

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: May 7, 2020

J. Dong  
Z. Li  
Huawei Technologies  
F. Qin  
China Mobile  
November 4, 2019

Control Plane Considerations for Enhanced VPN  
draft-dong-teas-enhanced-vpn-control-plane-00

Abstract

Enhanced VPN (VPN+) is an enhancement to VPN services to support the needs of new applications, particularly including the applications that are associated with 5G services. An enhanced VPN may be used for 5G transport network slicing, and will also be of use in more generic scenarios. [I-D.ietf-teas-enhanced-vpn] describes the framework and candidate component technologies for providing enhanced VPN services. This document describes the control plane requirements, functions and considerations to enable VPN+ services. Specifically, the scalability of control plane is analyzed, and the proposed optimization mechanisms are described.

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 RFC 2119 [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 May 7, 2020.

## 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. Requirements on Control Plane . . . . .	3
2.1. Support of Isolation . . . . .	3
2.1.1. Data Plane Isolation . . . . .	3
2.1.2. Control Plane Isolation . . . . .	4
2.2. Attributes of Network Slice . . . . .	5
2.3. Number of Network Slices . . . . .	5
3. Control Plane Functions . . . . .	6
3.1. Distributed Control Plane . . . . .	6
3.2. Centralized Controller . . . . .	7
4. Scalability Considerations . . . . .	8
4.1. Distributed Control Plane . . . . .	8
4.2. Centralized Control Plane . . . . .	9
5. Optimization Mechanisms . . . . .	9
6. IANA Considerations . . . . .	10
7. Security Considerations . . . . .	10
8. Acknowledgements . . . . .	10
9. References . . . . .	10
9.1. Normative References . . . . .	10
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	12

## 1. Introduction

Driven largely by needs arising from the 5G mobile network, the concept of network slicing has gained traction [TS28530]. Network slicing requires to partition the physical network to several pieces to provide each network slice with the required networking, computing, and storage resources and functions to meet the requirement of slice tenants. As specified in [I-D.ietf-teas-enhanced-vpn], a transport network slice is a virtual

(logical) network with a particular network topology and a set of shared or dedicated network resources, which are used to provide the network slice consumer with the required connectivity, appropriate isolation and specific Service Level Agreement (SLA).

Virtual private networks (VPNs) have served the industry well as a means of providing different tenants with logically separated networks in a common network. Basically the VPN service is provided with two network layers: the overlay and the underlay. The underlay is responsible for establishing the network paths based on the network infrastructure, and managing the network resources to meet the requirement of overlay. The overlay is used to distribute the membership and reachability information of different tenants, and provide separation of services between tenants.

The enhanced VPN service (VPN+) [I-D.ietf-teas-enhanced-vpn] is targeted at new applications which require better isolation from both control plane and data plane's perspective and have more stringent performance requirements than can be provided with existing overlay VPNs. To meet the requirement of enhance VPN services, a number of virtual networks need be created, each with a subset of the underlay network topology and a set of network resources allocated to meet the requirement of a specific enhanced VPN or a group of enhanced VPNs. In the context of 5G, each virtual network is considered as a transport network slice.

This document gives analysis to the control plane requirements, functions and considerations to enable enhanced VPN service. The focus of this document is on the underlay of the enhanced VPN, i.e. the transport network slice.

## 2. Requirements on Control Plane

### 2.1. Support of Isolation

#### 2.1.1. Data Plane Isolation

Isolation in data plane is the fundamental requirement of services which are deployed in a shared network infrastructure. Depends on the level of data plane isolation, the requirement can be categorized as soft isolation and hard isolation [I-D.ietf-teas-enhanced-vpn].

Soft isolation means that traffic of one application or tenant cannot be received or inspected by any other application or tenant in the same network. Usually soft isolation does not have strict resource or performance requirement, the underlying network resource can be shared by multiple applications or tenants, which is useful to achieve better economy with statistical multiplexing. However, with

soft isolation, when service in one of the virtual networks experience some event such as traffic burst or congestion, this may result in negative impacts to other virtual networks in terms of packet loss, delay and jitter, etc.

On the other hand, hard isolation means that any event happened to the traffic of one application or tenant in one virtual network will not interfere any other application or tenant in the same network, which means the characteristics of service can be guaranteed or more predictable. To achieve this, at least some of the network resource need to be dedicated, which may reduce the economy of multiplexing to some extent. Hard isolation is required by services that usually have their own private networks and expect to have the same network characteristics even in a shared network.

It is expected that the requirement of some services or tenants can be met with soft isolation, while hard isolation is required for services or tenants which require guaranteed or more predictable performance.

Although the soft and hard isolation characteristics are provided by the forwarding plane of network devices, the control plane needs to be aware of the data plane capability and provide necessary support for both soft and hard isolation. Specifically, network information needed for both soft and hard isolation needs to be collected and distributed in the network, and the route and path computation should be performed based the collected information to generate the forwarding entries for each virtual network.

#### 2.1.2. Control Plane Isolation

From routing's perspective, isolation in control plane can be achieved in different levels: isolation of routing database, and isolation of routing instances.

Isolation of routing database can enable customized routing and TE attributes for different virtual networks. This can be used to generate customized virtual network topologies and compute customized paths for different applications or tenants. The Multi-Topology Routing (MTR) mechanisms [RFC4915] [RFC5120] provides the basic functionality to define separated topology and routing database for different virtual networks. MTR was not widely used in current network due to lack of use cases and some constraints in IP forwarding, but it can be considered as a candidate technology for enhanced VPN, especially when used with data plane technologies such as Segment Routing (SR) [RFC8402]. There are also emerging technologies, such as Flex-Algo as described in

[I-D.ietf-lsr-flex-algo], which can provide customization of the topology and attributes for constraint route computation.

Isolation of routing instances can provide further customization and flexibility, as different tenants or applications may choose their preferred routing protocols and provision it with customized parameters, and the operation of one routing instance can be independent from others. The cost of routing instance isolation is that it requires further complexity and more overhead of control plane resources, in some cases the scalability can become challenging.

## 2.2. Attributes of Network Slice

According to the definition of transport network slice in [I-D.ietf-teas-enhanced-vpn], a transport network slice can be characterized by two major types of attributes: the network slice topology and the resources associated with the network slice. Each network slice tenant can specify his requirement on the connectivity and topology between the endpoints, and the requirement on service performance.

Depending on the deployment of network slices, it is possible that several network slices may have the same topology, and with soft isolation it is possible that several network slices may share the same set of network resource. While each transport network slice is determined by the combination of the topology and the resource.

The control plane SHOULD be able to describe and distribute both the topology attributes and the network resource attributes of each network slice.

## 2.3. Number of Network Slices

In 5G scenarios, the number of network slices in a network is relevant to how network slicing is used in the network and the evolution of 5G for vertical industrial services. Although there is no clear answer so far about how many network slices will be deployed in a network, the potential number of network slices is analyzed by classifying the network slicing deployment scenarios into three typical phases.

In the initial phase, network slicing can be used to isolate different types of business of one operator. For example, in a converged multi-service network, different network slices can be created to carry mobile service, fixed broadband service and enterprise service respectively, each type of service may be managed by a separate department or management team. Some particular service

types, such as multicast service may also be deployed in a dedicated network slice. It is also possible that a network infrastructure operator can provide network slices to other network operators as wholesale service. In this phase, the number of network slices in a network would be relatively small, such as in the order of 10 or so. This could be the typical case in the beginning of network slicing deployment.

In the second phase, network slicing can be used to provide isolated virtual networks for tenants of different vertical industries. At the early stage of the vertical industrial service deployment, a few tenants in some typical industries will begin to use network slicing to support their business, such as smart grid, public safety, games etc. Considering the number of the vertical industries, and the number of top tenants in each industry, the number of network slices may increase to around 100.

In the third phase, with the evolution of 5G, network slicing could be widely used by both vertical industrial tenants and premium business tenants. The total amount of network slices could increase to the order of 1000 or more. While it is expected that the number of network slices would be still less than the number of traditional VPN services in the network.

The control plane needs to be able to support different deployment phases of network slicing, and the number of network slices required in each phase.

### 3. Control Plane Functions

In order to meet the requirements as described in section 2, the control plane of enhanced VPN could be based on a hybrid of centralized controller and distributed control plane.

#### 3.1. Distributed Control Plane

In the overlay of enhanced VPN, the distributed control plane is used to advertise the routing information of different applications tenants. BGP based L3VPN [RFC4364] and EVPN [RFC7432] can provide the functionality needed for the overlay control plane of enhanced VPN. Whether some extensions in overlay control plane are needed will depend on the service requirements. This is out of the scope of this document.

In the underlay of enhanced VPN, the distributed control plane is responsible for advertising and collecting the customized topology and resource information of the virtual networks associated with different enhanced VPNs. A network node may participate in multiple

virtual networks, in this case the node needs to obtain the information of each virtual network it participates in, so that the node can generate the routing and forwarding entries for each virtual network independently.

Currently there are several candidate mechanisms for the underlay control plane. Either Multi-Topology Routing (MTR) [RFC4915] [RFC5120] or Flex-Algo [I-D.ietf-lsr-flex-algo] can be used to specify and distribute the customized topology and some of the TE attributes of the virtual networks, then independent route computation could be performed based on the routing database of each virtual network. However, in order to support both hard and soft isolation in one network, some extensions would be needed to specify and distribute the network resource information of the underlay network and its association with each virtual network. Such extensions are defined in [I-D.dong-lsr-sr-enhanced-vpn] and other accompanying documents.

### 3.2. Centralized Controller

With the introduction of SDN, a centralized controller can be used to collect the network topology and associated attributes from the underlay network, and provide global computation and optimization for the traffic engineered (TE) paths. Several existing control protocols have been designed for the interaction between the controller and the network nodes. While in order to provide the required functionality for different virtual networks, necessary extensions to these protocols would be needed.

- o BGP-LS [RFC7752] provides mechanisms to distribute the topology and TE information of the underlay network to the centralized controller. In the context of enhanced VPN, It be further extended to distribute the topology and resource attributes of the virtual networks to the controller.
- o PCEP [RFC5440] provides mechanisms for Path Computation Elements (PCEs) to perform path computations in response to Path Computation Client (PCC) requests. It can also be used for the creation and deletion of PCE-initiated paths in the network [RFC8281]. In the context of enhanced VPN, It can be further extended for path computation request, responses and path provisioning within a particular virtual network.
- o Netconf/YANG [RFC6241] [RFC7950] provides mechanisms for the configuration of network device and protocols. In the context of enhanced VPN, some extension to existing data models may be needed for the configuration of virtual network specific attributes.

The detailed protocol extensions are out of the scope of this document and will be specified in separate documents.

#### 4. Scalability Considerations

With the development and evolution of 5G network slicing, more and more transport network slices will be deployed. In different transport network slices derived from the same underlay network, the computed paths between the same pair of network nodes can be different, and the resource used for packet forwarding and processing in different network slices can also be different. In order to provide routing in different transport network slices, several aspects need to be considered, such as whether separated routing protocols or routing instances need to be provided for different transport network slices, and how to identify the same network node or link in different transport network slices. The answer to these problems will impact the scalability of both the control plane and the data plane.

In this section, the scalability of control plane is analyzed to understand whether or not the control plane mechanisms could support the required amount of transport network slices.

##### 4.1. Distributed Control Plane

As network slicing requires to provide customized topology and resource attributes to different applications or tenants, it is expected that more state will be introduced into the underlay network. The scalability of the distributed control plane of the underlay network needs to be considered in the following aspects:

- o The number of protocol instances to be maintained on each node
- o The number of the protocol sessions to be maintained on each node
- o The number of routes to be advertised in the network
- o The amount of information and attributes associated with each route to be advertised
- o The number of route computation (i.e. SPF) to be executed on each node

As the number of network slices increases, it is expected that for some of the aspects listed above, the overhead in control plane may be not be affordable. For example, the overhead of maintaining separated logical routing systems for different network slices is higher than maintaining separate routing instances, which is also



higher than maintaining separated network topologies in the same routing instance. In order to meet the requirement of increasing network slices in future, It is suggested to choose the control plane mechanisms which could improve the scalability while still provide the required functionality.

#### 4.2. Centralized Control Plane

Although the SDN approach can reduce the amount of control plane overhead in the distributed control plane, SDN may transfer some of the scalability concerns from the network to the centralized controller, thus the scalability of the controller with network slicing also needs to be considered.

In order to provide global optimization for TE paths in different network slices, the controller needs to keep the information of all the network slices up to date. To achieve this, the controller may need to maintain a communication channel with each network node in the network. When there is significant change in the network and multiple network slices requires global optimization, there may be a heavy processing burden at the controller, and a heavy load in the network surrounding the controller for the distribution of the updated network state.

#### 5. Optimization Mechanisms

For the distributed control plane, several optimization mechanisms are proposed to reduce the overhead and improve the control plane scalability.

The first mechanism is to reduce the amount of control plane sessions. For network slices which have the same peering relationship between two adjacent nodes, it is proposed that one single control session is shared by multiple network slices, information of different network slices can be exchanged over the same control session, with necessary information to distinguish them in the control message. This could reduce the overhead of maintaining large amount of control sessions, and could also reduce the amount of routing information flooding in the network.

The second mechanism is to decouple different types of attributes of a network slice, so that different types of information can be advertised and processed separately in control plane. One example is to decouple the topology and resource attribute of the network slice. This can reduce the amount of route computation introduced by the increased number of network slices. For a group of network slices which have the same network topology, the result of topology based computation could be shared, which means the SPF computation only

needs to be executed once for this group of network slices. This way, the computation overhead could be reduced, especially when there are a large number of network slices, with only a small set of different network topologies. In order to obtain this optimization benefit, network nodes need to be aware of which set of network slices have the same topology, even if the other attributes of the network slices (e.g. resource attributes) are different. Some mechanism to decouple the topology attributes and other attributes of the network slices would be needed. This methodology also applies to other attributes which can be processed independently.

For the centralized control plane, it is considered that the centralized controller is deployed as a complementary mechanism to the distributed control plane rather than a replacement, so that the computation burden in control plane could be shared by both the centralized controller and the network nodes, thus the scalability of both could be improved.

## 6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

## 7. Security Considerations

TBD

## 8. Acknowledgements

The authors would like to thank Zhibo Hu for his review and suggestions to this document.

## 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>>.

### 9.2. Informative References

- [I-D.dong-lsr-sr-enhanced-vpn]  
Dong, J. and S. Bryant, "IGP Extensions for Segment Routing based Enhanced VPN", draft-dong-lsr-sr-enhanced-vpn-01 (work in progress), October 2018.
- [I-D.ietf-lsr-flex-algo]  
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-04 (work in progress), September 2019.
- [I-D.ietf-teas-enhanced-vpn]  
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Networks (VPN+) Service", draft-ietf-teas-enhanced-vpn-03 (work in progress), September 2019.
- [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>>.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P. Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF", RFC 4915, DOI 10.17487/RFC4915, June 2007, <<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)", RFC 5120, DOI 10.17487/RFC5120, February 2008, <<https://www.rfc-editor.org/info/rfc5120>>.
- [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>>.
- [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>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.

- [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>>.
- [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>>.
- [RFC8281] Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "Path Computation Element Communication Protocol (PCEP) Extensions for PCE-Initiated LSP Setup in a Stateful PCE Model", RFC 8281, DOI 10.17487/RFC8281, December 2017, <<https://www.rfc-editor.org/info/rfc8281>>.
- [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>>.
- [TS28530] "3GPP TS28.530", 2016, <<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273>>.

## Authors' Addresses

Jie Dong  
Huawei Technologies

Email: [jie.dong@huawei.com](mailto:jie.dong@huawei.com)

Zhenbin Li  
Huawei Technologies

Email: [lizhenbin@huawei.com](mailto:lizhenbin@huawei.com)

Fengwei Qin  
China Mobile

Email: [qinfengwei@chinamobile.com](mailto:qinfengwei@chinamobile.com)

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 2, 2020

Y. Lee, Ed.  
SKKU  
D. Dhody, Ed.  
S. Karunanithi  
Huawei Technologies  
R. Vilalta  
CTTC  
D. King  
Lancaster University  
D. Ceccarelli  
Ericsson  
October 30, 2019

YANG models for VN/TE Performance Monitoring Telemetry and Scaling  
Intent Autonomics  
draft-ietf-teas-actn-pm-telemetry-autonomics-01

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.

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 May 2, 2020.

## 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
1.1. Terminology . . . . .	4
1.2. Tree diagram . . . . .	4
1.3. Prefixes in Data Node Names . . . . .	4
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 . . . . .	12
7. Yang Data Model . . . . .	15
7.1. ietf-te-kpi-telemetry model . . . . .	15
7.2. ietf-vn-kpi-telemetry model . . . . .	20
8. Security Considerations . . . . .	24
9. IANA Considerations . . . . .	25
10. Acknowledgements . . . . .	25
11. References . . . . .	25
11.1. Normative References . . . . .	25
11.2. Informative References . . . . .	27
Authors' Addresses . . . . .	28

## 1. Introduction

The YANG model discussed in [I-D.ietf-teas-actn-vn-yang] 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 [I-D.ietf-teas-yang-te] 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 [I-D.ietf-teas-actn-vn-yang] or TE-tunnel [I-D.ietf-teas-yang-te]. 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
te	ietf-te	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te-types]
te-tel	ietf-te-kpi-telemetry	[This I-D]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
vn-tel	ietf-vn-kpi-telemetry	[This I-D]

Table 1: Prefixes and corresponding YANG modules

## 2. Use-Cases

[I-D.xu-actn-perf-dynamic-service-control] 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.

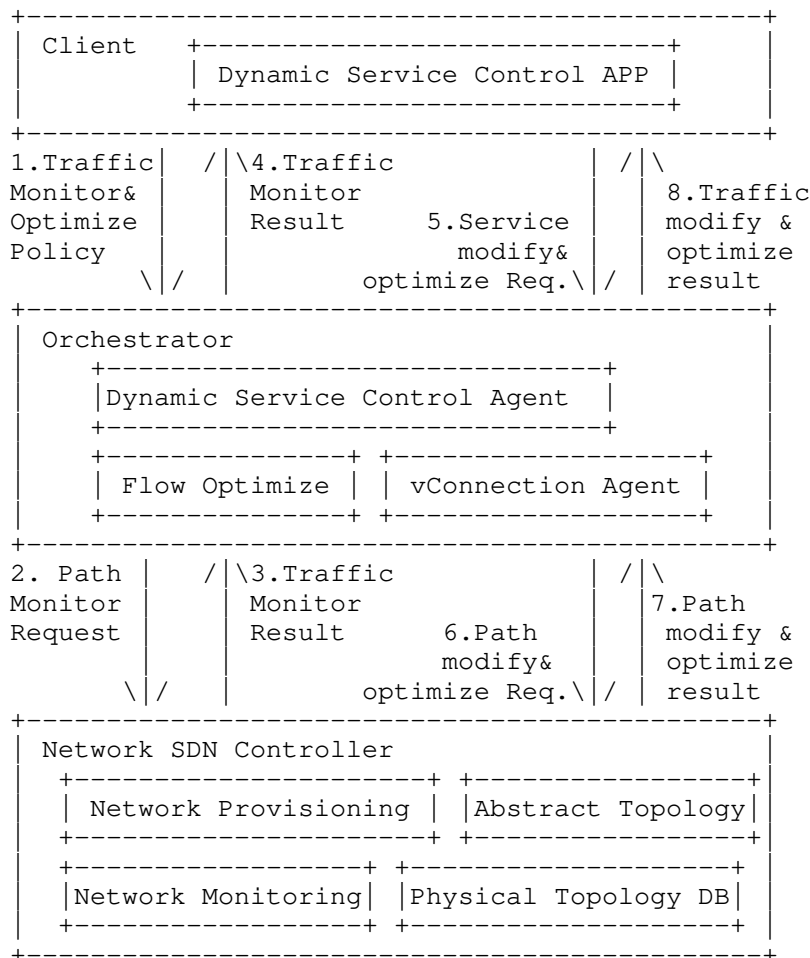


Figure 1: Workflows for dynamic service control based on traffic monitoring

Some of the key points from

[I-D.xu-actn-perf-dynamic-service-control] are as follows:

- o 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).

- o 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.
- o 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

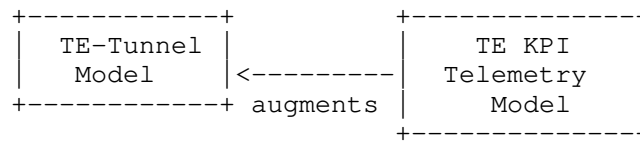
The YANG models developed in this document describe two models:

- (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 & Section 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 & Section 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)

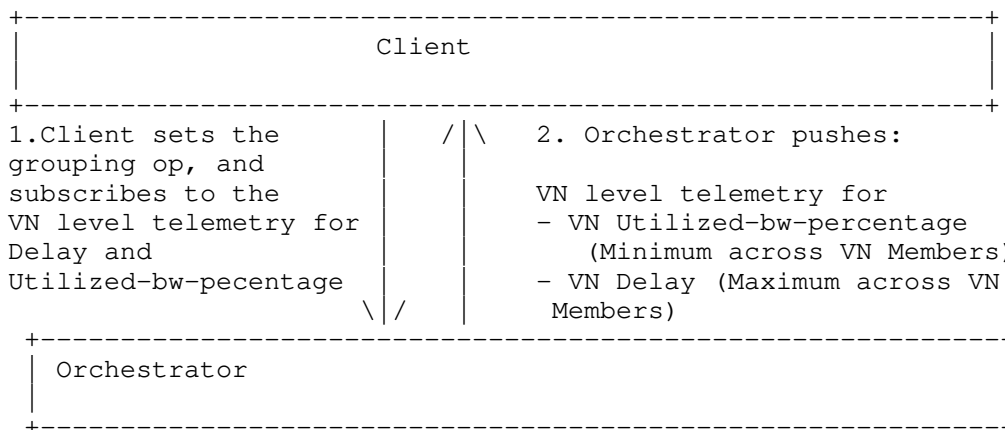
The TE KPI Telemetry Model augments the TE-Tunnel Model to enhance TE performance monitoring capability. This monitoring capability will facilitate proactive re-optimization and reconfiguration of TEs based on the performance monitoring data collected via the TE KPI Telemetry YANG model.



### 3.2. VN KPI Telemetry Model

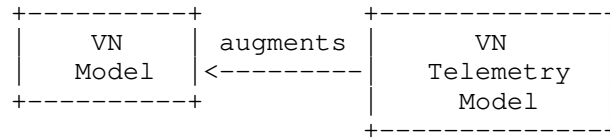
This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomies 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:

- o scale-out-intent or scale-in-intent: whether to scale-out or scale-in.
- o 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.)
- o threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.
- o 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.
- o Threshold-time: the duration for which the criteria must hold true.
- o 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 scaling-condition* [performance-type]
    |   |   |   +--rw performance-type      identityref
    |   |   |   +--rw threshold-value?      string
    |   |   |   +--rw scale-in-operation-type?
    |   |   |       scaling-criteria-operation
    |   +--rw scale-out-intent
    |   |   +--rw threshold-time?      uint32
    |   |   +--rw cooldown-time?      uint32
    |   |   +--rw scaling-condition* [performance-type]
    |   |   |   +--rw performance-type      identityref
    |   |   |   +--rw threshold-value?      string
    |   |   |   +--rw scale-out-operation-type?
    |   |   |       scaling-criteria-operation

```

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:

- o Set Threshold-time: x (sec) (duration for which the criteria must hold true)
- o Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)
- o 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

- o performance type: Two-way-delay
- o threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- o performance type: Utilized bandwidth
- o threshold-value: w megabytes

## 5. Notification

This model does not define specific notifications. To enable notifications, the mechanism defined in [RFC8641] and [RFC8640] can be used. This mechanism currently allows the user to:

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

### 5.1. YANG Push Subscription Examples

[RFC8641] 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 (Tunnell). 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>Tunnell</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 scaling-condition* [performance-type]
          +--rw performance-type      identityref
          +--rw threshold-value?      string
          +--rw scale-in-operation-type?
            scaling-criteria-operation
      +--rw scale-out-intent
        +--rw threshold-time?      uint32
        +--rw cooldown-time?      uint32
        +--rw scaling-condition* [performance-type]
          +--rw performance-type      identityref
          +--rw threshold-value?      string
          +--rw scale-out-operation-type?
```



```

|           scaling-criteria-operation
+---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

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 scaling-condition* [performance-type]
|           +---rw performance-type          identityref
|           +---rw threshold-value?          string
|           +---rw scale-in-operation-type?
|               |           scaling-criteria-operation
+---rw scale-out-intent
|   +---rw threshold-time?                  uint32
|   +---rw cooldown-time?                  uint32
|   +---rw scaling-condition* [performance-type]
|       +---rw performance-type              identityref
|       +---rw threshold-value?              string
|       +---rw scale-out-operation-type?
|           |           scaling-criteria-operation
+---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@2019-10-30.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
      "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
      Engineering Tunnels and Interfaces";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "I-D.ietf-teas-yang-te-types: Traffic Engineering Common
      YANG Types";
  }

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
    Working Group";
  contact
    "WG Web: <https://tools.ietf.org/wg/teas/>
    WG List: <mailto:teas@ietf.org>
    Editor: Young Lee <leeyoung@huawei.com>
    Dhruv Dhody <dhruv.ietf@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-ietf-teas-pm-telemetry-
   autonomics becomes an RFC.*/

revision 2019-10-30 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG models for VN/TE Performance Monitoring
    Telemetry and Scaling Intent Autonomics";
}

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";
    }
}

grouping scaling-in-intent {
    description
        "Basic scaling in intent";
```

```
    uses scaling-duration;
    list scaling-condition {
        key "performance-type";
        description
            "Scaling conditions";
        uses scaling-criteria;
        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";
        }
    }
}

grouping scaling-out-intent {
    description
        "Basic scaling out intent";
    uses scaling-duration;
    list scaling-condition {
        key "performance-type";
        description
            "Scaling conditions";
        uses scaling-criteria;
        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";
        }
    }
}

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 autonomies 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;
}
}
}

```

<CODE ENDS>

## 7.2. ietf-vn-kpi-telemetry model

The YANG code is as follows:

<CODE BEGINS> file "ietf-vn-kpi-telemetry@2019-10-30.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
            "I-D.ietf-teas-actn-vn-yang: A YANG Data Model for VN
            Operation";
    }
    import ietf-te {
        prefix te;
        reference
            "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
            Engineering Tunnels and Interfaces";
    }
}

```



```
import ietf-te-types {
  prefix te-types;
  reference
    "I-D.ietf-teas-yang-te-types: Traffic Engineering Common
    YANG Types";
}
import ietf-te-kpi-telemetry {
  prefix te-kpi;
  reference
    "RFC XXXX: 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
  "WG Web: <https://tools.ietf.org/wg/teas/>
  WG List: <mailto:teas@ietf.org>
  Editor: Young Lee <leeyoung@huawei.com>
  Dhruv Dhody <dhruv.ietf@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 2019-10-30 {
  description
```

```
    "Initial revision.";
  reference
    "RFC XXXX: YANG models for VN/TE Performance Monitoring
      Telemetry and Scaling Intent Autonomics";
}

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:

- o /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent
- o /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent
- o /vn:vn/vn:vn-list/vn-scaling-intent/scale-in-intent
- o /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

- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-06 (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-21 (work in progress), April 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-11 (work in progress), October 2019.
- [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>>.
- [RFC8233] Dhody, D., Wu, Q., Manral, V., Ali, Z., and K. Kumaki, "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, DOI 10.17487/RFC8233, September 2017, <<https://www.rfc-editor.org/info/rfc8233>>.
- [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>>.
- [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>>.

