.1. RMR Object . . . . .            | 4 |
| 2.2. Procedures . . . . .            | 4 |
| 2.2.1. PATH Message/State . . . . .  | 4 |
| 2.2.2. RESV Message/State . . . . .  | 5 |
| 3. Security Considerations . . . . . | 5 |
| 4. Acknowledgements . . . . .        | 6 |
| 5. Normative References . . . . .    | 6 |
| Authors' Addresses . . . . .         | 6 |

## 1. Introduction

Traditional RSVP-TE P2MP tunnel signaling could be quite involving. With RMR, this could be significantly simplified:

There is no need for ERO/RRO/SERO/SRRO or hop by hop routing. The tunnel ingress simply sends PATH messages in one or both directions of the ring, depending on how leaves are best reached. The <S2L Sub-LSP Descriptor List> only needs to list the tunnel leaves, and a transit router does not need to "branch" a PATH message into multiple ones. Therefore, unless there are many tunnel leaves on a huge ring, a single PATH message is enough. In the rare situation of a large tunnel with many leaves to list, a small number of PATH messages should suffice. Additionally, there is no need to signal and maintain individual sub-LSPs (one for each leaf) any more. As a result, corresponding PATH/RESV state is also reduced. Each node only needs to maintain a single PATH state and a single RESV state for each P2MP tunnel, and the RESV state does not need to track individual leaves - it just need to track if a RESV is received from downstream and/or if this node itself is a leaf.

A RESV message is triggered to the PHOP when the RESV state is first created (either because the node is a leaf or because a RESV message is received from downstream) and it is refreshed periodically. A RESV Tear is sent when the RESV state is deleted (when the node is no longer a Leaf and the RESV from downstream has timed out or a RESV Tear is received).

Optionally, the tunnel ingress may not need to list any/all leaves. It could simply send the PATH message around the ring, with the <S2L



Sub-LSP Descriptor List> listing the root itself. Through methods outside the scope of this document, a node determines if it is a leaf of the tunnel, and if yes, it will send back a RESV message. With this, a single PATH message is surely enough.

In this document, leaves in <S2L Sub-LSP Descriptor List> are referred to as explicit leaves, and leaves not listed there but self-determined by ring nodes are referred to as implicit leaves. There could be both explicit and implicit leaves for a tunnel. The ingress allows implicit leaves by including itself as the last one in the <S2L Sub-LSP Descriptor List>.

Optionally, the RESV message could also include a <S2L Sub-LSP Descriptor List> to list all the leaves on the established tunnel so that the each node knows its downstream leaves. In that case, when the set of downstream leaves changes, a RESV message with the new <S2L Sub-LSP Descriptor List> is triggered.

Adding/removing explicit leaves is straightforward. The ingress simply sends a triggered PATH message with new <S2L Sub-LSP Descriptor List>. As it passes around the ring, each node determines if it is an explicit leaf and updates its state accordingly. The triggered PATH message does not have to go all the way to the last leaf - if on a node the <S2L Sub-LSP Descriptor List> in the would-be-sent PATH message is the same as what was sent before, the triggered PATH message will not be sent further.

To indicate that the tunnel signaling is with above mentioned RMR optimizations, a new object is included in the PATH message to specify the Ring ID and direction.

Link/Node protection is achieved by tunneling packets to the next node using the Ring LSP to that node in the other direction. This does not need any additional signaling but is based on a reasonable premise that unicast Ring LSPs are always in place. Once the ingress learns the failure (through IGP discovery or through other error detection/notification mechanisms), global repair kicks in to reach some leaves via PATH message sent in the other direction. Before global repair is finished, traffic continues to flow in the original path except that at the failure point it is tunneled to the next node.

If an RMR is just part of a general RSVP network the optimization can also be applied on the ring nodes. If the tunnel ingress knows the leaves that are on the ring, it could put all those leaves in the single PATH message and construct the ERO/SERO only towards the entry point on the ring. The entry point then includes the RMR object in the PATH messages that it sends. For leaves beyond the ring, the

ingress may include the exit points on the ring as loose hops in the ERO/SERO, and when a ring node needs to send the PATH message off the ring, it removes the RMR object. Details will be provided in future revisions of this document.