## 11.2. Informative References

- [I-D.xu-actn-perf-dynamic-service-control]  
Xu, Y., Zhang, G., Cheng, W., and z. zhenghaomian@huawei.com, "Use Cases and Requirements of Dynamic Service Control based on Performance Monitoring in ACTN Architecture", draft-xu-actn-perf-dynamic-service-control-03 (work in progress), April 2015.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC8640] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Dynamic Subscription to YANG Events and Datastores over NETCONF", RFC 8640, DOI 10.17487/RFC8640, September 2019, <<https://www.rfc-editor.org/info/rfc8640>>.
- [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>>.

## Authors' Addresses

Young Lee (editor)  
SKKU

Email: [younglee.tx@gmail.com](mailto:younglee.tx@gmail.com)

Dhruv Dhody (editor)  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

Email: [dhruv.ietf@gmail.com](mailto:dhruv.ietf@gmail.com)

Satish Karunanithi  
Huawei Technologies  
Divyashree Techno Park, Whitefield  
Bangalore, Karnataka 560066  
India

Email: [satish.karunanithi@gmail.com](mailto:satish.karunanithi@gmail.com)



Ricard Vilalta  
CTTC  
Centre Tecnologic de Telecomunicacions de Catalunya (CTTC/CERCA)  
Barcelona  
Spain

Email: ricard.vilalta@cttc.es

Daniel King  
Lancaster University

Email: d.king@lancaster.ac.uk

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2020

Y. Lee, Ed.  
SKKU  
D. Dhody, Ed.  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
I. Bryskin  
Futurewei  
B. Yoon  
ETRI  
October 31, 2019

A Yang Data Model for VN Operation  
draft-ietf-teas-actn-vn-yang-07

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 May 3, 2020.

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
1.1. Terminology . . . . .	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 . . . . .	4
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 . . . . .	12
4.3.3. Others . . . . .	13
4.3.4. Summary . . . . .	14
5. VN YANG Model (Tree Structure) . . . . .	14
6. VN YANG Code . . . . .	16
7. JSON Example . . . . .	25
7.1. VN JSON . . . . .	26
7.2. TE-topology JSON . . . . .	32
8. Security Considerations . . . . .	48
9. IANA Considerations . . . . .	49
10. Acknowledgments . . . . .	50
11. References . . . . .	50
11.1. Normative References . . . . .	50
11.2. Informative References . . . . .	51
Appendix A. Contributors Addresses . . . . .	52
Authors' Addresses . . . . .	53

## 1. Introduction

This document provides a YANG 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-l1csm-yang] to provide a complete life-cycle service management and operations.

The YANG model discussed in this document basically provides the following:

- o Characteristics of Access Points (APs) that describe customer's end point characteristics;
- o 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;
- o Characteristics of Virtual Networks (VNs) that describe the customer's VNs in terms of VN Members comprising a VN, multi-source and/or multi-destination characteristics of VN Member, the VN's reference to TE-topology's Abstract Node;

The actual VN instantiation and computation is performed with Connectivity Matrices sub-module of TE-Topology Model [I-D.ietf-teas-yang-te-topo] 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.

### 1.2. 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]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[I-D.ietf-teas-yang-te]
te-topo	ietf-te-topology	[I-D.ietf-teas-yang-te-topo]

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.

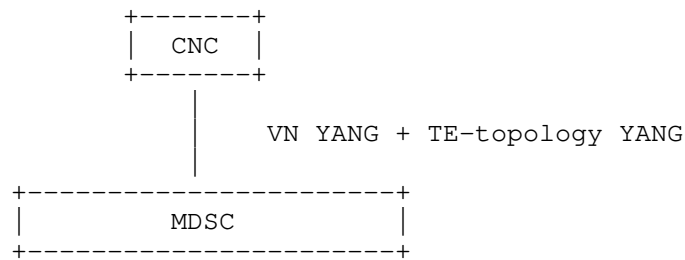


Figure 1: ACTN CMI

Both ACTN VN YANG and TE-topology models are used over the CMI to establish a VN over TE networks.

In the context of 5G transport application, 5G Traffic Provisioning Manager (TPM) that provides slicing requirements to the transport networks (i.e., MDSC) can be considered as a type of CNC. The ACTN CMI provides the necessary interface functions between 5G and transport networks in order to facilitate dynamic VN creation and its lifecycle management with proper feedback loop for monitoring.

### 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:

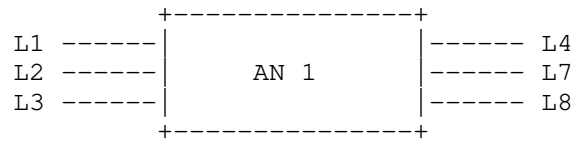


Figure 2: Abstract Node (One node topology)

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.

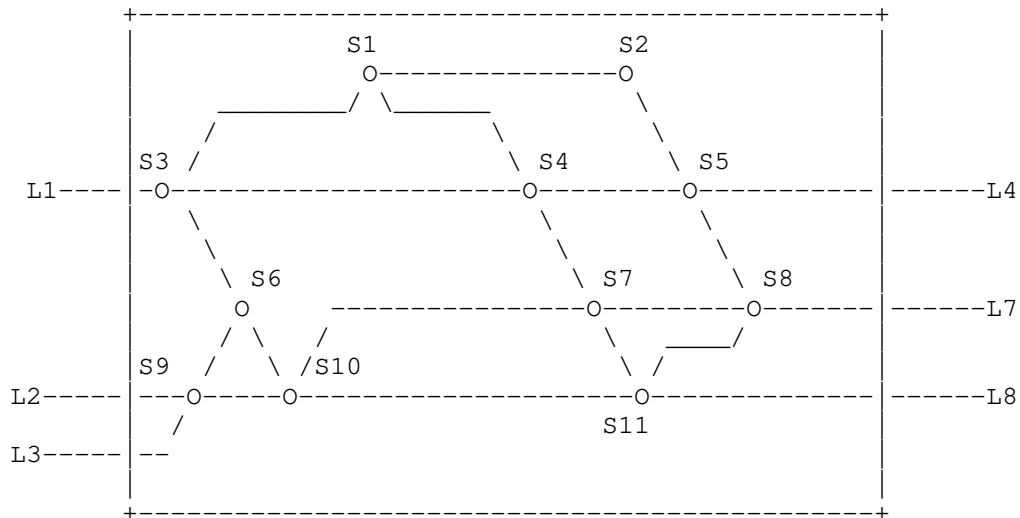


Figure 3: Type 2 topology

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

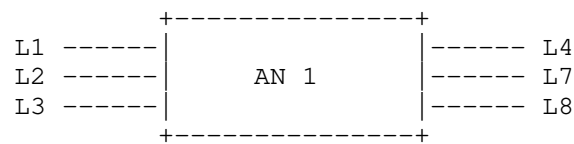
#### 3.1. Type 1 VN Illustration

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

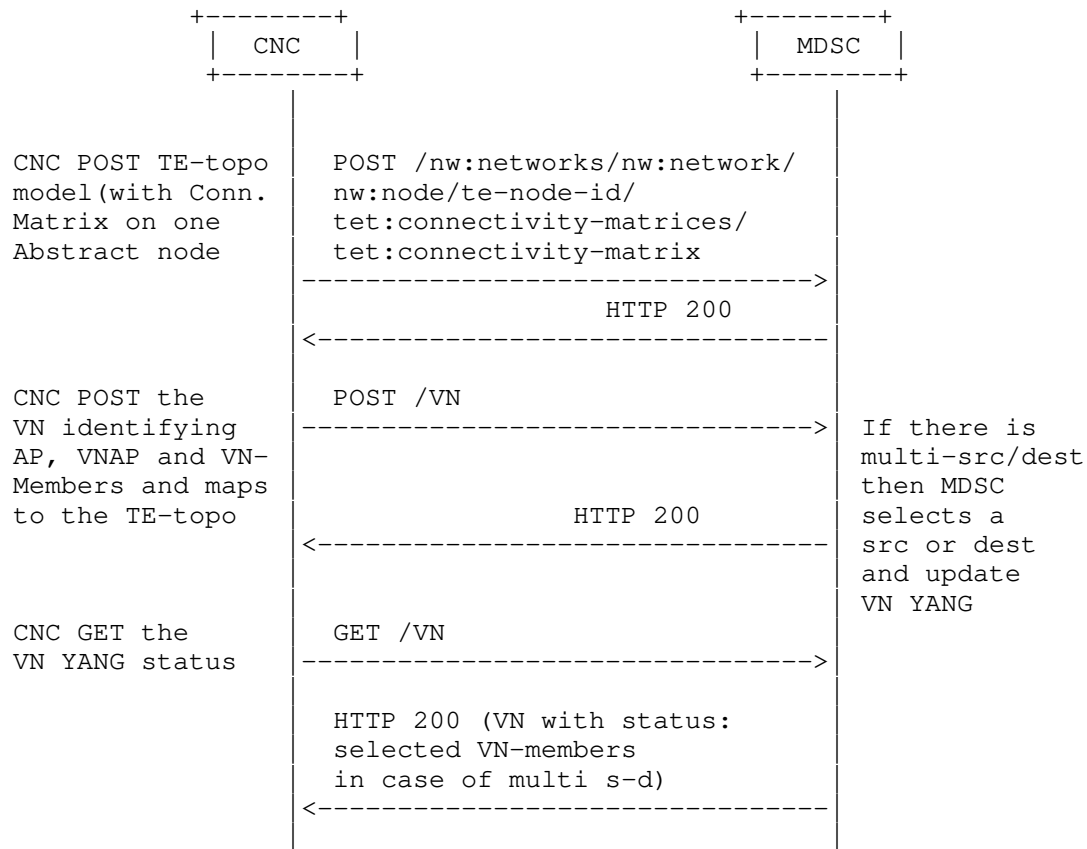
Where L1, L2, L3, L4, L7 and L8 correspond to Access Points.

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.



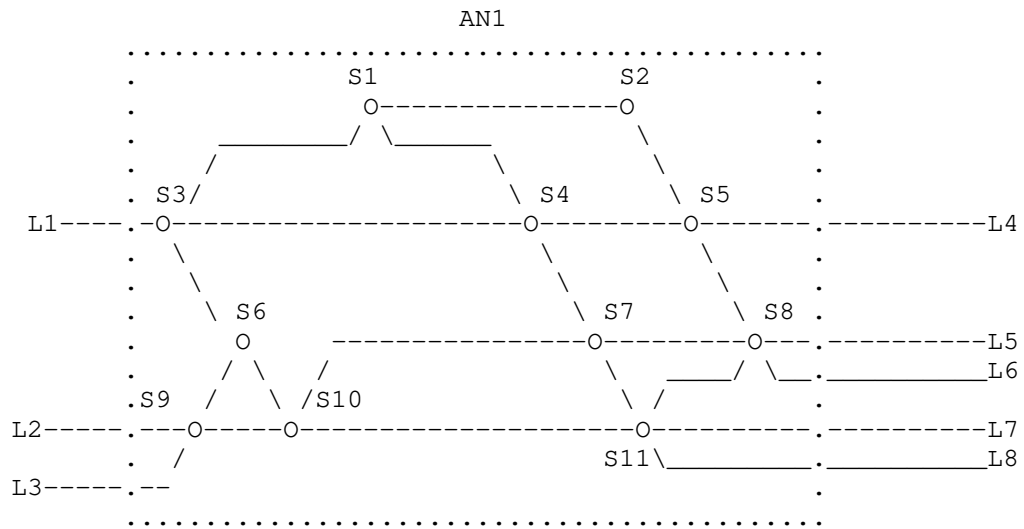


### 3.2. Type 2 VN Illustration

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)

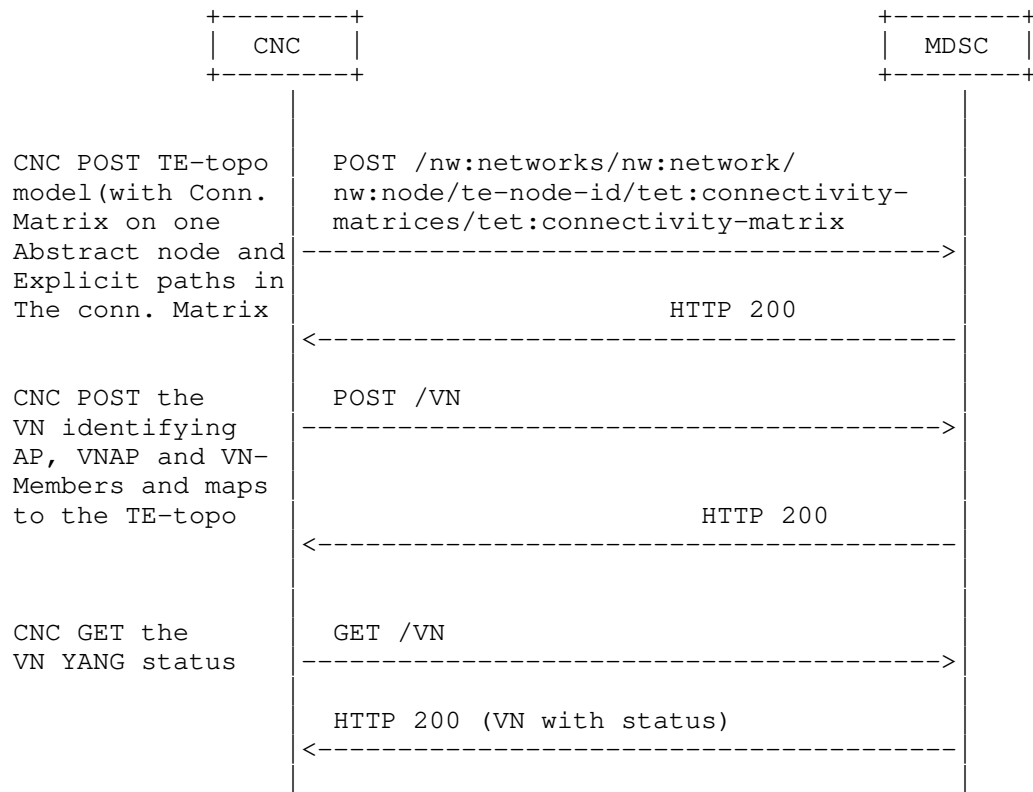
Where the following topology is the underlay for Abstraction Node 1 (AN1).



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

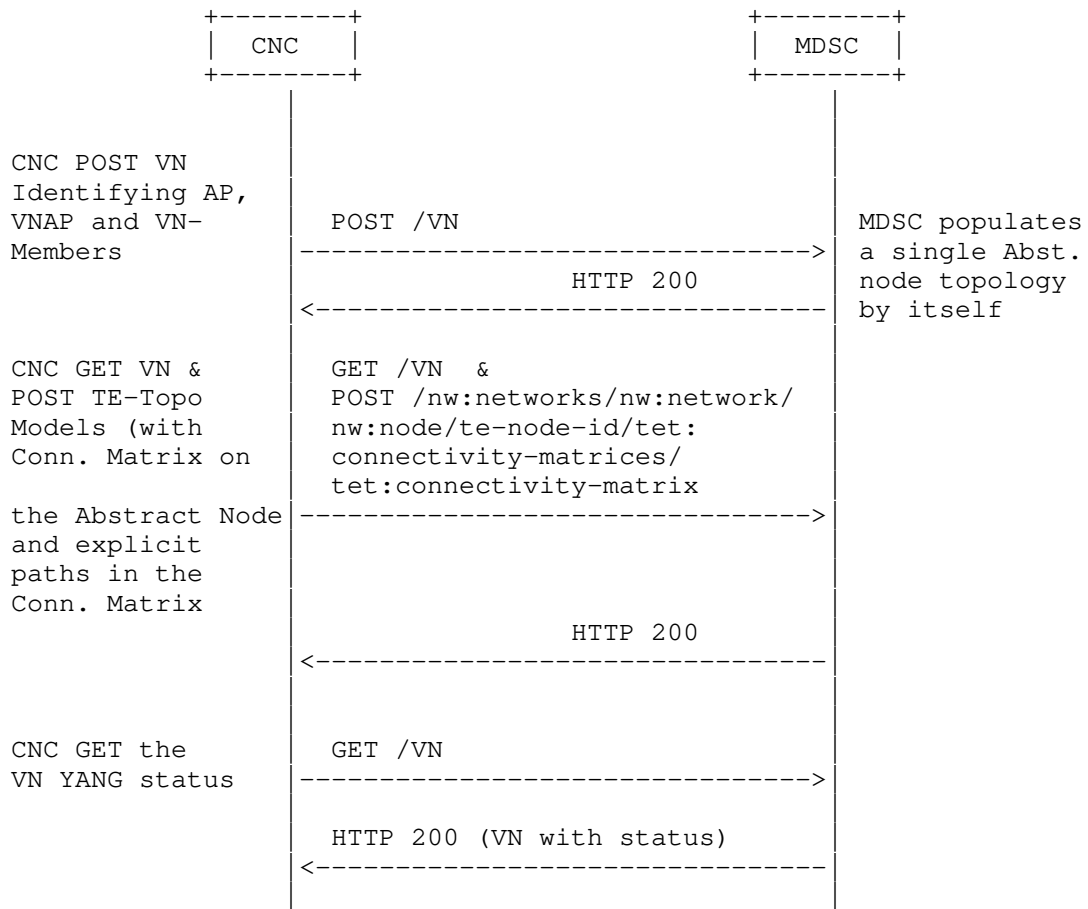
Case 1:

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.

###### 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.

```

+--rw vn
  +--rw vn-list* [vn-id]
    +--rw vn-id                uint32
    +--rw vn-name?             string
    +--rw vn-topology-id?      te-types:te-topology-id
    +--rw abstract-node?
      |
      |   -> /nw:networks/network/node/tet:te-node-id
    +--rw vn-member-list* [vn-member-id]
      |
      |   +--rw vn-member-id    uint32
      |   +--rw src
      |     |
      |     |   +--rw src?
      |     |   |
      |     |   |   -> /ap/access-point-list/access-point-id
      |     |   +--rw src-vn-ap-id?
      |     |   |
      |     |   |   -> /ap/access-point-list/vn-ap/vn-ap-id
      |     |   +--rw multi-src?    boolean {multi-src-dest}?
      |   +--rw dest
      |     |
      |     |   +--rw dest?
      |     |   |
      |     |   |   -> /ap/access-point-list/access-point-id
      |     |   +--rw dest-vn-ap-id?
      |     |   |
      |     |   |   -> /ap/access-point-list/vn-ap/vn-ap-id
      |     |   +--rw multi-dest?    boolean {multi-src-dest}?
      |   +--rw connectivity-matrix-id? leafref
      |   +--ro oper-status?        identityref
      +--ro if-selected?            boolean {multi-src-dest}?
      +--rw admin-status?          identityref
      +--ro oper-status?          identityref
      +--rw vn-level-diversity?    vn-disjointness

```

#### 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 autonomics as described in [I-D.ietf-teas-actn-pm-telemetry-autonomics].

#### 4.3.4. Summary

This section summarizes the innovative service features of the VN Yang.

- o Maintenance of AP and VNAP along with VN
- o VN construct to group of edge-to-edge links
- o VN Compute (pre-instantiate)
- o Multi-Source / Multi-Destination
- o 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 ap
  |   +--rw access-point-list* [access-point-id]
  |   |   +--rw access-point-id      uint32
  |   |   +--rw access-point-name?   string
  |   |   +--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          uint32
  |   |   |   +--rw vn?               -> /vn/vn-list/vn-id
  |   |   |   +--rw abstract-node?
  |   |   |   |   -> /nw:networks/network/node/tet:te-node-id
  |   |   |   +--rw ltp?              leafref
  |   +--rw vn
  |   |   +--rw vn-list* [vn-id]
  |   |   |   +--rw vn-id              uint32
  |   |   |   +--rw vn-name?          string
  |   |   |   +--rw vn-topology-id?   te-types:te-topology-id
  |   |   |   +--rw abstract-node?
  |   |   |   |   -> /nw:networks/network/node/tet:te-node-id
  |   |   +--rw vn-member-list* [vn-member-id]

```

```

+---rw vn-member-id                uint32
+---rw src
|   +---rw src?
|   |       -> /ap/access-point-list/access-point-id
+---rw src-vn-ap-id?
|   |       -> /ap/access-point-list/vn-ap/vn-ap-id
+---rw multi-src?                  boolean {multi-src-dest}?
+---rw dest
|   +---rw dest?
|   |       -> /ap/access-point-list/access-point-id
+---rw dest-vn-ap-id?
|   |       -> /ap/access-point-list/vn-ap/vn-ap-id
+---rw multi-dest?                  boolean {multi-src-dest}?
+---rw connectivity-matrix-id?     leafref
+---ro oper-status?                 identityref
+---ro if-selected?                 boolean {multi-src-dest}?
+---rw admin-status?                 identityref
+---ro oper-status?                 identityref
+---rw vn-level-diversity?          vn-disjointness

rpcs:
+---x vn-compute
+---w input
|   +---w abstract-node?
|   |       -> /nw:networks/network/node/tet:te-node-id
+---w vn-member-list* [vn-member-id]
|   +---w vn-member-id                uint32
|   +---w src
|   |   +---w src?
|   |   |       -> /ap/access-point-list/access-point-id
|   |   +---w src-vn-ap-id?
|   |   |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |   +---w multi-src?                  boolean {multi-src-dest}?
|   +---w dest
|   |   +---w dest?
|   |   |       -> /ap/access-point-list/access-point-id
|   |   +---w dest-vn-ap-id?
|   |   |       -> /ap/access-point-list/vn-ap/vn-ap-id
|   |   +---w multi-dest?                  boolean {multi-src-dest}?
|   +---w connectivity-matrix-id?     leafref
+---w vn-level-diversity?          vn-disjointness
+---ro output
|   +---ro vn-member-list* [vn-member-id]
|   +---ro vn-member-id                uint32
|   +---ro src
|   |   +---ro src?
|   |   |       -> /ap/access-point-list/access-point-id
|   |   +---ro src-vn-ap-id?

```



```

| | -> /ap/access-point-list/vn-ap/vn-ap-id
| +--ro multi-src?          boolean {multi-src-dest}?
+--ro dest
| +--ro dest?
| | -> /ap/access-point-list/access-point-id
| +--ro dest-vn-ap-id?
| | -> /ap/access-point-list/vn-ap/vn-ap-id
| +--ro multi-dest?          boolean {multi-src-dest}?
+--ro connectivity-matrix-id? leafref
+--ro if-selected?           boolean
| {multi-src-dest}?
+--ro compute-status?        identityref

```

## 6. VN YANG Code

The YANG code is as follows:

```

<CODE BEGINS> file "ietf-vn@2019-10-31.yang"
module ietf-vn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vn";
  prefix vn;

  /* Import network */

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

```

```
/* Import TE Topology */

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: <https://tools.ietf.org/wg/teas/>
  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 VN. It describes a
  VN operation module that takes place in the context of the
  CNC-MDSC Interface (CMI) of the ACTN architecture where the
  CNC is the actor of a VN Instantiation/modification/deletion.";

revision 2019-10-31 {
  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.";
}

/* Identity VN State*/

identity vn-state-type {
  description
    "Base identity for VN state";
}

identity vn-state-up {
  base vn-state-type;
  description
```

```
        "VN state up";
    }

    identity vn-state-down {
        base vn-state-type;
        description
            "VN state down";
    }

    /* Identity VN Admin State*/

    identity vn-admin-state-type {
        description
            "Base identity for VN admin states";
    }

    identity vn-admin-state-up {
        base vn-admin-state-type;
        description
            "VN administratively state up";
    }

    identity vn-admin-state-down {
        base vn-admin-state-type;
        description
            "VN administratively state down";
    }

    /* Identity VN Compute State*/

    identity vn-compute-state-type {
        description
            "Base identity for compute states";
    }

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

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

    identity vn-compute-state-computatione-failed {
```

```
    base vn-compute-state-type;
    description
        "State path compute failed";
}

/* typedef */

typedef vn-disjointness {
    type bits {
        bit node {
            position 0;
            description
                "node disjoint";
        }
        bit link {
            position 1;
            description
                "link disjoint";
        }
        bit srlg {
            position 2;
            description
                "srlg disjoint";
        }
    }
    description
        "type of the resource disjointness for VN level applied
        across all VN members in a VN";
}

/* Groupings */

grouping vn-ap {
    description
        "VNAP related information";
    leaf vn-ap-id {
        type uint32;
        description
            "unique identifier for the referred VNAP";
    }
    leaf vn {
        type leafref {
            path "/vn/vn-list/vn-id";
        }
        description
            "reference to the VN";
    }
    leaf abstract-node {
```

```
    type leafref {
      path "/nw:networks/nw:network/nw:node/tet:te-node-id";
    }
    description
      "a reference to the abstract node in TE Topology that
       represent the VN";
  }
  leaf ltp {
    type leafref {
      path "/nw:networks/nw:network/nw:node/"
        + "nt:termination-point/tet:te-tp-id";
    }
    description
      "Reference LTP in the TE-topology";
  }
} //vn-ap

grouping access-point {
  description
    "AP related information";
  leaf access-point-id {
    type uint32;
    description
      "unique identifier for the referred access point";
  }
  leaf access-point-name {
    type string;
    description
      "ap name";
  }
  leaf 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
      "max bandwidth of the AP";
  }
  leaf avl-bandwidth {
    type te-types:te-bandwidth;
    description
      "available bandwidth of the AP";
  }
}
/*add details and any other properties of AP,
```

```
not associated by a VN
CE port, PE port etc.
*/
list vn-ap {
  key "vn-ap-id";
  uses vn-ap;
  description
    "list of VNAP in this AP";
}
} //access-point

grouping vn-member {
  description
    "vn-member is described by this container";
  leaf vn-member-id {
    type uint32;
    description
      "vn-member identifier";
  }
  container src {
    description
      "the source of VN Member";
    leaf src {
      type leafref {
        path "/ap/access-point-list/access-point-id";
      }
      description
        "reference to source AP";
    }
    leaf src-vn-ap-id {
      type leafref {
        path "/ap/access-point-list/vn-ap/vn-ap-id";
      }
      description
        "reference to source VNAP";
    }
    leaf multi-src {
      if-feature "multi-src-dest";
      type boolean;
      description
        "Is source part of multi-source, where
        only one of the source is enabled";
    }
  }
}
container dest {
  description
    "the destination of VN Member";
  leaf dest {
```

```
        type leafref {
            path "/ap/access-point-list/access-point-id";
        }
        description
            "reference to destination AP";
    }
    leaf dest-vn-ap-id {
        type leafref {
            path "/ap/access-point-list/vn-ap/vn-ap-id";
        }
        description
            "reference to dest VNAP";
    }
    leaf multi-dest {
        if-feature "multi-src-dest";
        type boolean;
        description
            "Is destination part of multi-destination, where only one
            of the destination is enabled";
    }
}
leaf 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
        "reference to connectivity-matrix";
}
} //vn-member

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

/* Configuration data nodes */

container ap {
    description
```

```
    "AP configurations";
  list access-point-list {
    key "access-point-id";
    description
      "access-point identifier";
    uses access-point {
      description
        "access-point information";
    }
  }
}
container vn {
  description
    "VN configurations";
  list vn-list {
    key "vn-id";
    description
      "a virtual network is identified by a vn-id";
    leaf vn-id {
      type uint32;
      description
        "a unique vn identifier";
    }
    leaf vn-name {
      type string;
      description
        "vn name";
    }
    leaf vn-topology-id {
      type te-types:te-topology-id;
      description
        "An optional identifier to the TE Topology Model where the
        abstract nodes and links of the Topology can be found for
        Type 2 VNS";
    }
    leaf abstract-node {
      type leafref {
        path "/nw:networks/nw:network/nw:node/tet:te-node-id";
      }
      description
        "a reference to the abstract node in TE Topology";
    }
    list vn-member-list {
      key "vn-member-id";
      description
        "List of VN-members in a VN";
      uses vn-member;
      leaf oper-status {
```



```
        type identityref {
            base vn-state-type;
        }
        config false;
        description
            "VN-member operational state.";
    }
}
leaf if-selected {
    if-feature "multi-src-dest";
    type boolean;
    default "false";
    config false;
    description
        "Is the vn-member is selected among the multi-src/dest
        options";
}
leaf admin-status {
    type identityref {
        base vn-admin-state-type;
    }
    default "vn-admin-state-up";
    description
        "VN administrative state.";
}
leaf oper-status {
    type identityref {
        base vn-state-type;
    }
    config false;
    description
        "VN operational state.";
}
uses vn-policy;
} // vn-list
} // vn

/* RPC */

rpc vn-compute {
    description
        "The VN computation without actual instantiation";
    input {
        leaf abstract-node {
            type leafref {
                path "/nw:networks/nw:network/nw:node/tet:te-node-id";
            }
            description

```

```

        "a reference to the abstract node in TE Topology";
    }
    list vn-member-list {
        key "vn-member-id";
        description
            "List of VN-members in a VN";
        uses vn-member;
    }
    uses vn-policy;
}
output {
    list vn-member-list {
        key "vn-member-id";
        description
            "List of VN-members in a VN";
        uses vn-member;
        leaf if-selected {
            if-feature "multi-src-dest";
            type boolean;
            default "false";
            description
                "Is the vn-member is selected among the multi-src/dest
                options";
        }
        leaf compute-status {
            type identityref {
                base vn-compute-state-type;
            }
            description
                "VN-member compute state.";
        }
    }
}
} //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

- o 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).

- o 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;
- o 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

```
{
  "ap":{
    "access-point-list": [
      {
        "access-point-id": 101,
        "access-point-name": "101",
        "vn-ap": [
          {
            "vn-ap-id": 10101,
            "vn": 1,
            "abstract-node": "D1",
            "ltp": "1-0-1"
          },
          {
            "vn-ap-id": 10102,
            "vn": 2,
            "abstract-node": "D2",
            "ltp": "1-0-1"
          },
          {
            "vn-ap-id": 10103,
            "vn": 3,
            "abstract-node": "D3",
            "ltp": "1-0-1"
          }
        ]
      }
    ]
  },
  {
```

```
"access-point-id": 202,
"access-point-name": "202",
"vn-ap": [
  {
    "vn-ap-id": 20201,
    "vn": 1,
    "abstract-node": "D1",
    "ltp": "2-0-2"
  }
],
},
{
  "access-point-id": 303,
  "access-point-name": "303",
  "vn-ap": [
    {
      "vn-ap-id": 30301,
      "vn": 1,
      "abstract-node": "D1",
      "ltp": "3-0-3"
    },
    {
      "vn-ap-id": 30303,
      "vn": 3,
      "abstract-node": "D3",
      "ltp": "3-0-3"
    }
  ]
},
{
  "access-point-id": 440,
  "access-point-name": "440",
  "vn-ap": [
    {
      "vn-ap-id": 44001,
      "vn": 1,
      "abstract-node": "D1",
      "ltp": "4-4-0"
    }
  ]
},
{
  "access-point-id": 550,
  "access-point-name": "550",
  "vn-ap": [
    {
      "vn-ap-id": 55002,
      "vn": 2,
```

```

        "abstract-node": "D2",
        "ltp": "5-5-0"
    }
]
},
{
    "access-point-id": 770,
    "access-point-name": "770",
    "vn-ap": [
        {
            "vn-ap-id": 77001,
            "vn": 1,
            "abstract-node": "D1",
            "ltp": "7-7-0"
        },
        {
            "vn-ap-id": 77003,
            "vn": 3,
            "abstract-node": "D3",
            "ltp": "7-7-0"
        }
    ]
},
{
    "access-point-id": 880,
    "access-point-name": "880",
    "vn-ap": [
        {
            "vn-ap-id": 88001,
            "vn": 1,
            "abstract-node": "D1",
            "ltp": "8-8-0"
        },
        {
            "vn-ap-id": 88003,
            "vn": 3,
            "abstract-node": "D3",
            "ltp": "8-8-0"
        }
    ]
}
]
},
"vn":{
    "vn-list": [
        {
            "vn-id": 1,
            "vn-name": "vn1",

```

```
"vn-topology-id": "te-topology:abstract1",
"abstract-node": "D1",
"vn-member-list": [
  {
    "vn-member-id": 104,
    "src": {
      "src": 101,
      "src-vn-ap-id": 10101,
    },
    "dest": {
      "dest": 440,
      "dest-vn-ap-id": 44001,
    },
    "connectivity-matrix-id": 104
  },
  {
    "vn-member-id": 107,
    "src": {
      "src": 101,
      "src-vn-ap-id": 10101,
    },
    "dest": {
      "dest": 770,
      "dest-vn-ap-id": 77001,
    },
    "connectivity-matrix-id": 107
  },
  {
    "vn-member-id": 204,
    "src": {
      "src": 202,
      "dest-vn-ap-id": 20401,
    },
    "dest": {
      "dest": 440,
      "dest-vn-ap-id": 44001,
    },
    "connectivity-matrix-id": 204
  },
  {
    "vn-member-id": 308,
    "src": {
      "src": 303,
      "src-vn-ap-id": 30301,
    },
    "dest": {
      "dest": 880,
      "src-vn-ap-id": 88001,
    }
  }
]
```

```
    },
    "connectivity-matrix-id": 308
  },
  {
    "vn-member-id": 108,
    "src": {
      "src": 101,
      "src-vn-ap-id": 10101,
    },
    "dest": {
      "dest": 880,
      "dest-vn-ap-id": 88001,
    },
    "connectivity-matrix-id": 108
  }
]
},
{
  "vn-id": 2,
  "vn-name": "vn2",
  "vn-topology-id": "te-topology:abstract2",
  "abstract-node": "D2",
  "vn-member-list": [
    {
      "vn-member-id": 105,
      "src": {
        "src": 101,
        "src-vn-ap-id": 10102,
      },
      "dest": {
        "dest": 550,
        "dest-vn-ap-id": 55002,
      },
      "connectivity-matrix-id": 105
    }
  ]
},
{
  "vn-id": 3,
  "vn-name": "vn3",
  "vn-topology-id": "te-topology:abstract3",
  "abstract-node": "D3",
  "vn-member-list": [
    {
      "vn-member-id": 104,
      "src": {
        "src": 101,
      },
    },
  ]
}
```

```

        "dest": {
            "dest": 440,
            "multi-dest": true
        }
    },
    {
        "vn-member-id": 107,
        "src": {
            "src": 101,
            "src-vn-ap-id": 10103,
        },
        "dest": {
            "dest": 770,
            "dest-vn-ap-id": 77003,
            "multi-dest": true
        },
        "connectivity-matrix-id": 107,
        "if-selected": true,
    },
    {
        "vn-member-id": 204,
        "src": {
            "src": 202,
            "multi-src": true,
        },
        "dest": {
            "dest": 440,
        },
    },
    {
        "vn-member-id": 304,
        "src": {
            "src": 303,
            "src-vn-ap-id": 30303,
            "multi-src": true,
        },
        "dest": {
            "dest": 440,
            "src-vn-ap-id": 44003,
        },
        "connectivity-matrix-id": 304,
        "if-selected": true,
    },
]
}

```



```

    }
  }

```

## 7.2. TE-topology JSON

```

{
  "networks": {
    "network": [
      {
        "network-types": {
          "te-topology": {}
        },
        "network-id": "abstract1",
        "provider-id": 201,
        "client-id": 600,
        "te-topology-id": "te-topology:abstract1",
        "node": [
          {
            "node-id": "D1",
            "te-node-id": "2.0.1.1",
            "te": {
              "te-node-attributes": {
                "domain-id" : 1,
                "is-abstract": [null],
                "connectivity-matrices": {
                  "is-allowed": true,
                  "path-constraints": {
                    "bandwidth-generic": {
                      "te-bandwidth": {
                        "generic": [
                          {
                            "generic": "0x1p10",
                          }
                        ]
                      }
                    }
                  }
                }
              },
              "disjointness": "node link srlg",
            },
            "connectivity-matrix": [
              {
                "id": 104,
                "from": "1-0-1",
                "to": "4-4-0"
              },
              {

```

```
        "id": 107,
        "from": "1-0-1",
        "to": "7-7-0"
      },
      {
        "id": 204,
        "from": "2-0-2",
        "to": "4-4-0"
      },
      {
        "id": 308,
        "from": "3-0-3",
        "to": "8-8-0"
      },
      {
        "id": 108,
        "from": "1-0-1",
        "to": "8-8-0"
      },
    ],
  },
  "termination-point": [
    {
      "tp-id": "1-0-1",
      "te-tp-id": 10001,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    },
    {
      "tp-id": "1-1-0",
      "te-tp-id": 10100,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    }
  ]
}
```

```
},
{
  "tp-id": "2-0-2",
  "te-tp-id": 20002,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "2-2-0",
  "te-tp-id": 20200,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-0-3",
  "te-tp-id": 30003,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-3-0",
  "te-tp-id": 30300,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
}
```

```
    }
  },
  {
    "tp-id": "4-0-4",
    "te-tp-id": 40004,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "4-4-0",
    "te-tp-id": 40400,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "5-0-5",
    "te-tp-id": 50005,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "5-5-0",
    "te-tp-id": 50500,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
```

```
    }
  },
  {
    "tp-id": "6-0-6",
    "te-tp-id": 60006,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "6-6-0",
    "te-tp-id": 60600,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "7-0-7",
    "te-tp-id": 70007,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "7-7-0",
    "te-tp-id": 70700,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
```

```

    }
  },
  {
    "tp-id": "8-0-8",
    "te-tp-id": 80008,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "8-8-0",
    "te-tp-id": 80800,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
]
}
]
},
{
  "network-types": {
    "te-topology": {}
  },
  "network-id": "abstract2",
  "provider-id": 201,
  "client-id": 600,
  "te-topology-id": "te-topology:abstract2",
  "node": [
    {
      "node-id": "D2",
      "te-node-id": "2.0.1.2",
      "te": {
        "te-node-attributes": {
          "domain-id" : 1,
          "is-abstract": [null],
          "connectivity-matrices": {
            "is-allowed": true,

```

```
"underlay": {
  "enabled": true
},
"path-constraints": {
  "bandwidth-generic": {
    "te-bandwidth": {
      "generic": [
        {
          "generic": "0x1p10"
        }
      ]
    }
  }
},
"optimizations": {
  "objective-function": {
    "objective-function-type":
      "of-maximize-residual-bandwidth"
  }
},
"connectivity-matrix": [
  {
    "id": 105,
    "from": "1-0-1",
    "to": "5-5-0",
    "underlay": {
      "enabled": true,
      "primary-path": {
        "network-ref": "absolute",
        "path-element": [
          {
            "path-element-id": 1,
            "index": 1,
            "numbered-hop": {
              "address": "4.4.4.4",
              "hop-type": "STRICT"
            }
          },
          {
            "path-element-id": 2,
            "index": 2,
            "numbered-hop": {
              "address": "7.7.7.7",
              "hop-type": "STRICT"
            }
          }
        ]
      }
    }
  ]
}
```

```

    }
  }
]
}
},
"termination-point": [
  {
    "tp-id": "1-0-1",
    "te-tp-id": 10001,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "1-1-0",
    "te-tp-id": 10100,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "2-0-2",
    "te-tp-id": 20002,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "2-2-0",
    "te-tp-id": 20200,

```



```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "3-0-3",
  "te-tp-id": 30003,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-3-0",
  "te-tp-id": 30300,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "4-0-4",
  "te-tp-id": 40004,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "4-4-0",
  "te-tp-id": 40400,
```

```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "5-0-5",
  "te-tp-id": 50005,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-5-0",
  "te-tp-id": 50500,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "6-0-6",
  "te-tp-id": 60006,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "6-6-0",
  "te-tp-id": 60600,
```

```
"te": {
  "interface-switching-capability": [
    {
      "switching-capability": "switching-otn",
      "encoding": "lsp-encoding-oduk"
    }
  ]
},
{
  "tp-id": "7-0-7",
  "te-tp-id": 70007,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "7-7-0",
  "te-tp-id": 70700,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "8-0-8",
  "te-tp-id": 80008,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "8-8-0",
```

```

        "te-tp-id": 80800,
        "te": {
            "interface-switching-capability": [
                {
                    "switching-capability": "switching-otn",
                    "encoding": "lsp-encoding-oduk"
                }
            ]
        }
    ]
},
{
    "network-types": {
        "te-topology": {}
    },
    "network-id": "abstract3",
    "provider-id": 201,
    "client-id": 600,
    "te-topology-id": "te-topology:abstract3",
    "node": [
        {
            "node-id": "D3",
            "te-node-id": "3.0.1.1",
            "te": {
                "te-node-attributes": {
                    "domain-id": 3,
                    "is-abstract": [null],
                    "connectivity-matrices": {
                        "is-allowed": true,
                        "path-constraints": {
                            "bandwidth-generic": {
                                "te-bandwidth": {
                                    "generic": [
                                        {
                                            "generic": "0x1p10",
                                        }
                                    ]
                                }
                            }
                        }
                    },
                    "connectivity-matrix": [
                        {
                            "id": 107,
                            "from": "1-0-1",

```

```
        "to": "7-7-0"
      },
      {
        "id": 308,
        "from": "3-0-3",
        "to": "8-8-0"
      },
    ]
  }
},
"termination-point": [
  {
    "tp-id": "1-0-1",
    "te-tp-id": 10001,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "1-1-0",
    "te-tp-id": 10100,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  },
  {
    "tp-id": "2-0-2",
    "te-tp-id": 20002,
    "te": {
      "interface-switching-capability": [
        {
          "switching-capability": "switching-otn",
          "encoding": "lsp-encoding-oduk"
        }
      ]
    }
  }
],
```

```
{
  "tp-id": "2-2-0",
  "te-tp-id": 20200,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-0-3",
  "te-tp-id": 30003,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "3-3-0",
  "te-tp-id": 30300,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "4-0-4",
  "te-tp-id": 40004,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
}
```

```
},
{
  "tp-id": "4-4-0",
  "te-tp-id": 40400,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-0-5",
  "te-tp-id": 50005,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "5-5-0",
  "te-tp-id": 50500,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "6-0-6",
  "te-tp-id": 60006,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
}
```

```
},
{
  "tp-id": "6-6-0",
  "te-tp-id": 60600,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "7-0-7",
  "te-tp-id": 70007,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "7-7-0",
  "te-tp-id": 70700,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
},
{
  "tp-id": "8-0-8",
  "te-tp-id": 80008,
  "te": {
    "interface-switching-capability": [
      {
        "switching-capability": "switching-otn",
        "encoding": "lsp-encoding-oduk"
      }
    ]
  }
}
```



```

    },
    {
      "tp-id": "8-8-0",
      "te-tp-id": 80800,
      "te": {
        "interface-switching-capability": [
          {
            "switching-capability": "switching-otn",
            "encoding": "lsp-encoding-oduk"
          }
        ]
      }
    }
  ]
},
]
},
]
}
}

```

## 8. Security Considerations

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:

- o access-point-list:
  - \* access-point-id
  - \* max-bandwidth
  - \* avl-bandwidth
- o vn-ap:
  - \* vn-ap-id
  - \* vn
  - \* abstract-node
  - \* ltp
- o vn-list
  - \* vn-id
  - \* vn-topology-id
  - \* abstract-node
- o vn-member-id
  - \* src
  - \* src-vn-ap-id
  - \* dest
  - \* dest-vn-ap-id
  - \* connectivity-matrix-id

## 9. IANA Considerations

This document registers the following namespace URIs in 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.  
-----

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

-----  
name: ietf-vn  
namespace: urn:ietf:params:xml:ns:yang:ietf-vn  
prefix: vn  
reference: RFC XXXX (TDB)  
-----

## 10. Acknowledgments

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

## 11. References

### 11.1. Normative References

- [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-21 (work in  
progress), April 2019.
- [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.
- [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>>.
- [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>>.
- [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>>.

## 11.2. Informative References

- [I-D.ietf-ccamp-llcsm-yang]  
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-llcsm-yang-10 (work in progress), September 2019.
- [I-D.ietf-teas-actn-pm-telemetry-autonomics]  
Lee, Y., Dhody, D., Karunanithi, S., Vilata, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", draft-ietf-teas-actn-pm-telemetry-autonomics-01 (work in progress), October 2019.
- [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", draft-ietf-teas-te-service-mapping-yang-02 (work in progress), September 2019.

- [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>>.
- [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>>.
- [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. 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

#### Authors' Addresses

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

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

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

Igor Bryskin  
Futurewei

Email: i\_bryskin@yahoo.com

Bin Yeong Yoon  
ETRI

Email: byyun@etri.re.kr

TEAS Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 18, 2019

J. Dong  
S. Bryant  
Huawei  
Z. Li  
China Mobile  
T. Miyasaka  
KDDI Corporation  
Y. Lee  
Huawei  
February 14, 2019

A Framework for Enhanced Virtual Private Networks (VPN+) Service  
draft-ietf-teas-enhanced-vpn-01

Abstract

This document specifies a framework for using existing, modified and potential new networking technologies as components to provide an Enhanced Virtual Private Networks (VPN+) service. The purpose is to support the needs of new applications, particularly applications that are associated with 5G services by utilizing an approach that is based on existing VPN technologies and adds features that specific services require over and above traditional VPNs.

Typically, VPN+ will be used to form the underpinning of network slicing, but could also be of use in its own right. It is not envisaged that large numbers of VPN+ instances will be deployed in a network and, in particular, it is not intended that all VPNs supported by a network will use VPN+ techniques.

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 18, 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 . . . . .	3
2. Overview of the Requirements . . . . .	5
2.1. Isolation between Virtual Networks . . . . .	5
2.1.1. A Pragmatic Approach to Isolation . . . . .	7
2.2. Performance Guarantee . . . . .	8
2.3. Integration . . . . .	9
2.3.1. Abstraction . . . . .	10
2.4. Dynamic Configuration . . . . .	10
2.5. Customized Control . . . . .	11
2.6. Applicability . . . . .	11
3. Architecture of Enhanced VPN . . . . .	11
3.1. Layered Architecture . . . . .	13
3.2. Multi-Point to Multi-Point . . . . .	14
3.3. Application Specific Network Types . . . . .	14
3.4. Scaling Considerations . . . . .	14
4. Candidate Technologies . . . . .	15
4.1. Underlay Packet and Frame-Based Data Planes . . . . .	15
4.1.1. FlexE . . . . .	16
4.1.2. Dedicated Queues . . . . .	16
4.1.3. Time Sensitive Networking . . . . .	17
4.2. Packet and Frame-Based Network Layer . . . . .	17
4.2.1. Deterministic Networking . . . . .	17
4.2.2. MPLS Traffic Engineering (MPLS-TE) . . . . .	18
4.2.3. Segment Routing . . . . .	18
4.3. Non-Packet Technologies . . . . .	20
4.4. Control Plane . . . . .	20
4.5. Management Plane . . . . .	21
4.6. Applicability of ACTN to Enhanced VPN . . . . .	21
4.6.1. ACTN Used for VPN+ Delivery . . . . .	23
4.6.2. Enhanced VPN Features with ACTN . . . . .	25
5. Scalability Considerations . . . . .	27

5.1. Maximum Stack Depth of SR . . . . .	28
5.2. RSVP Scalability . . . . .	28
6. OAM Considerations . . . . .	29
7. Enhanced Resiliency . . . . .	29
8. Security Considerations . . . . .	30
9. IANA Considerations . . . . .	31
10. Contributors . . . . .	31
11. Acknowledgements . . . . .	31
12. References . . . . .	32
12.1. Normative References . . . . .	32
12.2. Informative References . . . . .	32
Authors' Addresses . . . . .	36

## 1. Introduction

Virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to 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 enhanced overlay services with advanced characteristics such as complete isolation from other services so that changes in network load or event of other services have no effect on the throughput or latency of the service provided to the customer.

Driven largely by needs surfacing from 5G, the concept of network slicing has gained traction [NGMN-NS-Concept] [TS23501] [TS28530] [BBF-SD406]. Network slicing requires the underlying network to support partitioning the network resources to provide the client with dedicated (private) networking, computing, and storage resources drawn from a shared pool. The slices may be seen as (and operated as) virtual networks.

Network abstraction is a technique that can be applied to a network domain to select network resources by policy to obtain a view of potential connectivity and a set of service functions.

Network slicing is an approach to network operations that builds on the concept of network abstraction to provide programmability, flexibility, and modularity. It may use techniques such as Software Defined Networking (SDN) [RFC7149] and Network Function Virtualization (NFV) to create multiple logical (virtual) networks, each tailored for a set of services or a particular tenant that share the same set of requirements, on top of a common network. How the network slices are engineered can be deployment-specific.

Thus, there is a need to create virtual networks with enhanced characteristics. The tenant of such a virtual network can require a degree of isolation and performance that previously could only be satisfied by dedicated networks. Additionally, the tenant may ask for some level of control to their virtual networks, e.g., to customize the service forwarding paths in a network slice.

These enhanced properties cannot be met with pure overlay networks, as they require tighter coordination and integration between the underlay and the overlay network. This document introduces a new network service called Enhanced VPN: VPN+. VPN+ refers to a virtual network which has dedicated network resources, including invoked service functions, allocated from the underlay network. Unlike a traditional VPN, an enhanced VPN can achieve greater isolation with strict guaranteed performance. These new properties, which have general applicability, may also be of interest as part of a network slicing solution, but it is not envisaged that VPN+ techniques will be applied to normal VPN services that can continue to be deployed using pre-existing mechanisms. Furthermore, it is not intended that large numbers of VPN+ instances will be deployed within a single network. See Section 5 for a discussion of scalability considerations.

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

- o The design of the enhanced data plane.
- o The necessary protocols in both underlay and the overlay of enhanced VPN.
- o The mechanisms to achieve integration between overlay and underlay.
- o The necessary Operation, Administration and Management (OAM) methods to instrument an enhanced VPN to make sure that the required Service Level Agreement (SLA) are met, and to take any corrective action to avoid SLA violation, such as switching to an alternate path.

The required network layered structure to achieve this is shown in Section 3.1.

Note that, in this document, the four terms "VPN", "Enhanced VPN" (or "VPN+"), "Virtual Network (VN)", and "Network Slice" may be considered as describing similar concepts dependent on the viewpoint from which they are used.

- o An enhanced VPN is clearly a form of VPN, but with additional service-specific commitments. Thus, care must be taken with the term "VPN" to distinguish normal or legacy VPNs from VPN+ instances.
- o A VN is a type of service that connects customer edge points with the additional possibility of requesting further service characteristics in the manner of an enhanced VPN.
- o An enhanced VPN or VN is made by creating a slice through the resources of the underlay network.
- o The general concept of network slicing in a TE network is a larger problem space than is addressed by VPN+ or VN, but those concepts are tools to address some aspects or realizations of network slicing.

## 2. Overview of the Requirements

In this section we provide an overview of the requirements of an enhanced VPN.

### 2.1. Isolation between Virtual Networks

One element of the SLA demanded for an enhanced VPN is the degree of isolation from other services in the network. Isolation is a feature requested by some particular customers in the network. Such feature is offered by a network operator where the traffic from one service instance is isolated from the traffic of other services. There are different grades of isolation that range from simple separation of traffic on delivery (ensuring that traffic is not delivered to the wrong customer) all the way to complete separation within the underlay so that the traffic from different services use distinct network resources.

The terms hard and soft isolation are introduced to give example of different isolation cases. A VPN has soft isolation if the traffic of one VPN cannot be received by the customers of another VPN. Both IP and MPLS VPNs are examples of soft isolated VPNs because the network delivers the traffic only to the required VPN endpoints. However, with soft isolation, traffic from one or more VPNs and regular network traffic may congest the network resulting in packet loss and delay for other VPNs operating normally. The ability for a VPN to be sheltered from this effect is called hard isolation, and this property is required by some critical applications.

The requirement is for an operator to provide both hard and soft isolation between the tenants/applications using one enhanced VPN and

the tenants/applications using another enhanced VPN. Hard isolation is needed so that applications with exacting requirements can function correctly, despite other demands (perhaps a burst on another VPN) competing for the underlying resources. In practice isolation may be offered as a spectrum between soft and hard, and in some cases soft and hard isolation may be used in a hierarchical manner.

An example of hard isolation is a network supporting both emergency services and public broadband multi-media services. During a major incident the VPNs supporting these services would both be expected to experience high data volumes, and it is important that both make progress in the transmission of their data. In these circumstances the VPNs would require an appropriate degree of isolation to be able to continue to operate acceptably.

In order to provide the required isolation, resources may have to be reserved in the data plane of the underlay network and dedicated to traffic from a specific VPN. This may introduce scalability concerns, thus some trade-off needs to be considered to provide the required isolation between network slices while still allowing reasonable sharing inside each network slice.

An optical layer can offer a high degree of isolation, at the cost of allocating resources on a long term and end-to-end basis. Such an arrangement means that the full cost of the resources must be borne by the service that is allocated with the resources. On the other hand, where adequate isolation can be achieved at the packet layer, this permits the resources to be shared amongst many services and only dedicated to a service on a temporary basis. This in turn, allows greater statistical multiplexing of network resources and thus amortizes the cost over many services, leading to better economy. However, the degree of isolation required by network slicing cannot be entirely met with existing mechanisms such as Traffic Engineered Label Switched Paths (TE-LSPs). This is because most implementations enforce the bandwidth in the data-plane only at the PEs, but at the P routers the bandwidth is only reserved in the control plane, thus bursts of data can accidentally occur at a P router with higher than committed data rate.

There are several new technologies that provide some assistance with these data plane issues. Firstly there is the IEEE project on Time Sensitive Networking [TSN] which introduces the concept of packet scheduling of delay and loss sensitive packets. Then there is [FLEXE] which provides the ability to multiplex multiple channels over one or more Ethernet links in a way that provides hard isolation. Finally there are advanced queueing approaches which allow the construction of virtual sub-interfaces, each of which is

provided with dedicated resource in a shared physical interface. These approaches are described in more detail later in this document.

In the remainder of this section we explore how isolation may be achieved in packet networks.

#### 2.1.1. A Pragmatic Approach to Isolation

A key question is whether it is possible to achieve hard isolation in packet networks, which were never designed to support hard isolation. On the contrary, they were designed to provide statistical multiplexing, a significant economic advantage when compared to a dedicated, or a Time Division Multiplexing (TDM) network. However there is no need to provide any harder isolation than is required by the application. Pseudowires [RFC3985] emulate services that would have had hard isolation in their native form. An approximation to this requirement is sufficient in most cases.

Thus, for example, using FlexE or a channelized sub-interface together with packet scheduling as interfaceslicing, optionally along with the slicing of node resources, a type of hard isolation can be provided that is adequate for many VPN+ applications. Other applications may be either satisfied with a classical VPN with or without reserved bandwidth, or may need dedicated point to point fiber. The needs of each application must be quantified in order to provide an economic solution that satisfies those needs without over-engineering.