## 2. Specification

## 2.1. RMR Object

The RMR object is a new object of the following:

- o Class Name: RMR
- o Class-Num: TBA1 (to be assigned by IANA)
- o C-Type: TBA2 (to be assigned by IANA)

The format of the object content following the common object header is the following:

[illegible]

Following the 4-octet Ring ID, there is an 8-bit Flags field. The first bit of the Flags field indicates the direction. If it is set, it is clockwise direction. Otherwise, it is anti-clockwise.

## 2.2. Procedures

This section describes the differences in the procedures for ring nodes to set up RSVP-TE P2MP tunnels across the ring, compared to the conventional non-RMR-aware case. For now it is assumed that all nodes (ingress, transit, and leaves) on the tunnel are on the ring.

More details will be provided in future revisions.

### 2.2.1. PATH Message/State

The tunnel ingress includes the RMR object with the Ring ID and the direction flag bit set accordingly. The explicit tunnel leaves are encoded in the <S2L Sub-LSP Descriptor List>, and no ERO/SERO is included. If the tunnel allows implicit leaves, the descriptor list encodes the ingress itself as the last element. The message is sent

to the next node on the ring in the direction specified in the RMR object, w/o using ERO/SERO or hop-by-hop routing.

When a node receives a PATH message with the RMR object, it checks if itself is listed in the <S2L Sub-LSP Descriptor List>, or if the <S2L Sub-LSP Descriptor List> encodes the tunnel ingress as the last element and this node itself is an implicit leaf. If yes, it creates corresponding RESV state and sends a RESV message to the PHOP.

The receiving node removes itself from the <S2L Sub-LSP Descriptor List> in the PATH message, and saves the list locally. The PATH message is sent to the next node on the ring in the specified direction with the saved <S2L Sub-LSP Descriptor List>, if one of the following conditions is met:

- o The <S2L Sub-LSP Descriptor List> encodes the tunnel ingress itself as the last element.
- o The <S2L Sub-LSP Descriptor List> is not empty and either the PATH state is newly created or the <S2L Sub-LSP Descriptor List> is different from the previously saved one.

If <S2L Sub-LSP Descriptor List> is empty and different from the previously saved one, a PATH Teardown is sent instead.

#### 2.2.2. RESV Message/State

A ring node may know that it is a leaf when the PATH message is first processed as described in the previous section. In case of implicit leaves, it may become a leaf after the PATH messages has been processed. A non-leaf node may also receive a RESV message from its NHOP. In all these cases, the node creates RESV state and sends a RESV message to the PHOP, w/o encoding RRO/SRRO.

If a ring node was a leaf but stops being a leaf, either because it is no longer listed in the <S2L Sub-LSP Descriptor List> or it is no longer an implicit leaf, it removes/updates corresponding local state. A RESV Teardown is sent to the PHOP if there is no RESV received from its downstream either.

### 3. Security Considerations

This document does not introduce new security risks.

## 4. Acknowledgements

## 5. Normative References

- [I-D.ietf-mpls-rmr]  
Kompella, K. and L. Contreras, "Resilient MPLS Rings",  
draft-ietf-mpls-rmr-09 (work in progress), January 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>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S.  
Yasukawa, Ed., "Extensions to Resource Reservation  
Protocol - Traffic Engineering (RSVP-TE) for Point-to-  
Multipoint TE Label Switched Paths (LSPs)", RFC 4875,  
DOI 10.17487/RFC4875, May 2007,  
<<https://www.rfc-editor.org/info/rfc4875>>.

## Authors' Addresses

Zhaohui Zhang  
Juniper Networks

EMail: [zzhang@juniper.net](mailto:zzhang@juniper.net)

Abhishek Deshmukh  
Juniper Networks

EMail: [adeshmukh@juniper.net](mailto:adeshmukh@juniper.net)

Ravi Singh  
Juniper Networks

EMail: [ravis@juniper.net](mailto:ravis@juniper.net)