This spectrum of isolation is shown in Figure 1:

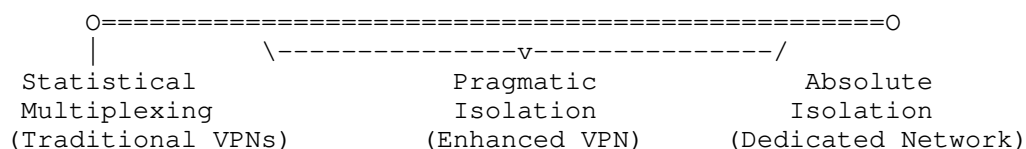


Figure 1: The Spectrum of Isolation

At one end of the above figure, we have traditional statistical multiplexing technologies that support 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 traditional transport networks. The goal of enhanced VPN is pragmatic isolation. This is isolation that is better than is obtainable from pure statistical multiplexing, more cost effective and flexible than a dedicated network, but which is a practical solution that is good enough for the majority of applications.

## 2.2. Performance Guarantee

There are several kinds of performance guarantees, including guaranteed maximum packet loss, guaranteed maximum delay and guaranteed delay variation. Note that these guarantees apply to the conformance traffic, the out-of-profile traffic will be handled following other requirements.

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

Guaranteed maximum latency is required in a number of applications particularly real-time control applications and some types of virtual reality applications. The work of the IETF Deterministic Networking (DetNet) Working Group [DETNET] is relevant; however the scope needs to be extended to methods of enhancing the underlay to better support the delay guarantee, and to integrate these enhancements with the overall service provision.

Guaranteed maximum delay variation is a service that may also be needed. [I-D.ietf-detnet-use-cases] calls up a number of cases where this is needed, for example electrical utilities have an operational need for this. Time transfer is one example of a service that needs this, 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 virtual networks. Alternatively a dedicated virtual network may be used to provide this as a shared service.

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

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

Best effort service is the basic service that current VPNs can provide.

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 best effort with over-capacity provisioning, or it can be based on TE-LSPs with bandwidth reservation. The instantaneous bandwidth is however, not necessarily assured, depending on the technique used. Providing assured bandwidth to VPNs, for example by using TE-LSPs, is not widely deployed at least partially due to scalability concerns. Guaranteed latency and enhanced delivery are not yet integrated with VPNs.

A guaranteed latency service has a latency upper bound provided by the network. Assuring the upper bound is more important than achieving the minimum latency.

In Section 2.1 we considered the work of the IEEE Time Sensitive Networking (TSN) project [TSN] and the work of the IETF DetNet Working group [DETNET] in the context of isolation. The TSN and DetNet work is of greater relevance in assuring end-to-end packet latency. It is also of importance in considering enhanced delivery.

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.

It is these last two characteristics that an enhanced VPN adds to a VPN service.

Flex Ethernet [FLEXE] is a useful underlay to provide these guarantees. This is a method of providing time-slot based channelization over an Ethernet bearer. Such channels are fully isolated from other channels running over the same Ethernet bearer. As noted elsewhere this produces hard isolation but makes the reclamation of unused bandwidth more difficult.

These approaches can be used in tandem. It is possible to use FlexE to provide tenant isolation, and then to use the TSN/Detnet approach to provide a performance guarantee inside the a slice or tenant VPN.

### 2.3. Integration

A solution to the enhanced VPN problem has to provide close integration of both overlay VPN and the underlay network resource. This needs be done in a flexible and scalable way so that it can be widely deployed in operator networks to support a reasonable number of enhanced VPN customers.



Taking mobile networks and in particular 5G into consideration, the integration of network and the service functions is a likely requirement. The work in IETF SFC working group [SFC] provides a foundation for this integration.

#### 2.3.1. Abstraction

Integration of the overlay VPN and the underlay network resources does not need to be a tight mapping. As described in [RFC7926], abstraction is the process of applying policy to a set of information about a 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 network as described in the framework for Abstraction and Control of TE Networks (ACTN) described in [RFC8453] as discussed further in Section 4.5.

#### 2.4. Dynamic Configuration

Enhanced VPNs need to be created, modified, and removed from the network according to service demand. An enhanced VPN that requires hard isolation 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 in particular the traffic in flight will be disrupted can be a difficult problem.

The data plane aspects of this problem are discussed further in Section 4.

The control plane aspects of this problem are discussed further in Section 4.4.

The management plane aspects of this problem are discussed further in Section 4.5

Dynamic changes both to the VPN and to the underlay transport network need to be managed to avoid disruption to sensitive services.

In addition to non-disruptively managing the network as a result of gross change such as the inclusion of a new VPN endpoint or a change to a link, VPN traffic might need to be moved as a result of traffic volume changes.

## 2.5. Customized Control

In some cases it is desirable that an enhanced VPN has a customized control plane, so that the tenant of the enhanced VPN can have some control to the resources and functions allocated to this enhanced VPN. For example, the tenant may be able to specify the service paths in his own enhanced VPN. Depending on the requirement, an enhanced VPN may have its own dedicated controller, or it may be provided with an interface to a control system which is shared with a set of other tenants, or it may be provided with an interface to the control system provided by the network operator.

Further detail on this requirement will be provided in a future version of the draft. A description of the management plane aspects of this feature can be found in Section 4.5.

## 2.6. Applicability

The technologies described in this document should be applicable to a number types of VPN services such as:

- o Layer 2 point to point services such as pseudowires [RFC3985]
- o Layer 2 VPNs [RFC4664]
- o Ethernet VPNs [RFC7209]
- o Layer 3 VPNs [RFC4364], [RFC2764]
- o Virtual Networks (VNs) [RFC8453]

Where such VPN or VN types need enhanced isolation and delivery characteristics, the technology described here can be used to provide an underlay with the required enhanced performance.

## 3. Architecture of Enhanced VPN

A number of enhanced VPN services will typically be provided by a common network infrastructure. Each enhanced VPN consists of both the overlay and a specific set of dedicated network resources and functions allocated in the underlay to satisfy the needs of the VPN tenant. The integration between overlay and various underlay resources ensures the isolation between different enhanced VPNs, and achieves the guaranteed performance for different services.

An enhanced VPN needs to be designed with consideration given to:

- o A enhanced data plane

- o A control plane to create enhanced VPN, making use of the data plane isolation and guarantee techniques
- o A management plane for enhanced VPN service life-cycle management

These required characteristics are expanded below:

- o Enhanced data plane
  - \* Provides the required resource isolation capability, e.g. bandwidth guarantee.
  - \* Provides the required packet latency and jitter characteristics
  - \* Provides the required packet loss characteristics
  - \* Provides the mechanism to identify network slice and the associated resources
- o Control plane
  - \* Collect the underlying network topology and resources available and export this to other nodes and/or the centralized controller as required.
  - \* Create the required virtual networks with the resource and properties needed by the enhanced VPN services that are assigned to it.
  - \* Determine the risk of SLA violation and take appropriate avoiding action
  - \* Determine the right balance of per-packet and per-node state according to the needs of enhanced VPN service to scale to the required size
- o Management plane
  - \* Provides the life-cycle management (creation, modification, decommissioning) of enhanced VPN
  - \* Provides an interface between the enhanced VPN provider and the enhanced VPN clients such that some of the operation requests can be met without interfering with the enhanced VPN of other clients.

### 3.1. Layered Architecture

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

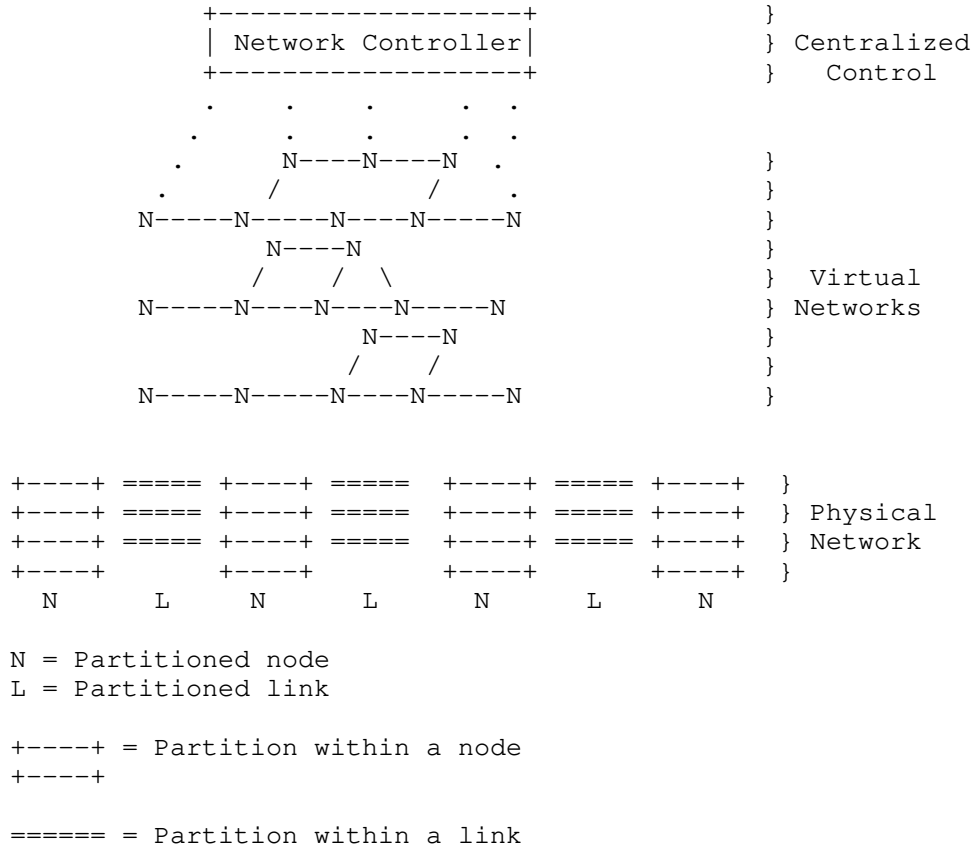


Figure 2: The Layered Architecture

Underpinning everything is the physical infrastructure layer consisting of partitioned links and nodes which provide the underlying resources used to provision the separated virtual networks. Various components and techniques as discussed in Section 4 can be used to provide the resource partition, such as FlexE, Time Sensitive Networking, Deterministic Networking, etc. These partitions may be physical, or virtual so long as the SLA required by the higher layers is met.

These techniques can be used to provision the virtual networks with dedicated resources that they need. To get the required

functionality there needs to be integration between these overlays and the underlay providing the physical resources.

The centralized controller is used to create the virtual networks, to allocate the resources to each virtual network and to provision the enhanced VPN services within the virtual networks. A distributed control plane may also be used for the distribution of the topology and attribute information of the virtual networks.

The creation and allocation process needs to take a holistic view of the needs of all of its tenants, and to partition the resources accordingly. However within a virtual network these resources can if required be managed via a dynamic control plane. This provides the required scalability and isolation.

### 3.2. Multi-Point to Multi-Point

At the VPN service level, the connectivity are usually mesh or partial-mesh. To support such kind of VPN service, the corresponding underlay is also an abstract MP2MP medium. However when service guarantees are provided, the point-to-point path through the underlay of the enhanced VPN needs to be specifically engineered to meet the required performance guarantees.

### 3.3. Application Specific Network Types

Although a lot of the traffic that will be carried over the enhanced VPN will likely be IPv4 or IPv6, the design has to 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 can be carried over the enhanced 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 two type for both PW over MPLS and for L2TPv3.

### 3.4. Scaling Considerations

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

Enhanced VPNs may need to install some additional state within the network to achieve the additional features that they require. Solutions must consider minimising and controlling the scale of such

state, and deployment architectures should constrain the number of enhanced VPNs that would exist where such services would place additional state in the network. It is expected that the number of enhanced VPN would be a small number in the beginning, and even in future the number of enhanced VPN will be much less than traditional VPNs, because traditional VPN would be enough for most existing services.

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

See Section 5 for a greater discussion of scalability considerations.

#### 4. Candidate Technologies

A VPN is a network created by applying a multiplexing technique to the underlying network (the underlay) in order 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 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 enhanced VPN service, this state will increase further.

In an enhanced VPN different subsets of the underlay resources are dedicated to different enhanced VPNs. Any enhanced VPN solution thus needs tighter coupling with underlay than is the case with existing VPNs. We cannot for example share the tunnel between enhanced VPNs which require hard isolation.

##### 4.1. Underlay Packet and Frame-Based Data Planes

A number of candidate underlay packet or frame-based data plane solutions which can be used provide the required isolation and guarantee are described in following sections.

- o FlexE

- o Time Sensitive Networking
- o Dedicated Queues

#### 4.1.1. FlexE

FlexE [FLEXE] is a method of creating a point-to-point Ethernet with a specific fixed bandwidth. FlexE provides the ability to multiplex multiple channels over an Ethernet link in a way that provides hard isolation. FlexE also supports the bonding of multiple links, which can be used to create larger links out of multiple slower links in a more efficient way than traditional link aggregation. FlexE also supports the sub-rating of links, which allows an operator to only use a portion of a link. However it 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 in turn means that it may be difficult to dynamically redistribute unused bandwidth to lower priority services. This may increase the cost of providing services on the network. On the other hand, FlexE can be used to provide hard isolation between different tenants on a shared interface. The tenant can then use other methods to manage the relative priority of their own traffic in each FlexE channel.

Methods of dynamically re-sizing FlexE channels and the implication for enhanced VPN is for further study.

#### 4.1.2. Dedicated Queues

In order to provide multiple isolated virtual networks for enhanced VPN, the conventional Diff-Serv based queuing system [RFC2475] [RFC4594] is insufficient, due to the limited number of queues which cannot differentiate between traffic of different enhanced VPNs, and the range of service classes that each need to provide to their tenants. This problem is particularly acute with an MPLS underlay due to the small number of traffic class services available. In order to address this problem and reduce the interference between enhanced VPNs, it is necessary to steer traffic of VPNs to dedicated input and output queues. Routers usually have large amount of queues and sophisticated queuing systems, which could be used or enhanced to provide the levels of isolation required by the applications of enhanced VPN. For example, on one physical interface, the queuing system can provide a set of virtual sub-interfaces, each allocated with dedicated queueing and buffer resources. Sophisticated queuing

systems of this type may be used to provide end-to-end virtual isolation between traffic of different enhanced VPNs.

#### 4.1.3. Time Sensitive Networking

Time Sensitive Networking (TSN) [TSN] is an IEEE project that is designing a method of carrying time sensitive information over Ethernet. It introduces the concept of packet scheduling where a high priority packet stream may be given a scheduled time slot thereby guaranteeing that it experiences no queuing delay and hence a reduced latency. However, when no scheduled packet arrives, its reserved time-slot is handed over to best effort traffic, thereby improving the economics of the network. The mechanisms defined in TSN can be used to meet the requirements of time sensitive services of an enhanced VPN.

Ethernet can be emulated over a Layer 3 network using a pseudowire. However the TSN 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 the following section of this document.

#### 4.2. Packet and Frame-Based Network Layer

We now consider the problem of slice differentiation and resource representation in the overlay network. The candidate technologies are:

- o Deterministic Networking
- o MPLS-TE
- o Segment Routing

##### 4.2.1. Deterministic Networking

Deterministic Networking (DetNet) [I-D.ietf-detnet-architecture] 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] do 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 packets being lost, and trims duplicate packets to prevent excessive flooding of the network



and to prevent multiple packets being delivered to the destination. It also seeks to set an upper bound on latency. The goal is not to minimize latency; the optimum upper bound paths may not be the minimum latency paths.

DetNet is based on flows. It currently does not specify the use of underlay topology other than the base topology. To be of use for enhanced VPN, DetNet needs to be integrated with different virtual topologies of enhanced VPNs.

The detailed design that allows the use DetNet in a multi-tenant network, and how to improve the scalability of DetNet in a multi-tenant network are topics for further study.

#### 4.2.2. MPLS Traffic Engineering (MPLS-TE)

MPLS-TE introduces the concept of reserving end-to-end bandwidth for a TE-LSP, which can be used as the underlay of VPNs. It also introduces the concept of non-shortest path routing through the use of the Explicit Route Object [RFC3209]. VPN traffic can be run over dedicated TE-LSPs to provide reserved bandwidth for each specific connection in a VPN. Some network operators have concerns about the scalability and management overhead of RSVP-TE system, and this has lead them to consider other solutions for their networks.

#### 4.2.3. Segment Routing

Segment Routing [RFC8402] is a method that prepends instructions to packets at the head-end node and optionally at various points as it passes through the network. These instructions allow the packets to be routed on paths other than the shortest path for various traffic engineering reasons. These paths can be strict or loose paths, depending on the compactness required of the instruction list and the degree of autonomy granted to the network, for example to support Equal Cost Multipath load-balancing (ECMP) [RFC2992].

With SR, a path needs to be dynamically created through a set of segments by simply specifying the Segment Identifiers (SIDs), i.e. instructions rooted at a particular point in the network. Thus if a path is to be provisioned from some ingress point A to some egress point B in the underlay, A is provided with a SID list from A to B and instructions on how to identify the packets to which the SID list is to be prepended.

By encoding the state in the packet, as is done in Segment Routing, per-path state is transitioned out of the network.

However, there are a number of limitations in current SR, which limit its applicability to enhanced VPNs:

- o Segments are shared between different VPNs paths
- o There is no reservation of bandwidth
- o There is limited differentiation in the data plane.

Thus some extensions to SR are needed to provide isolation between different enhanced VPNs. This can be achieved by including a finer granularity of state in the network in anticipation of its future use by authorized services. We therefore need to evaluate the balance between this additional state and the performance delivered by the network.

With current segment routing, the instructions are used to specify the nodes and links to be traversed. However, in order to achieve the required isolation between different services, new instructions can be created which can be prepended to a packet to steer it through specific network resources and functions.

Traditionally an SR traffic engineered path operates with a granularity of a link with hints about priority provided through the use of the traffic class (TC) field in the header. However to achieve the latency and isolation characteristics that are sought by the enhanced VPN users, steering packets through specific queues and resources will likely be required. The extent to which these needs can be satisfied through existing QoS mechanisms is to be determined. What is clear is that a fine control of which services wait for which, with a fine granularity of queue management policy is needed. Note that the concept of a queue is a useful abstraction for many types of underlay mechanism that may be used to provide enhanced isolation and latency support.

From the perspective of the control plane, and from the perspective of the segment routing, the method of steering a packet to a queue that provides the required properties is an abstraction that hides the details of the underlying implementation. How the queue satisfies the requirement is implementation specific and is transparent to the control plane and data plane mechanisms used. Thus, for example, a FlexE channel, or a time sensitive networking packet scheduling slot are abstracted to the same concept and bound to the data plane in a common manner.

We can also introduce such fine grained packet steering by specifying the queues through an SR instruction list. Thus new SR instructions may be created to specify not only which resources are traversed, but

in some cases how they are traversed. For example, it may be possible to specify not only the queue to be used but the policy to be applied when enqueueing and dequeuing.

This concept could be further generalized, since as well as queuing to the output port of a router, it is possible to consider queuing data to any resource, for example:

- o A network processor unit (NPU)
- o A central processing unit (CPU) Core
- o A Look-up engine

Both SR-MPLS and SRv6 are candidate network layer technologies for enhanced VPN. In some cases they can be supported by DetNet to meet the packet loss, delay and jitter requirement of particular service. However, currently the "pure" IP variant of DetNet [I-D.ietf-detnet-dp-sol-ip] does not support the Packet Replication, Elimination, and Re-ordering (PREOF) [I-D.ietf-detnet-architecture] functions. How to provide the DetNet enhanced delivery in an SRv6 environment needs further study.

#### 4.3. Non-Packet Technologies

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 (and hence slices).

The control and management planes for non-packet data plane technologies have most in common with MPLS-TE (Section 4.2.2) and offer the same set of advanced features [RFC3945]. Furthermore, management techniques such as ACTN ([RFC8453] and Section 4.4) can be used to aid in the reporting of underlying network topologies, and the creation of virtual networks with the resource and properties needed by the enhanced VPN services.

#### 4.4. Control Plane

Enhanced VPN would likely be based on a hybrid control mechanism, which takes advantage of the logically centralized controller for on-demand provisioning and global optimization, whilst still relies on distributed control plane to provide scalability, high reliability, fast reaction, automatic failure recovery etc. Extension and

optimization to the distributed control plane is needed to support the enhanced properties of VPN+.

RSVP-TE provides the signaling mechanism of establishing a TE-LSP with end-to-end resource reservation. It can be used to bind the VPN to specific network resource allocated within the underlay, but there are the above mentioned scalability concerns.

SR does not have the capability of signaling the resource reservation along the path, nor do its currently specified distributed link state routing protocols. On the other hand, the SR approach provides a way of efficiently binding the network underlay and the enhanced VPN overlay, as it reduces the amount of state to be maintained in the network. An SR-based approach with per-slice resource reservation can easily create dedicated SR network slices, and the VPN services can be bound to a particular SR network slice. A centralized controller can perform resource planning and reservation from the controller's point of view, but this does not ensure resource reservation is actually done in the network nodes. Thus, if a distributed control plane is needed, either in place of an SDN controller or as an assistant to it, the design of the control system needs to ensure that resources are uniquely allocated in the network nodes for the correct service, and not allocated to multiple services causing unintended resource conflict.

#### 4.5. Management Plane

The management plane mechanisms for enhanced VPN can be based on the VPN service models as defined in [RFC8299] and [RFC8466], possible augmentations and extensions to these models may be needed, which is out of the scope of this document.

Abstraction and Control of Traffic Engineered Networks (ACTN) [RFC8453] specifies the SDN based architecture for the control of TE networks. The ACTN related data models such as [I-D.ietf-teas-actn-vn-yang] and [I-D.lee-teas-te-service-mapping-yang] can be applicable in the provisioning of enhanced VPN service. The details are described in Section 4.6.

#### 4.6. Applicability of ACTN to Enhanced VPN

ACTN facilitates end-to-end connections and provides them to the user. The ACTN framework [RFC8453] highlights how:

- o Abstraction of the underlying network resources are provided to higher-layer applications and customers.

- o Virtualization of underlying resources, whose selection criterion is the allocation of those resources for the customer, application, or service.
- o Creation of a virtualized environment allowing operators to view and control multi-domain networks as a single virtualized network.
- o The presentation to customers of networks as a virtual network via open and programmable interfaces.

The infrastructure managed through ACTN comprises traffic engineered network resources, which may include:

- o Statistical packet bandwidth.
- o Physical forwarding plane sources, such as: wavelengths and time slots.
- o Forwarding and cross-connect capabilities.

The type of network virtualization enabled by ACTN provides customers and applications (tenants) with the capability to utilize and independently control allocated virtual network resources as if they were physically their own resources.

An ACTN Virtual Network (VN) is a client view of the ACTN managed infrastructure, and is presented by the ACTN provider as a set of abstracted resources.

Depending on the agreement between client and provider various VN operations and VN views are possible.

- o Virtual Network Creation: A VN could be pre-configured and created via static or dynamic request and negotiation between customer and provider. It must meet the specified SLA attributes which satisfy the customer's objectives.
- o Virtual Network Operations: The virtual network may be further modified and deleted based on customer request to request changes in the network resources reserved for the customer, and used to construct the network slice. The customer can further act upon the virtual network to manage traffic flow across the virtual network.
- o Virtual Network View: The VN topology from a customer point of view. These may be a variety of tunnels, or an entire VN topology. Such connections may comprise of customer end points, access links, intra-domain paths, and inter-domain links.

Dynamic VN Operations allow a customer to modify or delete the VN. The customer can further act upon the virtual network to create/modify/delete virtual links and nodes. These changes will result in subsequent tunnel management in the operator's networks.

#### 4.6.1. ACTN Used for VPN+ Delivery

ACTN provides VPN connections between multiple sites as requested via a VPN requestor enabled by the Customer Network Controller (CNC). The CNC is managed by the customer themselves, and interacts with the network provider's Multi-Domain Service Controller (MDSC). The Provisioning Network Controllers (PNC) remain entirely under the management of the network provider and are not visible to the customer.

The benefits of this model include:

- o Provision of edge-to-edge VPN multi-access connectivity.
- o Management is mostly performed by the network provider, with some flexibility delegated to the customer-managed CNC.

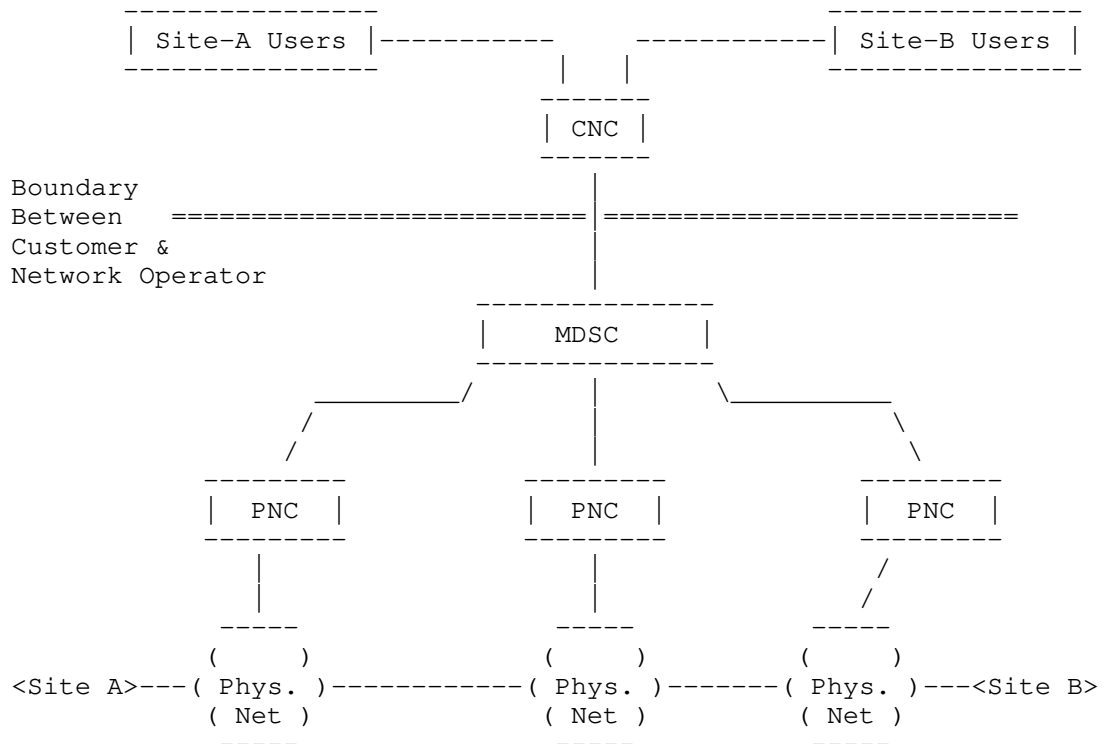


Figure 3: VPN Delivery in the ACTN Architecture

Figure 4 presents a more general representation of how multiple enhanced VPNs may be created from the resources of multiple physical networks using the CNC, MDSC, and PNC components of the ACTN architecture. Each enhanced VPN is controlled by its own CNC. The CNCs send requests to the provider's MDSC. The provider manages two physical networks each under the control of PNC. The MDSC asks the PNCs to allocate and provision resources to achieve the enhanced VPNs. In this figure, one enhanced VPN is constructed solely from the resources of one of the physical networks, while the the VPN uses resources from both physical networks.

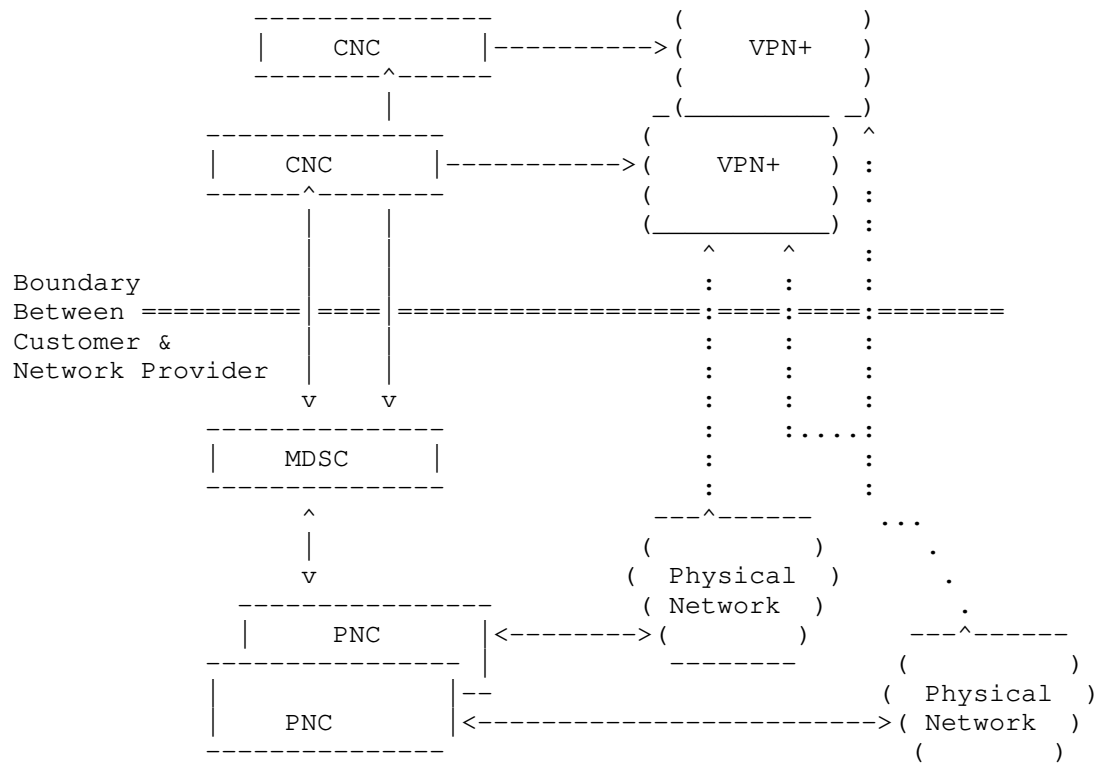


Figure 4: Generic VPN+ Delivery in the ACTN Architecture

#### 4.6.2. Enhanced VPN Features with ACTN

This section discusses how the features of ACTN can fulfill the enhanced VPN requirements described earlier in this document. As previously noted, key requirements of the enhanced VPN include:

1. Isolation between VPNs
2. Guaranteed Performance
3. Integration
4. Dynamic Configuration
5. Customized Control Plane

The subsections that follow outline how each requirement is met using ACTN.



#### 4.6.2.1. Isolation Between VPNs

The ACTN VN YANG model [I-D.ietf-teas-actn-vn-yang] and the TE-service mapping model [I-D.lee-teas-te-service-mapping-yang] fulfill the VPN isolation requirement by providing the following features for the VNs:

- o Each VN is identified with a unique identifier (vn-id and vn-name) and so is each VN member that belongs to the VN (vn-member-id).
- o Each instantiated VN is managed and controlled independent of other VNs in the network with proper protection level (protection).
- o Each VN is instantiated with an isolation requirement described by the TE-service mapping model [I-D.lee-teas-te-service-mapping-yang]. This mapping supports:
  - \* Hard isolation with deterministic characteristics (e.g., this case may need an optical bypass tunnel or a DetNet/TSN tunnel to guarantee latency with no jitter)
  - \* Hard isolation (i.e., dedicated TE resources in all underlays)
  - \* Soft isolation (i.e., resource in some layer may be shared while in some other layers is dedicated).
  - \* No isolation (i.e., sharing with other VN).

#### 4.6.2.2. Guaranteed Performance

Performance objectives of a VN need first to be expressed in order to assure the performance guarantee. [I-D.ietf-teas-actn-vn-yang] and [I-D.ietf-teas-yang-te-topo] allow configuration of several parameters that may affect the VN performance objectives as follows:

- o Bandwidth
- o Objective function (e.g., min cost path, min load path, etc.)
- o Metric Types and their threshold:
  - \* TE cost, IGP cost, Hop count, or Unidirectional Delay (e.g., can set all path delay <= threshold)

Once these requests are instantiated, the resources are committed and guaranteed through the life cycle of the VN.

#### 4.6.2.3. Integration

ACTN provides mechanisms to correlate customer's VN and the actual TE tunnels instantiated in the provider's network. Specifically:

- o Link each VN member to actual TE tunnel.
- o Each VN can be monitored on a various level such as VN level, VN member level, TE-tunnel level, and link/node level.

Service function integration with network topology (L3 and TE topology) is in progress in [I-D.ietf-teas-sf-aware-topo-model]. Specifically, [I-D.ietf-teas-sf-aware-topo-model] addresses a number of use-cases that show how TE topology supports various service functions.

#### 4.6.2.4. Dynamic Configuration

ACTN provides an architecture that allows the CNC to interact with the MDSC which is network provider's SDN controller. This gives the customer control of their VNs.

Specifically, the ACTN VN model [I-D.ietf-teas-actn-vn-yang] allows the VN to create, modify, and delete VNs.

#### 4.6.2.5. Customized Control

ACTN provides a YANG model that allows the CNC to control a VN as a "Type 2 VN" that allows the customer to provision tunnels that connect their endpoints over the customized VN topology.

For some VN members, the customers are allowed to configure the path (i.e., the sequence of virtual nodes and virtual links) over the VN/abstract topology.

### 5. Scalability Considerations

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

- o 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 we more precisely specify the path and resources that need to be exclusively available to a VPN.

- o Introduce the state to the network. This is normally done by creating a path using RSVP-TE, which can be extended to introduce any element that needs to be specified along the path, for example explicitly specifying queuing policy. It is of course possible to use other methods to introduce path state, such as via a Software Defined Network (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 its life-cycle. This is more state than is needed using SR, but the packet are shorter.
- o Provide a hybrid approach based on using binding SIDs to create path fragments, and bind them together with SR.

Dynamic creation of a VPN path using SR requires less state maintenance in the network core at the expense of larger VPN headers on the packet. 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 resource on the routers are specified. Reducing the state in the network is important to enhanced 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 needed 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, a segment routed approach allows much of this state to be spread amongst the network ingress nodes, and transiently carried in the packets as SIDs.

These approaches are for further study.

#### 5.1. Maximum Stack Depth of SR

One of the challenges with SR is the stack depth that nodes are able to impose on packets [I-D.ietf-isis-segment-routing-msd]. This leads to a difficult balance between adding state to the network and minimizing stack depth, or minimizing state and increasing the stack depth.

#### 5.2. RSVP Scalability

The traditional method of creating a resource allocated path through an MPLS network is to use the RSVP protocol. However there have been concerns that this requires significant continuous state maintenance

in the network. There are ongoing works to improve the scalability of RSVP-TE LSPs in the control plane [RFC8370].

There is also concern at the scalability of the forwarder footprint of RSVP as the number of paths through an LSR grows [I-D.sitaraman-mpls-rsvp-shared-labels] proposes to address this by employing SR within a tunnel established by RSVP-TE.

## 6. OAM Considerations

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

The enhanced VPN OAM design needs to consider the following requirements:

- o 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.
- o 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.
- o 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
- o Verification of the conformity of the path to the service requirement. This may need to be done as part of a commissioning test.

These issues will be discussed in a future version of this document.

## 7. Enhanced Resiliency

Each enhanced VPN has a life-cycle, and needs modification during deployment as the needs of its tenant change. Additionally, as the network as a whole evolves, there will 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. This is a single action by the head-end 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 up and meet the required SLA before traffic is transitioned to it. It is possible for deadlocks arise as a result of the network becoming fragmented over time, such that it is impossible to create a new path or modify a 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:

- o The problem of packets overtaking one and other if a path latency reduces during a transition.
- o The problem of the latency transient 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 proposed by the IETF deterministic network work with multiple in-network replication and the culling of later packets [I-D.ietf-detnet-architecture].

In addition to the approach used to protect high priority packets, consideration has to be given to the impact of best effort traffic on the high priority packets during a transient. 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.

## 8. Security Considerations

All types of virtual network require special consideration to be given to the isolation between the tenants. In this regard enhanced VPNs neither introduce, nor experience a greater security risk than another VPN of the same base type. However, in an enhanced virtual network service the isolation requirement needs to be considered. If a service requires a specific latency then it can be damaged by

simply delaying the packet through the activities of another tenant. In a network with virtual functions, depriving a function used by another tenant of compute resources can be just as damaging as delaying transmission of a packet in the network. The measures to address these dynamic security risks must be specified as part to the specific solution.

## 9. IANA Considerations

There are no requested IANA actions.

## 10. Contributors

Daniel King  
Email: daniel@olddog.co.uk

Adrian Farrel  
Email: adrian@olddog.co.uk

Jeff Tansura  
Email: jefftant.ietf@gmail.com

Qin Wu  
Email: bill.wu@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

## 11. Acknowledgements

The authors would like to thank Charlie Perkins, James N Guichard and John E Drake 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).

## 12. References

### 12.1. 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>>.

### 12.2. Informative References

- [BBF-SD406] "BBF SD-406: End-to-End Network Slicing", 2016, <<https://wiki.broadband-forum.org/display/BBF/SD-406+End-to-End+Network+Slicing>>.
- [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.ietf-detnet-architecture] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", draft-ietf-detnet-architecture-11 (work in progress), February 2019.
- [I-D.ietf-detnet-dp-sol-ip] Korhonen, J. and B. Varga, "DetNet IP Data Plane Encapsulation", draft-ietf-detnet-dp-sol-ip-01 (work in progress), October 2018.
- [I-D.ietf-detnet-use-cases] Grossman, E., "Deterministic Networking Use Cases", draft-ietf-detnet-use-cases-20 (work in progress), December 2018.
- [I-D.ietf-isis-segment-routing-msd] Tantsura, J., Chunduri, U., Aldrin, S., and L. Ginsberg, "Signaling MSD (Maximum SID Depth) using IS-IS", draft-ietf-isis-segment-routing-msd-19 (work in progress), October 2018.

- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., Yoon, B., Wu, Q., and P. Park, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-04 (work in progress), February 2019.
- [I-D.ietf-teas-sf-aware-topo-model]  
Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-02 (work in progress), September 2018.
- [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-19 (work in progress), February 2019.
- [I-D.lee-teas-te-service-mapping-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Tantsura, J., Fioccola, G., and Q. Wu, "Traffic Engineering and Service Mapping Yang Model", draft-lee-teas-te-service-mapping-yang-13 (work in progress), December 2018.
- [I-D.sitaraman-mpls-rsvp-shared-labels]  
Sitaraman, H., Beeram, V., Parikh, T., and T. Saad, "Signaling RSVP-TE tunnels on a shared MPLS forwarding plane", draft-sitaraman-mpls-rsvp-shared-labels-03 (work in progress), December 2017.
- [NGMN-NS-Concept]  
"NGMN NS Concept", 2016, <[https://www.ngmn.org/fileadmin/user\\_upload/161010\\_NGMN\\_Network\\_Slicing\\_framework\\_v1.0.8.pdf](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>>.
- [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>>.
- [RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <<https://www.rfc-editor.org/info/rfc2992>>.



- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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](mailto:jie.dong@huawei.com)

Stewart Bryant  
Huawei

Email: [stewart.bryant@gmail.com](mailto:stewart.bryant@gmail.com)

Zhenqiang Li  
China Mobile

Email: [lizhenqiang@chinamobile.com](mailto:lizhenqiang@chinamobile.com)

Takuya Miyasaka  
KDDI Corporation

Email: [ta-miyasaka@kddi.com](mailto:ta-miyasaka@kddi.com)

Young Lee  
Huawei

Email: leeyoung@huawei.com

TEAS Working Group  
Internet Draft  
Category: Informational

Haomian Zheng  
Xianlong Luo  
Yi Lin  
Huawei Technologies  
Yang Zhao  
China Mobile  
Yunbin Xu  
CAICT  
Sergio Belotti  
Dieter Beller  
Nokia  
November 4, 2019

Expires: May 4, 2020

## Interworking of GMPLS Control and Centralized Controller System

draft-ietf-teas-gmpls-controller-inter-work-02

### 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 May 4, 2020.

#### 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. Discovery Options .....	6
3.1. LMP .....	6
4. Routing Options .....	6
4.1. OSPF-TE .....	7
4.2. ISIS-TE .....	7
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) .....	8
6. Signaling Options .....	8
6.1. RSVP-TE .....	8





7. Interworking Scenarios .....	9
7.1. Topology Collection & Synchronization .....	9
7.2. Multi-domain Service Provisioning .....	9
7.3. Multi-layer Service Provisioning .....	12
7.4. Recovery .....	12
7.5. Controller Reliability .....	12
8. Manageability Considerations .....	13
9. Security Considerations .....	13
10. IANA Considerations.....	13
11. References .....	13
11.1. Normative References .....	13
11.2. Informative References .....	15
12. Authors' Addresses .....	17

## 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, a 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).

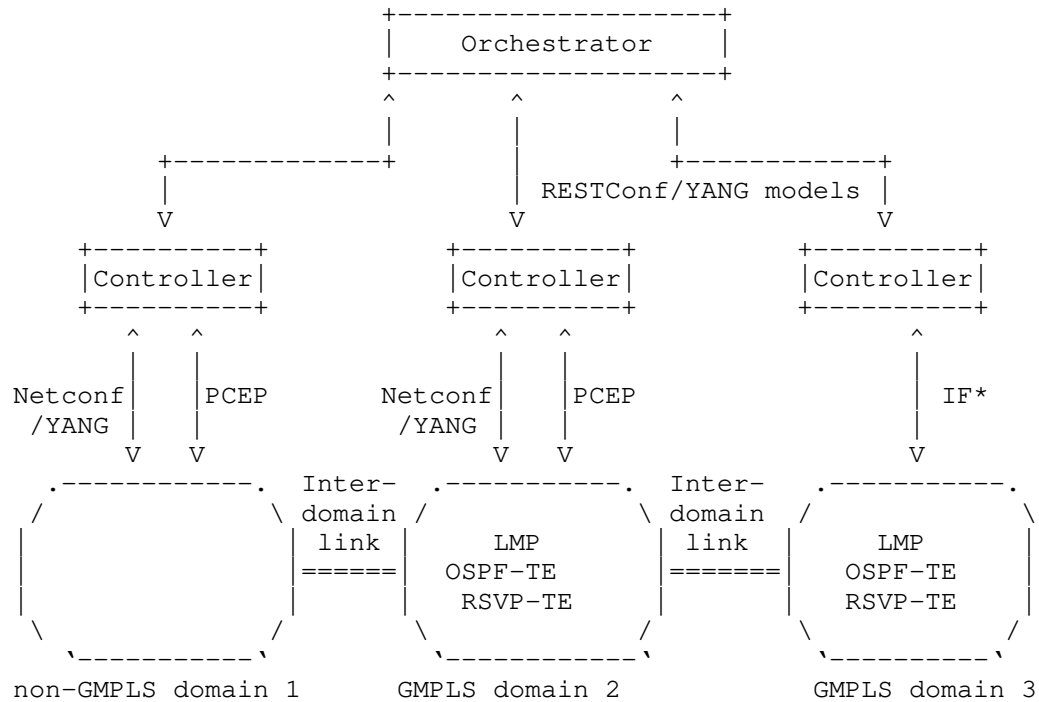


Figure 1: Example of GMPLS/non-GMPLS interworks with Controllers

Figure 1 shows the scenario with two GMPLS domains and one non-GMPLS domain. This system supports the interworking among non-GMPLS domain, GMPLS domain and the controller hierarchies. For domain 1, the network element were not enabled with GMPLS so the control can be purely from the controller, via Netconf/YANG and/or PCEP. For domain 2 and 3, 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. Discovery Options

In GMPLS control, the link connectivity need to be verified between each pair of nodes. In this way, link resources, which are fundamental resources in the network, are discovered by both ends of the link.

#### 3.1. LMP

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.

### 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 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 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 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. Several methods can be used to create the inter-domain path:

#### 1) With end-to-end RSVP-TE session:

In this method, the SDN controller of the source domain triggers the source node to create the end-to-end RSVP-TE session, and the assignment and distribution of the labels on the inter-domain links are done by the boarder nodes of each domain, using RSVP-TE



protocol. Therefore, this method requires the interworking of RSVP-TE protocols between different domains.

There are two possible methods:

#### 1.1) One single end-to-end RSVP-TE session

In this method, an end-to-end RSVP-TE session from the source NE to the destination NE will be used to create the inter-domain path. 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.

#### 1.2) LSP Stitching

The LSP stitching method defined in [RFC5150] can also be used to create the end-to-end LSP. I.e., when the source node receives an end-to-end path creation request (e.g., using PCEP or Netconf protocol), the source node starts an end-to-end RSVP-TE session along the end points of each LSP segment (refers to S-LSP in [RFC5150]) of each domain, to assign the labels on the inter-domain links between each pair of neighbor S-LSPs, and stitch the end-to-end LSP to each S-LSP. See Figure 2 as an example. Note that the S-LSP in each domain can be either created by each domain controller in advance, or created dynamically triggered by the end-to-end RSVP-TE session.

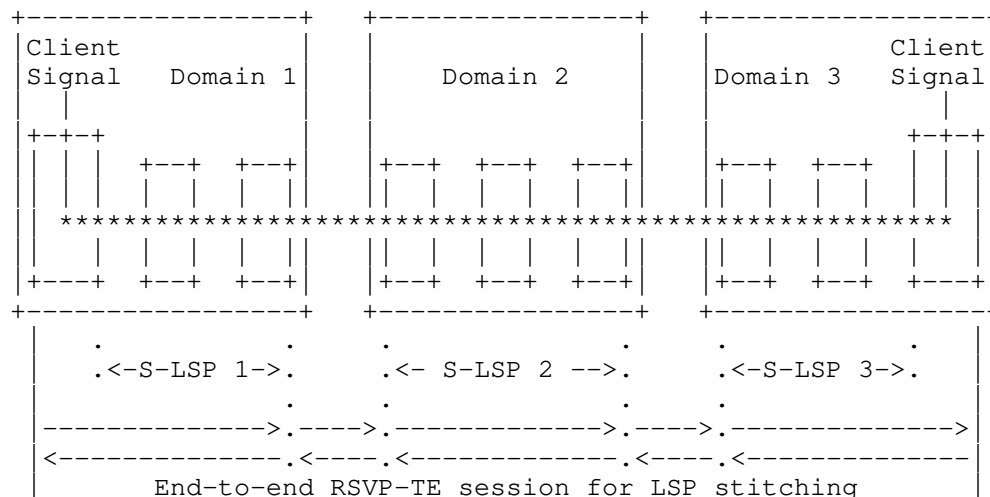


Figure 2: LSP stitching





## 2) Without end-to-end RSVP-TE session:

In this method, each SDN controller is responsible to create the path segment within its domain. The boarder node does not need to communicate with other boarder nodes in other domains for the distribution of labels on inter-domain links, so end-to-end RSVP-TE session through multiple domains is not required, and the interworking of RSVP-TE protocol between different domains is not needed.

Note that path segments in the source domain and the destination domain are "asymmetrical" segments, because the configuration of client signal mapping into server layer tunnel is needed at only one end of the segment, while configuration of server layer cross-connect is needed at the other end of the segment. For example, the path segment 1 and 3 in Figure 3 are asymmetrical segments, because one end of the segment requires mapping GE into ODU0, while the other end of the segment requires setting up ODU0 cross-connect.

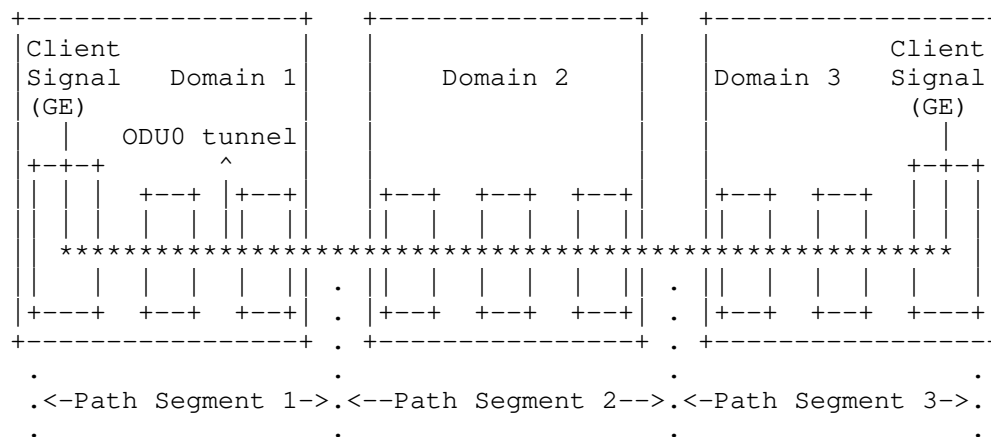


Figure 3: Example of asymmetrical path segment

The PCEP / GMPLS protocols should support creation of such asymmetrical segment.

Note also that mechanisms to assign the labels in the inter-domain links are also needed to be considered. There are two possible methods:

### 2.1) Inter-domain labels assigned by NEs:

The concept of Stitching Label that allows stitching local path segments was introduced in [RFC5150] and [sPCE-ID], in order to form the inter-domain path crossing several different domains. It also describes the BRPC and H-PCE PCInitiate procedure, i.e., the ingress



boarder node of each downstream domain assigns the stitching label for the inter-domain link between the downstream domain and its upstream neighbor domain, and this stitching label will be passed to the upstream neighbor domain by PCE protocol, which will be used for the path segment creation in the upstream neighbor domain.

#### 2.2) Inter-domain labels assigned by SDN controller:

If the resource of inter-domain links are managed by the multi-domain SDN controller, each single-domain SDN controller can provide to the multi-domain SDN controller the list of available labels (e.g. timeslots if OTN is the scenario) using IETF Topology model and related technology specific extension. Once that multi-domain SDN controller has computed e2e path RSVP-TE or PCEP can be used in the different domains to setup related segment tunnel consisting with label inter-domain information, e.g. for PCEP the label ERO can be included in the PCInitiate message to indicate the inter-domain labels, so that each boarder node of each domain can configure the correct cross-connect within itself.

### 7.3. Multi-layer Service Provisioning

For further study. Plan to be updated in the next version.

### 7.4. 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.

### 7.5. 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.
- [RFC5150] Ayyangar, A., Kompella, K., Vasseur, J.P., Farrel, A., "Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)", RFC 5150, February, 2008.
- [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.
- [sPCE-ID] Dugeon, O. et al., "PCEP Extension for Stateful Inter-Domain Tunnels", draft-dugeon-pce-stateful-interdomain, 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@caict.ac.cn

Yang Zhao  
China Mobile  
Email: zhaoyangjy@chinamobile.com

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

Dieter Beller  
Nokia  
Email: Dieter.Beller@nokia.com

Yi Lin  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base,  
Bantian, Longgang District,  
Shenzhen 518129 P.R.China  
Email: yi.lin@huawei.com





TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: March 12, 2020

Y. Lee, Ed.  
SKKU  
D. Dhody, Ed.  
G. Fioccola  
Q. Wu, Ed.  
Huawei Technologies  
D. Ceccarelli  
Ericsson  
J. Tantsura  
Apstra  
September 9, 2019

Traffic Engineering (TE) and Service Mapping Yang Model  
draft-ietf-teas-te-service-mapping-yang-02

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). This model is referred to as TE Service Mapping Model and is applicable generically to the operator's need for seamless control and management of their VPN services with TE tunnel support.

The model is principally used to allow monitoring and diagnostics of the management systems to show how the service requests are mapped onto underlying network resource and TE models.

Status of This Memo

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 March 12, 2020.



## 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 . . . . .	3
1.1. Terminology . . . . .	4
1.2. Tree diagram . . . . .	4
1.3. Prefixes in Data Node Names . . . . .	4
2. TE and Service Related Parameters . . . . .	5
2.1. VN/Tunnel Selection Requirements . . . . .	5
2.2. Availability Requirement . . . . .	6
3. YANG Modeling Approach . . . . .	7
3.1. Forward Compatibility . . . . .	8
4. L3VPN Architecture in the ACTN Context . . . . .	8
4.1. Service Mapping . . . . .	11
4.2. Site Mapping . . . . .	11
5. Applicability of TE-Service Mapping in Generic context . . . . .	12
6. YANG Data Trees . . . . .	12
6.1. L3SM . . . . .	12
6.2. L2SM . . . . .	13
6.3. L1CSM . . . . .	14
7. YANG Data Models . . . . .	15
7.1. ietf-te-service-mapping-types . . . . .	15
7.2. ietf-l3sm-te-service-mapping . . . . .	21
7.3. ietf-l2sm-te-service-mapping . . . . .	23
7.4. ietf-l1csm-te-service-mapping . . . . .	25
8. Security Considerations . . . . .	27
9. IANA Considerations . . . . .	28
10. Acknowledgements . . . . .	29
11. References . . . . .	29
11.1. Normative References . . . . .	29
11.2. Informative References . . . . .	31
Appendix A. Contributor Addresses . . . . .	32
Authors' Addresses . . . . .	32

## 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 modelling 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]:

- o 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.

- o 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 generic service models. Section 3 discusses YANG modelling approach.

#### 1.1. 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].

#### 1.2. Tree diagram

A simplified graphical representation of the data model is used in Section 5 of this this document. The meaning of the symbols in these diagrams is defined in [RFC8340].

#### 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
tsm- types l1csm	ietf-te-service-mapping- types ietf-l1csm	[RFCXXXX] [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-yan g]
nw	ietf-network	[RFC8345]
te- types	ietf-te-types	[I-D.ietf-teas-yang-te-ty pes]
te	ietf-te	[I-D.ietf-teas-yang-te]

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 TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example,

for network slicing. The establishment of TE tunnels is subject to the network operator's policies.

To summarize, there are three modes of VN/Tunnel selection operations to be supported as follows. Additional modes may be defined in the future.

- o New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN. Under this mode, the following sub-categories can be supported:
  1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.
  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.
  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.
- o 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.
- o VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

## 2.2. Availability Requirement

Availability is another service requirement or intent that may influence the selection or provisioning of TE tunnels or a VN to support the requested service. Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure.

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this 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-llcsm-yang], [RFC8466], and [RFC8299]). Figure 1 shows the scope of the Augmented LxSM Model.

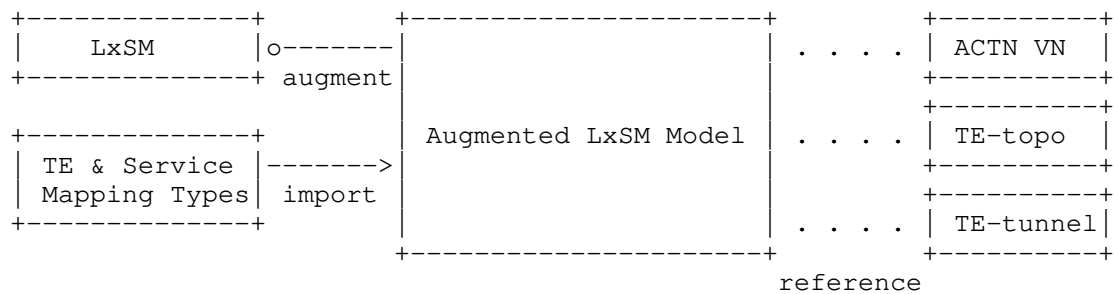


Figure 1: Augmented LxSM Model

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 role of the augmented LxSm service model is to expose the mapping relationship between service models and TE models so that VN/VPN service instantiations provided by the underlying TE networks can be viewed outside of the MDSC, for example by an operator who is diagnosing the behaviour of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various

constraints via TE-topology model [I-D.ietf-teas-yang-te-topo], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

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

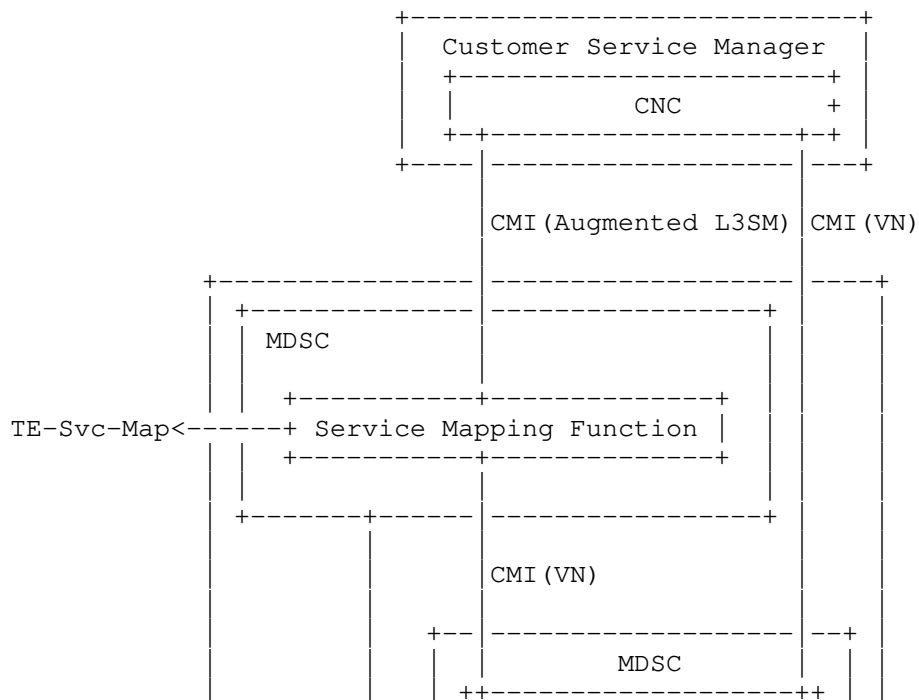
### 3.1. Forward Compatibility

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.

## 4. L3VPN Architecture in the ACTN Context

Figure 2 shows the architectural context of this document referencing the ACTN components and interfaces.



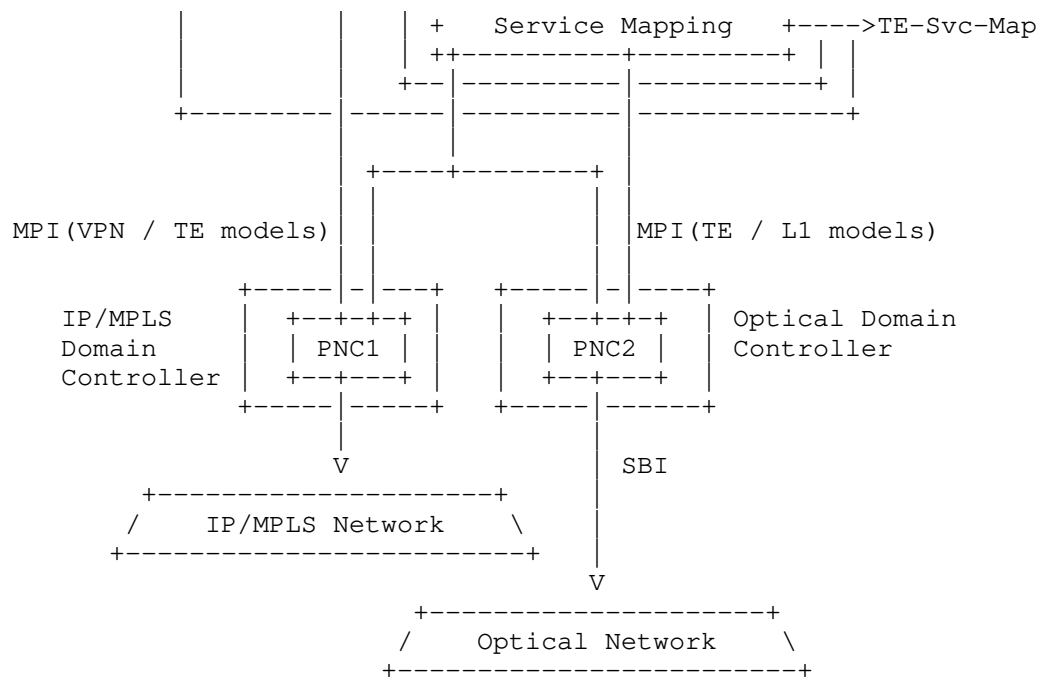


Figure 2: L3VPN Architecture from the IP+Optical Network Perspective

There are three main entities in the ACTN architecture and shown in Figure 2.

- o 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.
- o 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.



- o PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 2 shows two distinct PNCs.
- \* IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.
- \* Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

There are four main interfaces shown in Figure 2.

- o 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).
- o 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.
- o SBI: The Southbound Interface is used by the PNC to control network devices and is out of scope for this document.

The TE Service Mapping Model as described in this document can be used to see the mapping between service models and VN models and TE Tunnel/Topology models. That mapping may occur in the CNC if a service request is mapped to a VN request. Or it may occur in the MDSC where a service request is mapped to a TE tunnel, TE topology, or VPN network configuration model. The TE Service Mapping Model may be read from the CNC or MDSC to understand how the mapping has been made and to see the purpose for which network resources are used.

As shown in Figure 2, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

The high-level control flows for one example are as follows:

1. A customer asks for an L3VPN between CE1 and CE2 using the Augmented L3SM model.

2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [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. 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 availability-type? identityref
        +--rw (te)?
          +--:(vn)
            | +--rw vn-ref?          -> /vn:vn/vn-list/vn-id
          +--:(te-topo)
            | +--rw vn-topology-id?  te-types:te-topology-id
            | +--rw abstract-node?
            |   -> /nw:networks/network/node/node-id
          +--:(te-tunnel)
            +--rw te-tunnel-list*    te:tunnel-ref
augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
  /l3vpn-svc:site-network-accesses
  /l3vpn-svc:site-network-access:
    +--rw (te)?
      +--:(vn)
        | +--rw vn-ref?
        |   -> /vn:ap/access-point-list/access-point-id
      +--:(te)
        +--rw ltp?                  te-types:te-tp-id

```

## 6.2. 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 availability-type?  identityref
        +--rw (te)?
          +--:(vn)
            | +--rw vn-ref?          -> /vn:vn/vn-list/vn-id
          +--:(te-topo)
            | +--rw vn-topology-id?  te-types:te-topology-id
            | +--rw abstract-node?
            |   -> /nw:networks/network/node/node-id
          +--:(te-tunnel)
            +--rw te-tunnel-list*    te:tunnel-ref
augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
  /l2vpn-svc:site-network-accesses
  /l2vpn-svc:site-network-access:
    +--rw (te)?
      +--:(vn)
        | +--rw vn-ref?
        |   -> /vn:ap/access-point-list/access-point-id
      +--:(te)
        +--rw ltp?          te-types:te-tp-id

```

### 6.3. 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 availability-type?  identityref
        +--rw (te)?
          +--:(vn)
            | +--rw vn-ref?          -> /vn:vn/vn-list/vn-id
          +--:(te-topo)
            | +--rw vn-topology-id?  te-types:te-topology-id
            | +--rw abstract-node?
            |   -> /nw:networks/network/node/node-id
          +--:(te-tunnel)
            +--rw te-tunnel-list*    te:tunnel-ref
  augment /llcsm:ll-connectivity/llcsm:access/llcsm:unis/llcsm:uni:
    +--rw (te)?
      +--:(vn)
        | +--rw vn-ref?
        |   -> /vn:ap/access-point-list/access-point-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@2019-09-09.yang"

```

module ietf-te-service-mapping-types {

  yang-version 1.1;

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

  prefix tsm;

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

  import ietf-network {
    prefix nw;
  }

```

```
reference
  "RFC 8345: A YANG Data Model for Network Topologies";
}

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

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

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:     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) 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.";

revision 2019-09-09 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * 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 for availability-type
 */

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

identity level-1 {
  base availability-type;
  description
    "level 1: 99.9999%";
}

identity level-2 {
  base availability-type;
  description
    "level 2: 99.999%";
}

identity level-3 {
  base availability-type;
  description
    "level 3: 99.99%";
}

identity level-4 {
  base availability-type;
  description
    "level 4: 99.9%";
}

identity level-5 {
  base availability-type;
  description
    "level 5: 99%";
}

/*
 * Groupings
```

```
*/

grouping te-ref {
  description
    "The reference to TE.";
  choice te {
    description
      "The TE";
    case vn {
      leaf vn-ref {
        type leafref {
          path "/vn:vn/vn:vn-list/vn:vn-id";
        }
        description
          "The reference to VN";
        reference
          "RFC 8453: Framework for Abstraction and Control of TE
            Networks (ACTN)";
      }
    }
  }
  case te-topo {
    leaf vn-topology-id {
      type te-types:te-topology-id;
      description
        "An identifier to the TE Topology Model where the abstract
          nodes and links of the Topology can be found for Type 2
          VNS";
      reference
        "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
          Engineering (TE) Topologies";
    }
    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
        "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
          Engineering (TE) Topologies";
    }
  }
}
case te-tunnel {
  leaf-list te-tunnel-list {
    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";
    }
}
}
} //grouping

grouping te-endpoint-ref {
    description
        "The reference to TE endpoints.";
    choice te {
        description
            "The TE";
        case vn {
            leaf vn-ref {
                type leafref {
                    path "/vn:ap/vn:access-point-list/vn:access-point-id";
                }
                description
                    "The reference to VN AP";
                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
                    "I-D.ietf-teas-yang-te-topo: YANG Data Model for Traffic
                    Engineering (TE) Topologies";
            }
        }
    }
} //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";
    }
    leaf availability-type {
        type identityref {
            base availability-type;
        }
        description
            "Availability Requirement for the Service";
    }
    uses te-ref;
}
} // grouping
} // module
```

<CODE ENDS>

## 7.2. ietf-l3sm-te-service-mapping

```
<CODE BEGINS> file "ietf-l3sm-te-service-mapping@2019-09-09.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 tsm-types;
        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:  <http://tools.ietf.org/wg/teas/>
        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) 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.";

revision 2019-09-09 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * 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 tsm-types: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 tsm-types:te-endpoint-ref;
} //augment
} //module
```

<CODE ENDS>

### 7.3. ietf-l2sm-te-service-mapping

```
<CODE BEGINS> file "ietf-l2sm-te-service-mapping@2019-09-09.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 tsm-types;
    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:  <http://tools.ietf.org/wg/teas/>
    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) 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."

revision 2019-09-09 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Traffic Engineering and Service Mapping Yang Model";
}

/*
 * Augmentation to L3SM
 */
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 tsm-types:te-mapping;
  }
}

augment "/l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site"
  + "l2vpn-svc:site-network-accesses"
  + "l2vpn-svc:site-network-access" {
  description
    "This augment is only valid for TE mapping of L2SM network-access
    to TE endpoints";
```

```
    uses tsm-types:te-endpoint-ref;
  }//augment
}//module
<CODE ENDS>
```

#### 7.4. ietf-llcsm-te-service-mapping

```
<CODE BEGINS> file "ietf-llcsm-te-service-mapping@2019-09-09.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 tsm-types;
    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:    <http://tools.ietf.org/wg/teas/>
    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) 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.";

```
revision 2019-09-09 {
  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 tsm-types:te-mapping;
  }
} //augment

augment "/l1csm:l1-connectivity/l1csm:access/l1csm:unis/"
  + "l1csm:uni" {
  description
    "This augment is only valid for TE mapping of L1CSM UNI to TE
    endpoints";
  uses tsm-types:te-endpoint-ref;
} //augment
} //module

<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:

- o /l3vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- o /l3vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.
- o /l2vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- o /l2vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.
- o /l1-connectivity/services/service/te-service-mapping/te-mapping/ - configure TE Service mapping.
- o /l1-connectivity/access/unis/uni/te/ - 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 four 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-l1csm-te-service-mapping  
Registrant Contact: The IESG.  
XML: N/A, the requested URI is an XML namespace.

This document request the IANA to register four 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: tsm  
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]

## 10. Acknowledgements

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

## 11. References

### 11.1. Normative References

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.
- [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>>.
- [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>>.
- [I-D.ietf-ccamp-l1csm-yang]  
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang-10 (work in progress), September 2019.
- [I-D.ietf-teas-actn-vn-yang]  
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-06 (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-21 (work in progress), April 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-10 (work in progress), July 2019.

## 11.2. Informative References

- [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>>.
- [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>>.
- [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>>.
- [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-actn-yang]  
Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O., Shin, J., and S. Belotti, "Applicability of YANG models for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-yang-04 (work in progress), August 2019.

## Appendix A. Contributor Addresses

Adrian Farrel  
Old Dog Consulting

EMail: [adrian@olddog.co.uk](mailto:adrian@olddog.co.uk)

Italo Busi  
Huawei Technologies

EMail: [Italo.Busi@huawei.com](mailto:Italo.Busi@huawei.com)

Haomian Zheng  
Huawei Technologies

EMail: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

## Authors' Addresses

Young Lee (editor)  
SKKU

Email: [younglee.tx@gmail.com](mailto:younglee.tx@gmail.com)

Dhruv Dhody (editor)  
Huawei Technologies

Email: [dhruv.ietf@gmail.com](mailto:dhruv.ietf@gmail.com)

Giuseppe Fioccola  
Huawei Technologies

Email: [giuseppe.fioccola@huawei.com](mailto:giuseppe.fioccola@huawei.com)

Qin Wu (editor)  
Huawei Technologies

Email: [bill.wu@huawei.com](mailto:bill.wu@huawei.com)

Daniele Ceccarelli  
Ericsson  
Torshamnsgatan, 48  
Stockholm, Sweden

Email: [daniele.ceccarelli@ericsson.com](mailto:daniele.ceccarelli@ericsson.com)

Jeff Tantsura  
Apstra

Email: [jefftant@gmail.com](mailto:jefftant@gmail.com)



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 9, 2020

X. Liu  
Volta Networks  
I. Bryskin  
Futurewei  
V. Beeram  
T. Saad  
Juniper Networks  
H. Shah  
Ciena  
O. Gonzalez de Dios  
Telefonica  
July 8, 2019

YANG Data Model for Layer 3 TE Topologies  
draft-ietf-teas-yang-l3-te-topo-05

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 January 9, 2020.

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
1.1. Terminology . . . . .	3
1.2. Tree Diagrams . . . . .	3
2. Modeling Considerations for L3 TE Topologies . . . . .	3
2.1. Relationship Between Layer 3 Topology and TE topology . .	3
2.2. Relationship Modeling . . . . .	4
2.2.1. Topology Referencing . . . . .	4
2.2.2. Node Referencing . . . . .	4
2.2.3. Link Termination Point Referencing . . . . .	4
2.2.4. Link Referencing . . . . .	5
2.3. Topology Type Modeling . . . . .	5
3. Packet Switching Technology Extensions . . . . .	5
3.1. Technology Specific Link Attributes . . . . .	5
3.2. Performance Metric . . . . .	6
4. Complete Model Tree Structure . . . . .	7
4.1. Layer 3 TE Topology Module . . . . .	7
4.2. Packet Switching TE Topology Module . . . . .	7
5. YANG Modules . . . . .	27
5.1. Layer 3 TE Topology Module . . . . .	27
5.2. Packet Switching TE Topology Module . . . . .	32
6. IANA Considerations . . . . .	39
7. Security Considerations . . . . .	41
8. References . . . . .	43
8.1. Normative References . . . . .	43
8.2. Informative References . . . . .	45
Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations . . . . .	46
A.1. Layer 3 TE Topology State Module . . . . .	46
A.2. Packet Switching TE Topology State Module . . . . .	49
Appendix B. Data Tree Example . . . . .	55
Authors' Addresses . . . . .	65

## 1. Introduction

This document defines a YANG [RFC7950] data model for describing the relationship between a layer 3 network topology [RFC8346] and a Traffic Engineering (TE) topology [I-D.ietf-teas-yang-te-topo].

When traffic engineering is enabled on a layer 3 network topology, there will be a corresponding TE topology. The TE topology may or

may not be congruent to the layer 3 network 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 topology to 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 allows 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

The YANG modules `ietf-l3-te-topology` and `ietf-l3-te-topology-state` model configuration and operational state of layer 3 TE network topologies. These two modules augment `ietf-l3-te-topology` and `ietf-l3-te-topology-state` respectively, so a layer 3 TE network topology is a layer 3 network topology with additional TE capability enabled.

### 2.1. Relationship Between Layer 3 Topology and TE topology

In general, layer 3 network topology model and TE topology model can be used independently. When traffic engineering is enabled on a layer 3 network topology, there will be associations between objects in layer 3 network topologies and objects in TE topologies. The properties of these relations 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 network may have zero or one associated object in the corresponding TE network.

## 2.2. Relationship Modeling

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

### 2.2.1. Topology Referencing

When TE is enabled on a layer 3 network topology, if the TE topology is not congruent to the layer 3 network topology, the layer 3 network 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
```

If the TE topology is congruent to the layer 3 network 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 network topology, if the TE topology is not congruent to the layer 3 network topology, a layer 3 network node may have a reference to the corresponding TE node. 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 network topology, if the TE topology is not congruent to the layer 3 network topology, a layer 3 link termination point may have a reference to the corresponding TE link termination point. 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 network topology, if the TE topology is not congruent to the layer 3 network topology, a layer 3 link may have a reference to the corresponding TE link. 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 topology with TE enabled.

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

### 3. Packet Switching Technology Extensions

The technology agnostic TE Topology model is defined in [I-D.ietf-teas-yang-te-topo], which is extended by this document to cover the Packet Switch Capable (PSC) technology [RFC3471] [RFC7074].

#### 3.1. 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

```

### 3.2. Performance Metric

[RFC7471], [RFC8570] and [RFC7823] specify TE performance metric parameters and their usage. The packet switching augmentations specified in this moducment 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
      | .....

```

Such an augmentation has been applied to:

- o Connectivity matrices container
- o Connectivity matrix entry
- o Local ink connectivities container
- o Local ink 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

#### 4. Complete Model Tree Structure

##### 4.1. 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

```

##### 4.2. Packet Switching TE Topology Module

This is an augmentation to base TE topology model.

```

module: ietf-te-topology-packet
  augment /nw:networks/nw:network/nw:node/tet:te
    /tet:te-node-attributes/tet:connectivity-matrices:
      +--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
+--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 /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
    /tet:connectivity-matrix:
      +--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
      +--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 /nw:networks/nw:network/nw:node/tet:te
    /tet:information-source-entry/tet:connectivity-matrices:
+---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
+---ro throttle
    +---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
|   +--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
+--ro throttle
+--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:tunnel-termination-point
    /tet:local-link-connectivities:
+---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
+--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 /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
    +---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
+---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 /nw:networks/tet:te/tet:templates/tet:link-template
    /tet:te-link-attributes:
+---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
+--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 /nw:networks/nw:network/nt:link/tet:te
    /tet:te-link-attributes:
    +---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
+--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 /nw:networks/nw:network/nt:link/tet:te

```



```

        /tet:information-source-entry:
+--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
+--ro throttle
|   +--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/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

```

## 5. YANG Modules

### 5.1. Layer 3 TE Topology Module

This module references [RFC8345], [RFC8346], and [I-D.ietf-teas-yang-te-topo].

```

<CODE BEGINS> file "ietf-l3-te-topology@2019-06-28.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 2019-06-28 {  
    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>
```

## 5.2. Packet Switching TE Topology Module

This module references [RFC7471], [RFC7823], [RFC8294], [RFC8345], [RFC8346], [RFC8570], [I-D.ietf-teas-yang-te-types], and [I-D.ietf-teas-yang-te-topo].

```
<CODE BEGINS> file "ietf-te-topology-packet@2019-06-28.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 2019-06-28 {
  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 connectivity-matrix */
  augment "/nw:networks/nw:network/nw:node/tet:te/"
    + "tet:te-node-attributes/tet:connectivity-matrices" {
    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" {
    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" {
```

```
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" {
  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" {
  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" {
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-psc1']" {
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 "tet:interface-switching-capability "
        + "[tet:switching-capability = 'te-types:switching-psc1']" {
```

```
        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 "tet:interface-switching-capability "
        + "[tet:switching-capability = 'te-types:switching-psc1']" {
        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-psc1' " {
        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 "tet:switching-capability = 'te-types:switching-psc1' " {
      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 "tet:switching-capability = 'te-types:switching-psc1' " {
      description
        "Valid only for PSC";
    }
    description
      "Parameters for PSC TE topology.";
    uses packet-switch-capable-container;
  }
}
<CODE ENDS>

```

## 6. 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: l3te  
reference: RFC XXXX  
-----

-----  
name: ietf-l3-te-topology-state  
namespace: urn:ietf:params:xml:ns:yang:ietf-l3-te-topology-state  
prefix: l3te-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  
-----



## 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 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/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 network 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 layer 3 node, 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 layer 3 link termination point, 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 layer 3 link, 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 behaviours 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 network 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.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [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>>.
- [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-09 (work in progress), May 2019.
- [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.

## 8.2. 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>>.

## 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@2019-06-28.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 2019-06-28 {

## 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@2019-06-28.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 2019-06-28 {
  description
    "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Layer 3 TE Topologies";
}

/*
 * Augmentations
 */
/* 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" {
  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" {
  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" {
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" {
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" {
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" {
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-psc1']" {
    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 "tet-s:interface-switching-capability "
    + "[tet-s:switching-capability = 'te-types:switching-psc1']" {
    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 "tet-s:interface-switching-capability "
```

```

    + "[tet-s:switching-capability = 'te-types:switching-psc1']" {
      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-psc1' " {
    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 "tet-s:switching-capability = 'te-types:switching-psc1' " {
    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 "tet-s:switching-capability = 'te-types:switching-psc1' " {
    description
      "Valid only for PSC";
  }
  description
    "Parameters for PSC TE topology.";
  uses tet-pkt:packet-switch-capable-container;
}

```

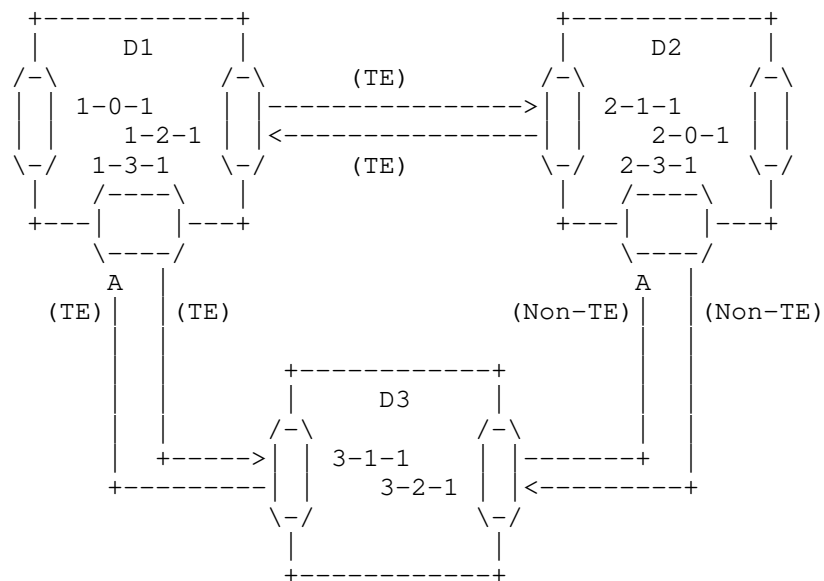
```

}
<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-psc1",
                                "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-psc1",
                        "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-psc1",
          "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-psc1",
            "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-psc1",
                    "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-psc1",
            "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  
Futurewei

EMail: igor.bryskin@futurewei.com

Vishnu Pavan Beeram  
Juniper Networks

EMail: vbeeram@juniper.net

Tarek Saad  
Juniper Networks

EMail: tsaad@juniper.net

Himanshu Shah  
Ciena

EMail: hshah@ciena.com

Oscar Gonzalez de Dios  
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

TEAS Working Group  
Internet Draft  
Intended status: Standard Track  
Expires: January 2020

Italo Busi (Ed.)  
Huawei  
Sergio Belotti (Ed.)  
Nokia

July 8, 2019

Yang model for requesting Path Computation  
draft-ietf-teas-yang-path-computation-06.txt

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), 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 January 8, 2020.

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

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths.

This document defines a YANG data model for a stateless RPC to request path computation. This model complements the stateful solution defined in [TE-TUNNEL].

Moreover this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

## Table of Contents

1. Introduction.....	3
1.1. Terminology.....	4
2. Use Cases.....	5
2.1. Packet/Optical Integration.....	5
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.....	19
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 TE topology and path computation.....	24
3.3. Stateless and Stateful Path Computation.....	27
3.3.1. Temporary reporting of the computed path state.....	29
4. Path Computation and Optimization for multiple paths.....	31

5. YANG Model for requesting Path Computation.....	32
5.1. Synchronization of multiple path computation requests....	32
5.2. Returned metric values.....	34
6. YANG model for stateless TE path computation.....	36
6.1. YANG Tree.....	36
6.2. YANG Module.....	46
7. Security Considerations.....	61
8. IANA Considerations.....	62
9. References.....	62
9.1. Normative References.....	62
9.1. Informative References.....	64
10. Acknowledgments.....	64
Appendix A. Examples of dimensioning the "detailed connectivity matrix"	66

## 1. Introduction

There are scenarios, typically in a hierarchical SDN context, where the topology information provided by a TE network provider may not be sufficient for its client to perform end-to-end path computation. In these cases the client would need to request the provider to calculate some (partial) feasible paths, complementing his topology knowledge, to make his end-to-end path computation feasible.

This type of scenarios can be applied to different interfaces in different reference architectures:

- o ABNO control interface [RFC7491], in which an Application Service Coordinator can request ABNO controller to take in charge path calculation (see Figure 1 in [RFC7491]).
- o ACTN [RFC8453], where a controller hierarchy is defined, the need for path computation arises on both interfaces CMI (interface between Customer Network Controller (CNC) and Multi Domain Service Coordinator (MDSC)) and/or MPI (interface between MSDC-PNC). [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 or RESTCONF).

Path Computation Elements, 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 [TE-TOPO]. Furthermore, the technology specific details of the TED are modeled in the augmented TE topology models (e.g. [OTN-TOPO] for OTN ODU technologies).

The availability of such topology models allows providing 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.

Note: This document assumes that the client of the YANG data model defined in this document may not implement a "PCE" functionality, as defined in [RFC4655].

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 model [TE-TUNNEL].

This document proposes a YANG model for a path computation request defined as a stateless RPC, which complements the stateful solution defined in [TE-TUNNEL].

Moreover, this document describes some use cases where a path computation request, via YANG-based protocols (e.g., NETCONF or RESTCONF), can be needed.

### 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 LSP by operating on the TED and considering bandwidth and other constraints applicable to the TE LSP service request. [RFC4655]

## 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 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 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 network is used to provide connectivity to some nodes of a Packet network (see Figure 1).

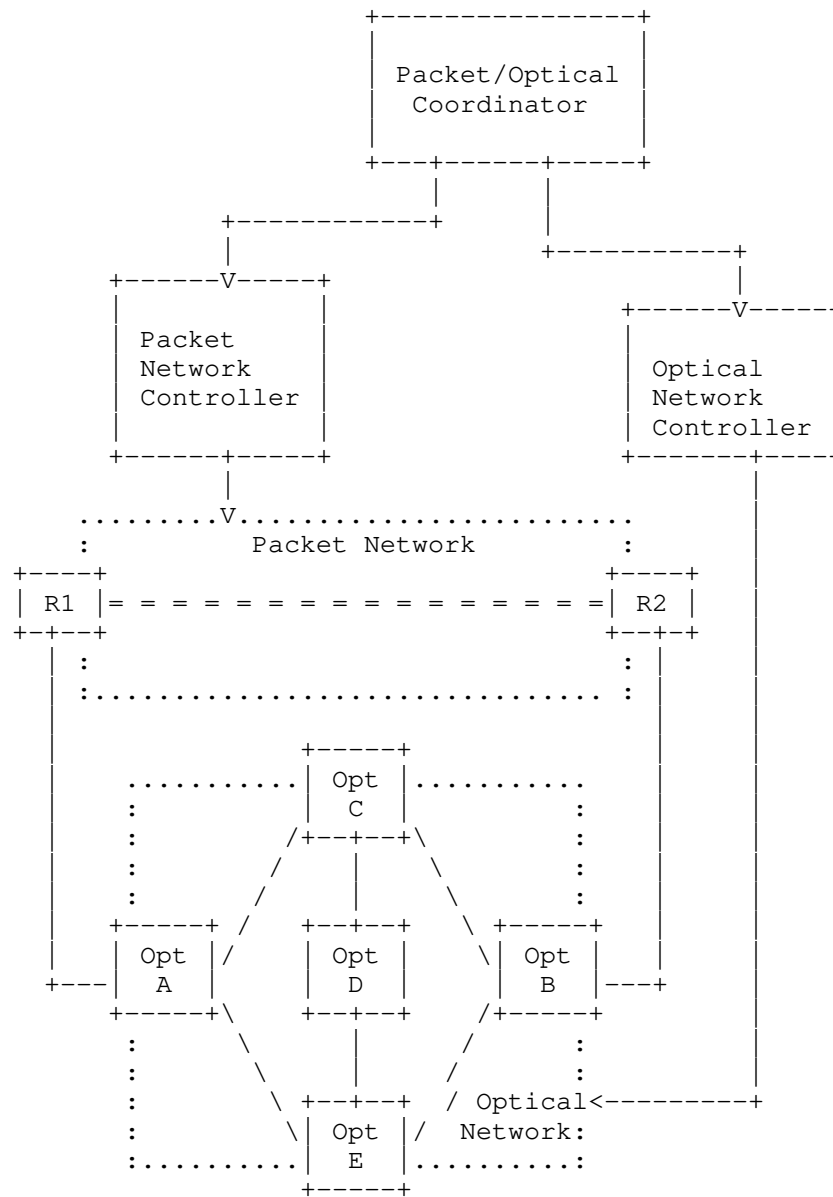


Figure 1 - Packet/Optical Integration Use Case



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 network 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 network controller can provide the packet/optical coordinator the information it needs to setup connectivity between packet nodes through the Optical network (e.g., the access links).

The path computation request helps the coordinator to know the real connections that can be provided by the optical network.



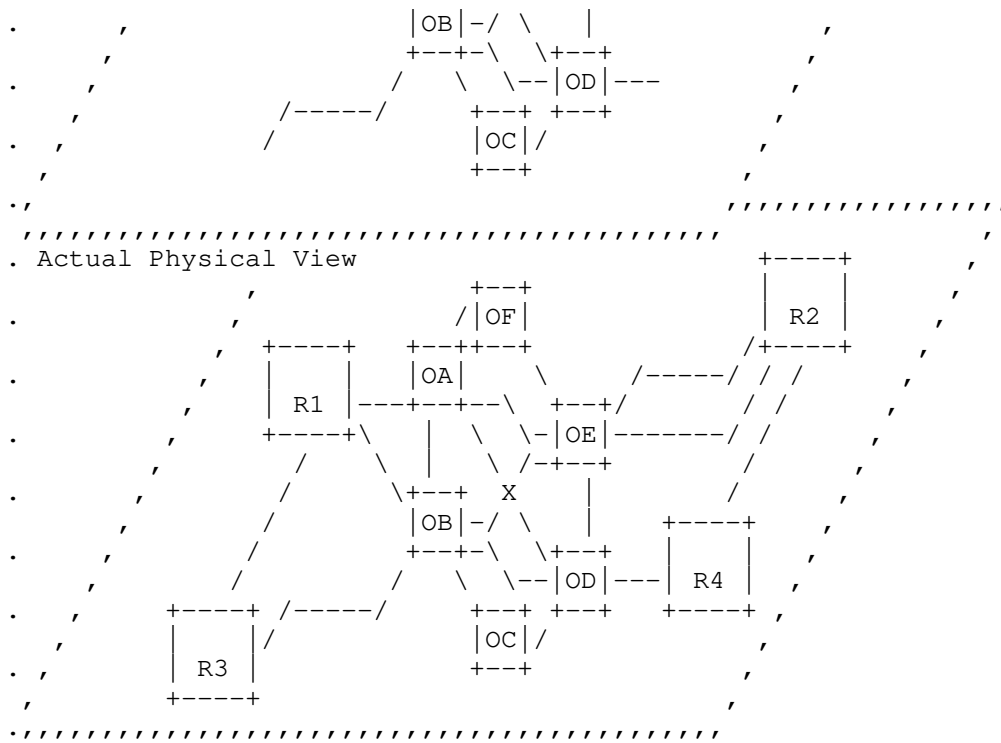


Figure 2 - Packet and Optical Topology Abstractions

In this use case, the coordinator needs to setup an optimal underlying path for an IP link between R1 and R2.

As depicted in Figure 2, the 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 from the current status of the physical resources within the optical network and on vendor-specific optical attributes.

The 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 setup the optimal end-to-end path crossing optical network.

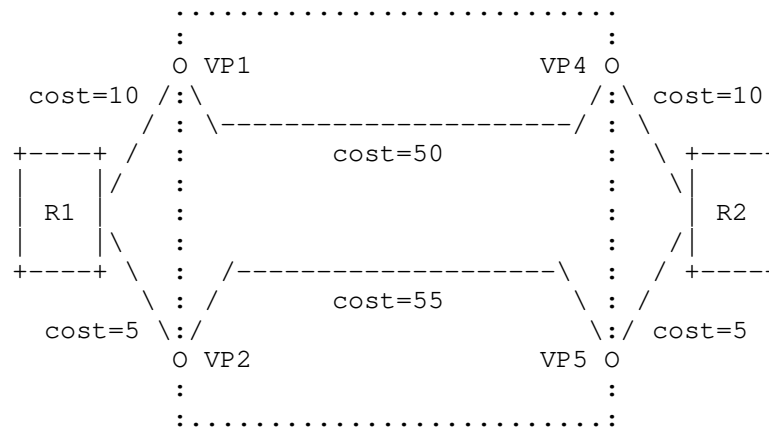


Figure 3 - Packet/Optical Path Computation Example

For example, in Figure 3, the Coordinator can request the Optical network controller to compute the paths between VP1-VP4 and VP2-VP5 and then decide to setup the optimal end-to-end path using the VP2-VP5 Optical path even 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 network controller when requested by the Coordinator is no longer valid/available when the Coordinator requests it to be setup 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.

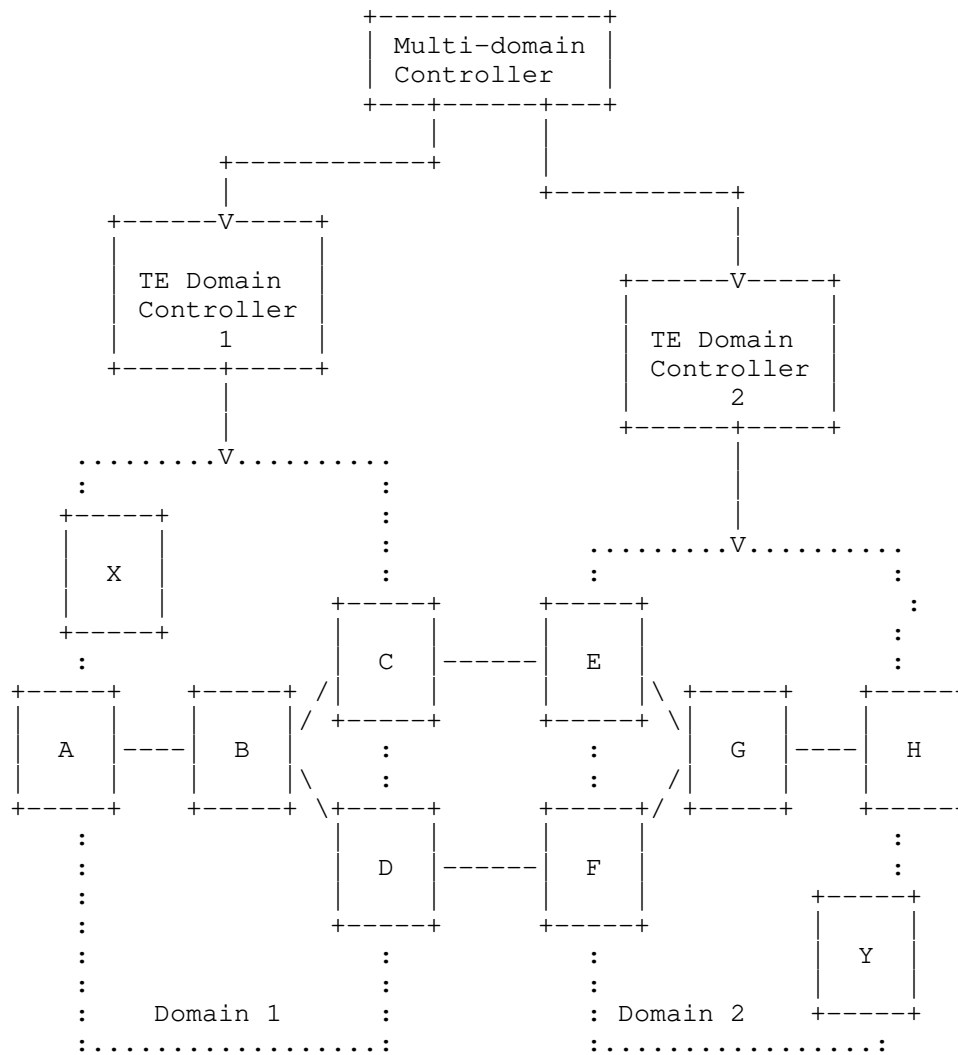


Figure 4 - Multi-domain multi-link interconnection

In order to setup 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 from the current status of the physical resources within each TE network. 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 setup 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:

- o The multi-domain controller asks TE domain controllers to provide set of paths between A-C, A-D, E-H and F-H
- o 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)
- o 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 setup the A-D and F-H intra-domain paths
- o 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 to 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.

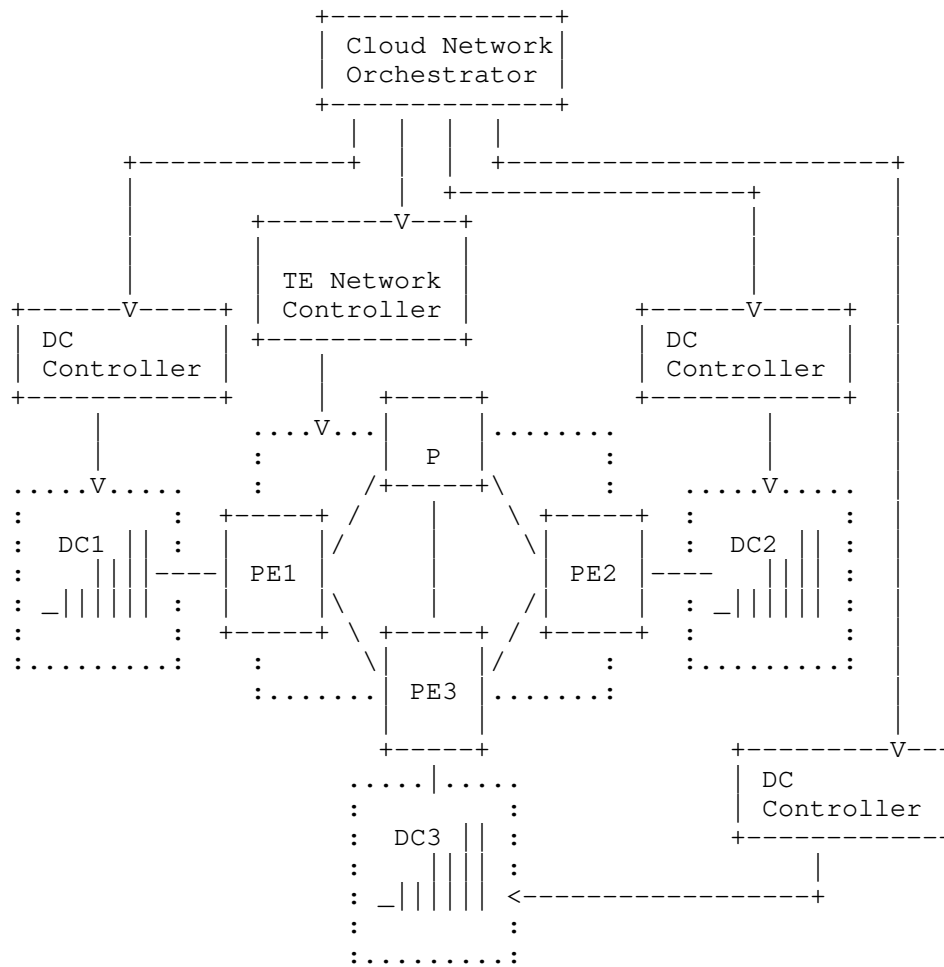


Figure 5 - Data Center Interconnection Use Case

In this use case, there is 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 centers

resources. It may not be able to make this decision because it has only an abstract view of the TE network (as in use case in 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.



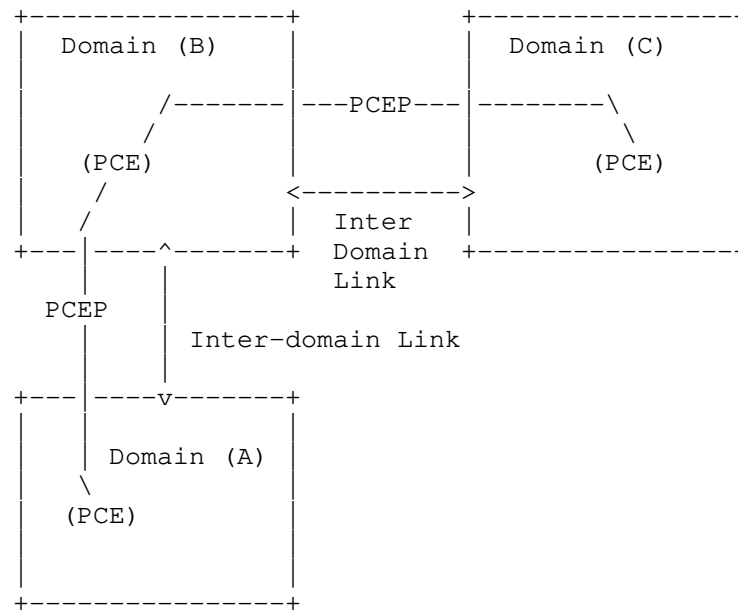


Figure 6 - BRPC Scenario

In this use case, a client can use the YANG model defined in this document to request path computation to 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 involve 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.

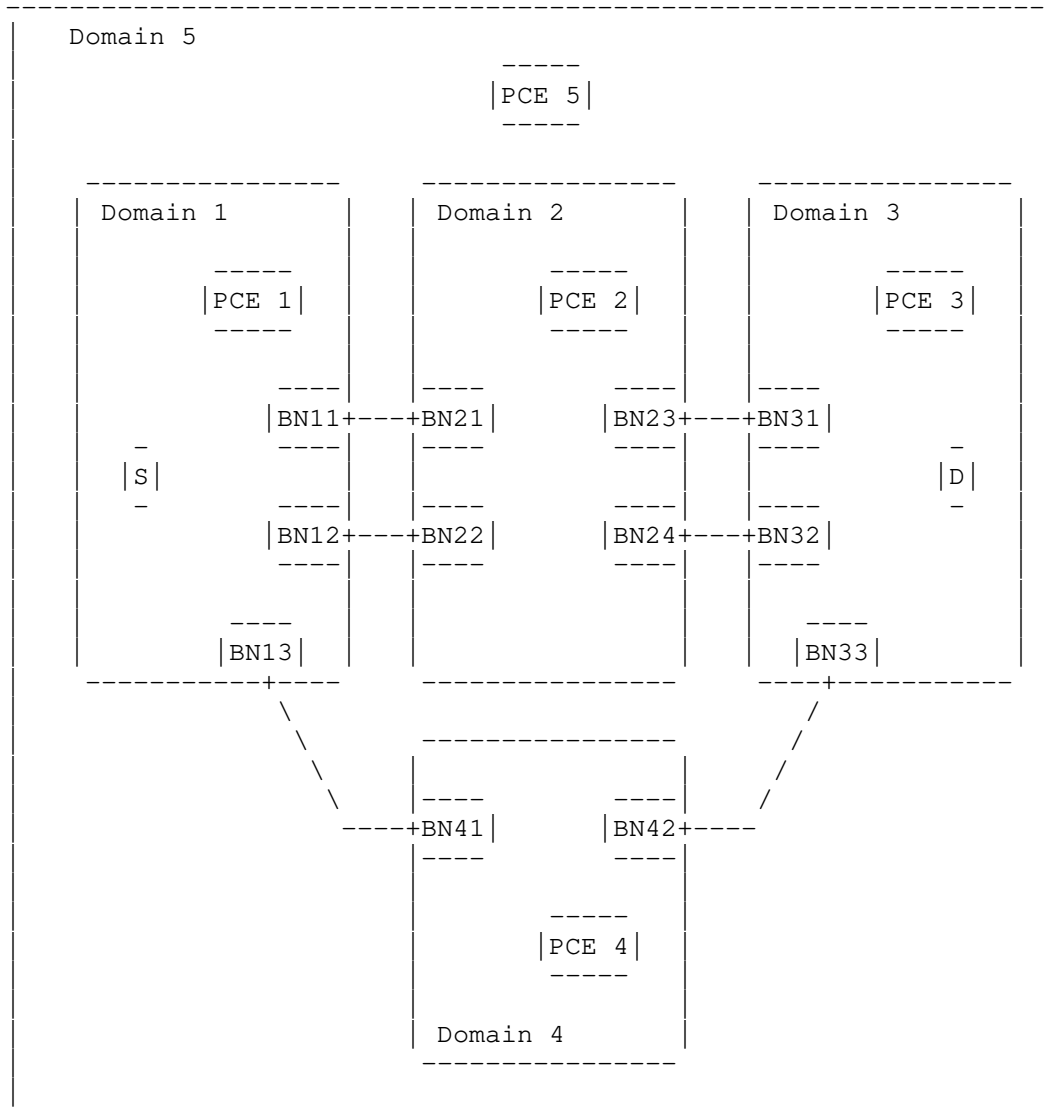


Figure 7 - Hierarchical domain topology from [RFC6805]

In this use case, a client can use the YANG 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 model defined in this document. For example the YANG 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 model defined in this document.

Section 3.1 describes the motivation for a YANG model to request path computation.

Section 3.2 describes the motivation for a YANG model which complements the TE Topology YANG model defined in [TE-TOPO].

Section 3.3 describes the motivation for a stateless YANG RPC which complements the TE Tunnel YANG model defined in [TE-TUNNEL].

#### 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., [TE-TOPO] as well as that are used to configure and manage TE Tunnels, i.e., [TE-TUNNEL].

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:

- o There is no need for an error-prone mapping or correlation of information.
- o It is possible to use the same endpoint identifiers in path computation requests and in the topology modeling.

- o 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 (setup/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 SDN controller to the orchestrator is aggregated using the "virtual node model", defined in [RFC7926].

TE Topology information, e.g., as provided by [TE-TOPO], could in theory be used by an underlying SDN controllers to provide TE information to its client thus allowing a PCE available within its client to perform multi-domain path computation by its own, without requesting path computations to the underlying SDN 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 to the underlying controllers to get the information it needs to compute the optimal end-to-end path.

This section analyzes the need for a client to request underlying SDN controllers for path computation even in case it implements a

PCE functionality, as well as how the TE Topology information and the path computation can be complementary.

In nutshell, there is a scalability trade-off between providing all the TE information needed by PCE, when implemented by the client, to take optimal path computation decisions by its own versus sending too many requests to underlying SDN Domain Controllers to compute a set of feasible optimal intra-domain TE paths.

### 3.2.1. TE Topology Aggregation

Using the TE Topology model, as defined in [TE-TOPO], the underlying SDN controller can export the whole TE domain as a single abstract TE node with a "detailed connectivity matrix".

The concept of a "detailed connectivity matrix" is defined in [TE-TOPO] to provide specific TE attributes (e.g., delay, SRLGs and summary TE metrics) as an extension of the "basic connectivity matrix", which is based on the "connectivity matrix" defined in [RFC7446].

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 network controller can make the information shown in Figure 3 available to the Coordinator as part of the TE Topology information and the Coordinator could use this information to calculate by its own the optimal path between R1 and R2, without requesting any additional information to the Optical network 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 to Orchestrator) 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).



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.

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 src to dest, 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 reachability 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.

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.

Instead, if the information in the "detailed connectivity matrix" is not complete/accurate, we can have the following drawbacks considering for example the case in Figure 8:



- o 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;
- o 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.

Using the approach proposed in this document, the client, when it needs to setup 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:

- o 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 Orchestrator can successfully setup the end-to-end path passing through this Optical path;
- o 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 Orchestrator 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 [TE-TOPO], the underlying SDN controller can export an abstract TE Topology, composed by a set of TE nodes and TE links, representing the abstract view of the topology controlled by each domain controller.

Considering the example in Figure 4, the TE domain controller 1 can export a TE Topology encompassing the TE nodes A, B, C and D and the TE Link interconnecting them. In a similar way, TE domain controller 2 can export a TE Topology encompassing the TE nodes E, F, G and H and the TE Link 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 setup 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:

- o Multi-domain controller's PCE, based on its own information, can compute the optimal multi-domain path being A-B-C-E-G-H, and then request the TE domain controllers to setup the A-B-C and E-G-H intra-domain paths
- o But, during path setup, the 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
- o So what the multi-domain controller computed is not good and need 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 setup. Similarly, it is possible to first compute an abstract end-to-end path within the multi-domain Orchestrator (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 that performing per-domain path computation and then stitch them.

### 3.2.3. Complementary use of TE topology and path computation

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 TE topology information provided by the domain controllers to limit the number of requests.



- o 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-E-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-E-F but it cannot know which one of the two possible options 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, E and F (and not to all the possible TE domain controllers).

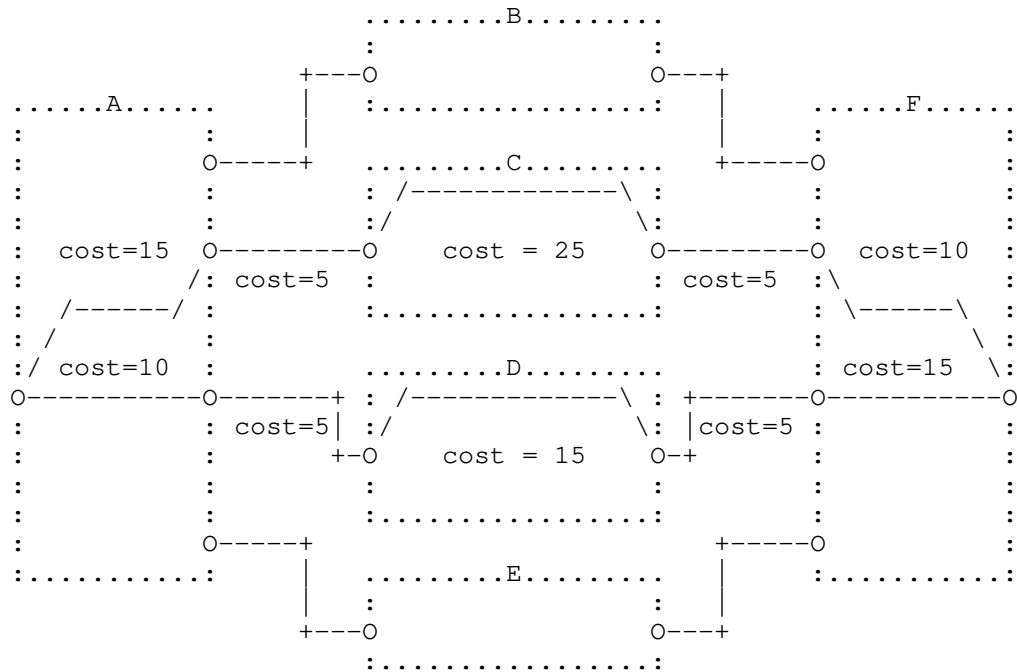


Figure 10 - Multi-domain with many domains (Path Computation information)

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. Stateless and Stateful Path Computation

The TE Tunnel YANG model, defined in [TE-TUNNEL], can support the need to request path computation.

It is possible to request path computation by configuring a "compute-only" TE tunnel and retrieving the computed path(s) in the LSP(s) Record-Route Object (RRO) list as described in section 3.3.1 of [TE-TUNNEL].

This is a stateful solution since the state of each created "compute-only" TE tunnel needs to be maintained and updated, when underlying network conditions change.

It is very useful to provide options for both stateless and stateful path computation mechanisms. It is suggested to use stateless mechanisms as much as possible and to rely on stateful path computation when really needed.

Stateless RPC allows requesting path computation using a simple atomic operation and it is the natural option/choice, especially with stateless PCE. The stateless path computation solution assumes that the underlying SDN controller (e.g., a PNC) will compute a path twice during the process to setup 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 setup 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 SDN controller (e.g. PNC) and therefore the path that is setup at time T2 is exactly the same path that has been computed at time T1.

Since the operation is stateless, there is no guarantee that the returned path would still be available when path setup is requested: this does not cause major issues in case the time between path computation and path setup 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 setup 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 necessary to verify that the actual path that has been setup 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 setup path is not exactly the same as the path returned by path computation, its metrics and constraints are "good enough" (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 setup and path verification).

In case the stateless solution is not sufficient and it would be the need to setup at T2 exactly the same path computed at T1 a stateful solution, based on "compute-only" TE tunnel, could be used to get 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 config DS 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 setup of the LSP.

It is worth noting that also the stateful solution, although increasing the likelihood that the computed path is available at path setup, does not guaranteed that because notifications may not be reliable or delivered on time. Path verification is needed also when stateful path computation is used.

The stateful path computation has also the following drawbacks:

- o Several messages required for any path computation
- o Requires persistent storage in the provider controller
- o Need for garbage collection for stranded paths

- o 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 where the underlying SDN controller, after having received a path computation RPC request, maintains some "temporary state" associated with the computed path, allowing the client to request the setup of exactly that path, if still available.

This is similar to the stateful solution but, to avoid the drawbacks of the stateful approach, is leveraging the path computation RPC and the separation between configuration and operational DS, as defined in the NMDA architecture [RFC8342].

The underlying SDN controller, after having computed a path, as requested by a path computation RPC, also creates a te-tunnel instance within the operational DS, to store that computed path. This would be similar to the stateful solution with the only difference that there is no associated te-tunnel instance within the running DS.

Since underlying SDN controller stores in the operational DS 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 to setup that specific path by creating a te-tunnel instance (not in compute-only mode) in the running DS using the same tunnel-name of the existing te-tunnel in the operational datastore: this will trigger the underlying SDN controller to setup that path, if still available.

There are still cases where the path being setup is not exactly the same as the path that has been computed:

- o When the tunnel is configured with path constraints which are not compatible with the computed path
- o When the tunnel setup is requested after the resources of the computed path are no longer available

- o When the tunnel setup is requested after the computed path is no longer known (e.g. due to a server reboot) by the underlying SDN controller

In all these cases, the underlying SDN controller should compute and setup a new path.

Therefore the "path verification" phase, as described in section 3.3 above, is still needed to check that the path that has been setup 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 DS. This operation is fully controlled by the underlying SDN 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 requests 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 SDN controller can remove from the operational DS all the paths computed with a given transaction-id which have not been setup either when it receives a Path Delete RPC request for that transaction-id or, automatically, right after the setup up of a path that have been previously computed with that transaction-id.

This possibility is useful when multiple paths are computed but, at most, only one is setup (e.g., in multi-domain path computation scenario scenarios). After the selected path has been setup (e.g, in one domain during multi-domain path setup), all the other alternative computed paths can be automatically deleted by the underlying SDN controller (since no longer needed). The client can also request, using the Path Delete RPC request, the underlying SDN controller to remove all the computed paths, if none of them is going to be setup (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 DS when no path is setup and no explicit delete RPC 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 setup 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 a not-synchronized 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 Model for requesting Path Computation

This document define a YANG stateless RPC to request path computation as an "augmentation" of tunnel-rpc, defined in [TE-TUNNEL]. 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-rpc/te:input/te:tunnel-info:
  +---- path-request* [request-id]
  .....
```

```
augment /te:tunnels-rpc/te:output/te:result:
  +--ro response* [response-id]
  +--ro response-id      uint32
  +--ro (response-type)?
    +--:(no-path-case)
    |   +--ro no-path!
    +--:(path-case)
      +--ro computed-path
      .....
```

This model extensively re-uses the grouping defined in [TE-TUNNEL] to ensure maximal syntax and semantics commonality.

### 5.1. Synchronization of multiple path computation requests

The YANG model permits to synchronize a set of multiple path requests (identified by specific request-id) all related to a "svec" container emulating the syntax of "SVEC" PCEP object [RFC5440].

```
+---- synchronization* [synchronization-id]
  +---- synchronization-id      uint32
  +---- svec
    +---- relaxable?             boolean
    +---- disjointness?         te-types:te-path-disjointness
    +---- request-id-number*    uint32
  +---- svec-constraints
    +---- path-metric-bound* [metric-type]
      +---- metric-type         identityref
      +---- upper-bound?       uint64
```

```

+---- path-srlgs-values
|   +---- usage?    identityref
|   +---- values*   srlg
+---- path-srlgs-names
|   +---- path-srlgs-name* [usage]
|       +---- usage      identityref
|       +---- srlg-name* [name]
|           +---- name    string
+---- exclude-objects
.....
+---- optimizations
|   +---- (algorithm)?
|       +---:(metric)
|           +---- optimization-metric* [metric-type]
|               +---- metric-type      identityref
|               +---- weight?          uint8
|       +---:(objective-function)
|           +---- objective-function
|               +---- objective-function-type?  identityref

```

The model, in addition to the metric types, defined in [TE-TUNNEL], which can be applied to each individual path request, defines additional specific metrics types that apply to a set of synchronized requests, as referenced in [RFC5541].

```

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

```

```

identity svec-metric-cumul-te {
  base svec-metric-type;
  description
    "TE cumulative path metric";
}

```

```

identity svec-metric-cumul-igp {
  base svec-metric-type;
  description
    "IGP cumulative path metric";
}

```

```

    }

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

    identity svec-metric-aggregate-bandwidth-consumption {
        base svec-metric-type;
        description
            "Cumulative bandwidth consumption of the set of
            synchronized paths";
    }

    identity svec-metric-load-of-the-most-loaded-link {
        base svec-metric-type;
        description
            "Load of the most loaded link";
    }

```

## 5.2. Returned metric values

This YANG 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]:

```

augment /te:tunnels-rpc/te:output/te:result:
  +--ro response* [response-id]
    +--ro response-id      uint32
    +--ro (response-type)?
      +--:(no-path-case)
      |   +--ro no-path!
      +--:(path-case)
        +--ro computed-path
          +--ro path-id?          yang-types:uuid
          +--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-values
|   +--ro usage?              identityref
|   +--ro values*            srlg
+--ro path-srlgs-names
|   +--ro path-srlgs-name* [usage]
|       +--ro usage          identityref
|       +--ro srlg-name* [name]
|           +--ro name        string
+--ro path-route-objects
.....

```

It also allows to request in the input of RPC which information (metrics, srlg and/or affinities) should be returned:

```

module: ietf-te-path-computation
augment /te:tunnels-rpc/te:input/te:tunnel-info:
+---- path-request* [request-id]
|   +---- request-id          uint32
|   .....
|   +---- requested-metrics* [metric-type]
|       | +---- metric-type    identityref
|       +---- return-srlgs?    boolean
|       +---- return-affinities? boolean
|       .....

```

This feature is essential for using a stateless 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 TE network 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 TE network controllers and choose the best one for the optimal e2e path.

## 6. YANG model for stateless TE path computation

### 6.1. YANG Tree

Figure 11 below shows the tree diagram of the YANG model defined in module `ietf-te-path-computation.yang`.

```

module: ietf-te-path-computation
  augment /te:tunnels-rpc/te:input/te:tunnel-info:
    +----- path-request* [request-id]
      |
      | +----- request-id                               uint32
      | +----- encoding?                               identityref
      | +----- switching-type?                         identityref
      | +----- source?                                inet:ip-address
      | +----- destination?                           inet:ip-address
      | +----- src-tp-id?                              binary
      | +----- dst-tp-id?                              binary
      | +----- bidirectional?                          boolean
      | +----- te-topology-identifier
      | | +----- provider-id?   te-global-id
      | | +----- client-id?    te-global-id
      | | +----- topology-id?  te-topology-id
      | +----- 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-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
+----- 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)?

```

```

types:generalized-label
+---:(generic)
+----- generic?
+----- rt-
+----- 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-
+---:(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-
+---:(as-number)
+----- as-number-hop
+----- as-number    inet:as-number
+----- hop-type?    te-hop-type
+---:(label)
+----- label-hop
+----- te-label
+----- (technology)?
+----- +---:(generic)

```

```

types:generalized-label
    +---- generic?
    rt-
    +---- direction?
    te-label-direction
    +---- tiebreakers
    +---- tiebreaker* [tiebreaker-type]
    +---- tiebreaker-type identityref
    +--:(objective-function)
    {path-optimization-objective-function}?
    +---- objective-function
    +---- objective-function-type? identityref
+---- 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-name?          string
|   +---- (path)?
|       +---:(primary)
|       |   +---- primary-path-name?      string
|       +---:(secondary)
|       |   +---- secondary-path-name?     string
+---- synchronization* [synchronization-id]
|   +---- synchronization-id      uint32
+---- 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* [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
+---- optimizations                 +---- direction?   te-label-direction
|   +---- (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-rpc/te:output/te:result:
+--ro response* [response-id]
+--ro response-id      uint32
+--ro (response-type)?
+--:(no-path-case)
| +--ro no-path!
+--:(path-case)
+--ro computed-path
+--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 tunnel-ref?                         te:tunnel-ref
+--ro (path)?
  +--:(primary)
  | +--ro primary-path-ref?      leafref
  +--:(secondary)
    +--ro secondary-path-ref?  leafref
augment /te:tunnels-rpc/te:input/te:tunnel-info:
  +---- deleted-paths-transaction-id*  string
augment /te:tunnels-rpc/te:output/te:result:
  +---- deleted-paths-transaction-id*  string

```

Figure 11 - TE path computation YANG tree

## 6.2. YANG Module

```

<CODE BEGINS>file "ietf-te-path-computation@2019-03-11.yang"
module iETF-te-path-computation {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-te-path-computation";
  // replace with IANA namespace when assigned

  prefix "tepc";

  import iETF-inet-types {
    prefix "inet";
  }

  import iETF-te {
    prefix "te";
  }

  import iETF-te-types {
    prefix "te-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>

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

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

";

description "YANG model for stateless TE path computation";

revision "2019-03-11" {
  description
    "Initial revision";
  reference
    "draft-ietf-teas-yang-path-computation";
}

/*
 * Features
 */

feature stateless-path-computation {
  description
    "This feature indicates that the system supports
    stateless path computation.";
}

/*
 * Groupings
 */

grouping path-info {
  uses te-types:generic-path-properties;
  description "Path computation output information";
```

```
}

grouping requested-info {
  description
    "This grouping defines the information (e.g., metrics)
    which must be returned in the 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-state {
  description
```

```
    "Configuration for the transient state used
    to report the computed path";
  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).
      ";
  }
}
leaf tunnel-name {
  type string;
  description
    "
    The suggested name to be assigned to the te-tunnel
    instance which is created to report the computed path.
```

In case multiple paths are requested with the same suggested name, the server will create only one te-tunnel instance to report all the computed paths with the same suggested name.

A different name can be assigned by server (e.g., when a te-tunnel with this name already exists).

```
    ";
  }
  choice path {
    description
      "The transient state of the computed path can be reported
      as a primary or a secondary path of a te-tunnel";
    case primary {
      leaf primary-path-name {
        type string;
        description
          "
          The suggested name to be assigned to the
          p2p-primary-path instance which is created
          to report the computed path.

          A different name can be assigned by the server
          (e.g., when a p2p-primary-path with this name
          already exists).
          ";
      }
    }
  }
  case secondary {
    leaf secondary-path-name {
      type string;
      description
        "
        The suggested name to be assigned to the
        p2p-secondary-path instance which is created
        to report the computed path.

        A different name can be assigned by the server
        (e.g., when a p2p-secondary-path with this
```

```
        name already exists).

        If not specified, the a p2p-primary-path is created
        by the server.
        ";
    }
}
}

grouping reported-state {
  description
    "Information about the transient state created
    to report 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 empty.
      ";
  }
  choice path {
    description
      "The transient state of the computed path can be reported
      as a primary or a secondary 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:p2p-primary-paths/te:p2p-primary-path/" +
            "te:name";
        }
        must "../tunnel-ref" {
          description
```

```
        "The primary-path-name can only be reported
        if also the tunnel is reported
        to provide the complete reference.";
    }
    description
    "
        Reference to the p2p-primary-path that reports
        the transient state of the computed path.

        If no transient state is created,
        this attribute is empty.
    ";
}
}
case secondary {
    leaf secondary-path-ref {
        type leafref {
            path "/te:te/te:tunnels/" +
                "te:tunnel[te:name=current()/../tunnel-ref]/" +
                "te:p2p-secondary-paths/te:p2p-secondary-path/" +
                "te:name";
        }
        must "../tunnel-ref" {
            description
            "The secondary-path-name can only be reported
            if also the tunnel is reported to provide
            the complete reference.";
        }
        description
        "
            Reference to the p2p-secondary-path that reports
            the transient state of the computed path.

            If no transient state is created,
            this attribute is empty.
        ";
    }
}
}
```

```
}

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

identity svec-metric-cumul-te {
  base svec-metric-type;
  description
    "TE cumulative path metric";
}

identity svec-metric-cumul-igp {
  base svec-metric-type;
  description
    "IGP cumulative path metric";
}

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

identity svec-metric-aggregate-bandwidth-consumption {
  base svec-metric-type;
  description
    "Cumulative bandwidth consumption of the set of
    synchronized paths";
}

identity svec-metric-load-of-the-most-loaded-link {
  base svec-metric-type;
  description
    "Load of the most loaded link";
}

grouping svec-metrics-bounds_config {
```

```
description
  "TE path metric bounds grouping for computing a set of
  synchronized requests";
leaf metric-type {
  type identityref {
    base svec-metric-type;
  }
  description "TE path metric type usable for computing a set of
  synchronized requests";
}
leaf upper-bound {
  type uint64;
  description "Upper bound on end-to-end svec path metric";
}
}

grouping svec-metrics-optimization_config {
  description
    "TE path metric bounds grouping for computing a set of
    synchronized requests";

  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";
  }
}

grouping svec-exclude {
  description "List of resources to be excluded by all the paths
  in the SVEC";
  container exclude-objects {
    description "resources to be excluded";
  }
}
```



```
    list excludes {
      key index;
      ordered-by user;
      leaf index {
        type uint32;
        description "XRO subobject index";
      }
      description
        "List of explicit route objects to always exclude
        from synchronized path computation";
      uses te-types:explicit-route-hop;
    }
  }
}

grouping synchronization-constraints {
  description "Global constraints applicable to synchronized
  path computation";
  container svec-constraints {
    description "global svec constraints";
    list path-metric-bound {
      key metric-type;
      description "list of bound metrics";
      uses svec-metrics-bounds_config;
    }
  }
  uses te-types:generic-path-srlgs;
  uses svec-exclude;
}

grouping synchronization-optimization {
  description "Synchronized request optimization";
  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";
        uses svec-metrics-optimization_config;
      }
    }
    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-info {
  description "Information for sync";
  list synchronization {
    key "synchronization-id";
    description "sync list";
    leaf synchronization-id {
      type uint32;
      description "index";
    }
  }
  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 no-path-info {
    description "no-path-info";
    container no-path {
        presence "Response without path information, due to failure
         performing the path computation";
        description "if path computation cannot identify a path,
         rpc returns no path.";
    }
}

/*
 * These groupings should be removed when defined in te-types
 */

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-p2p-common-params {
  description
    "Common grouping to define the TE tunnel parameters";

  uses encoding-and-switching-type;
  leaf source {
    type inet:ip-address;
    description "TE tunnel source address.";
  }
  leaf destination {
    type inet:ip-address;
    description "P2P tunnel destination address";
  }
  leaf src-tp-id {
    type binary;
    description
      "TE tunnel source termination point identifier.";
  }
  leaf dst-tp-id {
    type binary;
    description
      "TE tunnel destination termination point identifier.";
```

```
    }
    leaf bidirectional {
      type boolean;
      default 'false';
      description "TE tunnel bidirectional";
    }
  }

/*
 * AUGMENTS TO TE RPC
 */

augment "/te:tunnels-rpc/te:input/te:tunnel-info" {
  description "Path Computation RPC input";
  list path-request {
    key "request-id";
    description "request-list";
    leaf request-id {
      type uint32;
      mandatory true;
      description
        "Each path computation request is uniquely identified
        by the request-id-number.";
    }
    uses tunnel-p2p-common-params;
    uses te-types:te-topology-identifier;
    uses te-types:path-constraints-route-objects;
    uses te-types:generic-path-constraints;
    uses te-types:generic-path-optimization;
    uses te:path-access-segment-info;
    uses requested-info;
    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).";
      uses requested-state;
    }
  }
}
```

```
    }
    uses synchronization-info;
}

augment "/te:tunnels-rpc/te:output/te: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.";
    }
    choice response-type {
      config false;
      description "response-type";
      case no-path-case {
        uses no-path-info;
      }
      case path-case {
        container computed-path {
          uses path-info;
          uses reported-state;
          description "Path computation service.";
        }
      }
    }
  }
}

augment "/te:tunnels-rpc/te:input/te:tunnel-info" {
  description "Path Delete RPC input";
  leaf-list deleted-paths-transaction-id {
    type string;
    description
      "The list of the transaction-id values of the
```

```
        transient states to be deleted";
    }
}

augment "/te:tunnels-rpc/te:output/te:result" {
    description "Path Delete RPC output";
    leaf-list deleted-paths-transaction-id {
        type string;
        description
            "The list of the transaction-id values of the
             transient states that have been successfully deleted";
    }
}
}
}
<CODE ENDS>
```

Figure 12 - TE path computation YANG module

## 7. Security Considerations

This document describes use cases of requesting Path Computation using YANG 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 draft 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].

This document also defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC7950] apply for this document as well.

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.

Note - The security analysis of each leaf is for further study.

## 8. 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-path-computation  
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: tepc

## 9. References

### 9.1. Normative References

- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [RFC5440] Vasseur, JP., Le Roux, JL. et al., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.
- [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. et al., "Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)", RFC 5541, June 2009.



- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8341] Bierman, A., and M. Bjorklund, "Network Configuration Access Control Model", RFC 8341, March 2018.
- [RFC7491] Farrel, A., King, D., "A PCE-Based Architecture for Application-Based Network Operations", RFC 7491, March 2015.
- [RFC7926] Farrel, A. et al., "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", RFC 7926, July 2016.
- [RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018.
- [RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.
- [RFC8454] Lee, Y. et al., "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC8454, September 2018.
- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, 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.

### 9.1. Informative References

- [RFC4655] Farrel, A. et al., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
- [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>>.
- [RFC7139] Zhang, F. et al., "GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks", RFC 7139, March 2014.
- [RFC7446] Lee, Y. et al., "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, February 2015.
- [RFC8233] Dhody, D. et al., "Extensions to the Path Computation Element Communication Protocol (PCEP) to Compute Service-Aware Label Switched Paths (LSPs)", RFC 8233, September 2017
- [RFC8342] Bjorklund, M. et al. "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018
- [OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang, work in progress.
- [ITU-T G.709-2016] ITU-T Recommendation G.709 (06/16), "Interface for the optical transport network", June 2016.

### 10. 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 model [TE-TUNNEL], 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 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 setup is not necessarily the same as the path that have 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 setup meets the required end-to-end metrics and constraints.

This document was prepared using 2-Word-v2.0.template.dot.

## Appendix A. Examples of dimensioning the "detailed connectivity matrix"

In the following table, a list of the possible constraints, associated with their potential cardinality, is reported.

The maximum number of potential connections to be computed and reported is, in first approximation, the multiplication of all of them.

Constraint	Cardinality
End points	$N(N-1)/2$ if connections are bidirectional (OTN and WDM), $N(N-1)$ for unidirectional connections.
Bandwidth	In WDM networks, bandwidth values are expressed in GHz.  On fixed-grid WDM networks, the central frequencies are on a 50GHz grid and the channel width of the transmitters are typically 50GHz such that each central frequency can be used, i.e., adjacent channels can be placed next to each other in terms of central frequencies.  On flex-grid WDM networks, the central frequencies are on a 6.25GHz grid and the channel width of the transmitters can be multiples of 12.5GHz.  For fixed-grid WDM networks typically there is only one possible bandwidth value (i.e., 50GHz) while for flex-grid WDM networks typically there are 4 possible bandwidth values (e.g., 37.5GHz, 50GHz, 62.5GHz, 75GHz).  In OTN (ODU) networks, bandwidth values are expressed as pairs of ODU type and, in case of ODUFlex, ODU rate in bytes/sec as described in section 5 of [RFC7139].  For "fixed" ODUk types, 6 possible bandwidth values are possible (i.e., ODU0, ODU1, ODU2, ODU2e, ODU3, ODU4).  For ODUFlex(GFP), up to 80 different bandwidth values can be specified, as defined in Table 7-8 of [ITU-T G.709-2016].  For other ODUFlex types, like ODUFlex(CBR), the number of possible bandwidth values depends on the rates of the

clients that could be mapped over these ODUFlex types, as shown in Table 7.2 of [ITU-T G.709-2016], which in theory could be a countinuum of values. However, since different ODUFlex bandwidths that use the same number of TSs on each link along the path are equivalent for path computation purposes, up to 120 different bandwidth ranges can be specified.

Ideas to reduce the number of ODUFlex bandwidth values in the detailed connectivity matrix, to less than 100, are for further study.

Bandwidth specification for ODUCn is currently for further study but it is expected that other bandwidth values can be specified as integer multiples of 100Gb/s.

In IP we have bandwidth values in bytes/sec. In principle, this is a countinuum of values, but in practice we can identify a set of bandwidth ranges, where any bandwidth value inside the same range produces the same path.

The number of such ranges is the cardinality, which depends on the topology, available bandwidth and status of the network. Simulations (Note: reference paper submitted for publication) show that values for medium size topologies (around 50-150 nodes) are in the range 4-7 (5 on average) for each end points couple.

**Metrics** IGP, TE and hop number are the basic objective metrics defined so far. There are also the 2 objective functions defined in [RFC5541]: Minimum Load Path (MLP) and Maximum Residual Bandwidth Path (MBP). Assuming that one only metric or objective function can be optimized at once, the total cardinality here is 5.

With [RFC8233], a number of additional metrics are defined, including Path Delay metric, Path Delay Variation metric and Path Loss metric, both for point-to-point and point-to-multipoint paths. This increases the cardinality to 8.

**Bounds** Each metric can be associated with a bound in order to find a path having a total value of that metric lower than the given bound. This has a potentially very high cardinality (as any value for the bound is allowed). In

practice there is a maximum value of the bound (the one with the maximum value of the associated metric) which results always in the same path, and a range approach like for bandwidth in IP should produce also in this case the cardinality. Assuming to have a cardinality similar to the one of the bandwidth (let say 5 on average) we should have 6 (IGP, TE, hop, path delay, path delay variation and path loss; we don't consider here the two objective functions of [RFC5541] as they are conceived only for optimization)\*5 = 30 cardinality.

#### Technology

constraints For further study

**Priority** We have 8 values for setup priority, which is used in path computation to route a path using free resources and, where no free resources are available, resources used by LSPs having a lower holding priority.

**Local prot** It's possible to ask for a local protected service, where all the links used by the path are protected with fast reroute (this is only for IP networks, but line protection schemas are available on the other technologies as well). This adds an alternative path computation, so the cardinality of this constraint is 2.

#### Administrative

**Colors** Administrative colors (aka affinities) are typically assigned to links but when topology abstraction is used affinity information can also appear in the detailed connectivity matrix.

There are 32 bits available for the affinities. Links can be tagged with any combination of these bits, and path computation can be constrained to include or exclude any or all of them. The relevant cardinality is 3 (include-any, exclude-any, include-all) times  $2^{32}$  possible values. However, the number of possible values used in real networks is quite small.

#### Included Resources

A path computation request can be associated to an ordered set of network resources (links, nodes) to be included along the computed path. This constraint would

have a huge cardinality as in principle any combination of network resources is possible. However, as far as the Orchestrator doesn't know details about the internal topology of the domain, it shouldn't include this type of constraint at all (see more details below).

#### Excluded Resources

A path computation request can be associated to a set of network resources (links, nodes, SRLGs) to be excluded from the computed path. Like for included resources, this constraint has a potentially very high cardinality, but, once again, it can't be actually used by the Orchestrator, if it's not aware of the domain topology (see more details below).

As discussed above, the Orchestrator can specify include or exclude resources depending on the abstract topology information that the domain controller exposes:

- o In case the domain controller exposes the entire domain as a single abstract TE node with his own external terminations and detailed connectivity matrix (whose size we are estimating), no other topological details are available, therefore the size of the detailed connectivity matrix only depends on the combination of the constraints that the Orchestrator can use in a path computation request to the domain controller. These constraints cannot refer to any details of the internal topology of the domain, as those details are not known to the Orchestrator and so they do not impact size of the detailed connectivity matrix exported.

- o Instead in case the domain controller exposes a topology including more than one abstract TE nodes and TE links, and their attributes (e.g. SRLGs, affinities for the links), the Orchestrator knows these details and therefore could compute a path across the domain referring to them in the constraints. The detailed connectivity matrixes, whose size need to be estimated here, are the ones relevant to the abstract TE nodes exported to the Orchestrator. These detailed connectivity matrixes and therefore theirs sizes, while cannot depend on the other abstract TE nodes and TE links, which are external to the given abstract node, could depend to SRLGs (and other attributes, like affinities) which could be present also in the portion of the topology represented by the abstract nodes, and therefore contribute to the size of the related detailed connectivity matrix.

We also don't consider here the possibility to ask for more than one path in diversity or for point-to-multi-point paths, which are for further study.

Considering for example an IP domain without considering SRLG and affinities, we have an estimated number of paths depending on these estimated cardinalities:

Endpoints =  $N*(N-1)$ , Bandwidth = 5, Metrics = 6, Bounds = 20,  
Priority = 8, Local prot = 2

The number of paths to be pre-computed by each IP domain is therefore  $24960 * N(N-1)$  where N is the number of domain access points.

This means that with just 4 access points we have nearly 300000 paths to compute, advertise and maintain (if a change happens in the domain, due to a fault, or just the deployment of new traffic, a substantial number of paths need to be recomputed and the relevant changes advertised to the upper controller).

This seems quite challenging. In fact, if we assume a mean length of 1K for the json describing a path (a quite conservative estimate), reporting 300000 paths means transferring and then parsing more than 300 Mbytes for each domain. If we assume that 20% (to be checked) of this paths change when a new deployment of traffic occurs, we have 60 Mbytes of transfer for each domain traversed by a new end-to-end path. If a network has, let say, 20 domains (we want to estimate the load for a non-trivial domain setup) in the beginning a total



initial transfer of 6Gigs is needed, and eventually, assuming 4-5 domains are involved in mean during a path deployment we could have 240-300 Mbytes of changes advertised to the higher order controller.

Further bare-bone solutions can be investigated, removing some more options, if this is considered not acceptable; in conclusion, it seems that an approach based only on the information provided by the detailed connectivity matrix is hardly feasible, and could be applicable only to small networks with a limited meshing degree between domains and renouncing to a number of path computation features.

#### Contributors

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  
Huawei  
Email: leeyoung@huawei.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  
Email: italo.busi@huawei.com

Sergio Belotti (Editor)  
Nokia  
Email: sergio.belotti@nokia.com

Victor Lopez  
Telefonica  
Email: victor.lopezalvarez@telefonica.com

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

Anurag Sharma  
Google  
Email: ansha@google.com

Yan Shi  
China Unicom  
Email: shiyan49@chinaunicom.cn

Ricard Vilalta  
CTTC  
Email: ricard.vilalta@cttc.es

Karthik Sethuraman  
NEC  
Email: karthik.sethuraman@necam.com

Michael Scharf  
Nokia  
Email: michael.scharf@gmail.com

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



TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 5, 2020

V. Beeram  
T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems, Inc.  
X. Liu  
Jabil  
I. Bryskin  
Huawei Technologies  
July 04, 2019

A YANG Data Model for Resource Reservation Protocol (RSVP)  
draft-ietf-teas-yang-rsvp-11

Abstract

This document defines a YANG data model for the configuration and management of RSVP Protocol. The model covers the building blocks of the RSVP protocol that can be augmented and used by other RSVP extension models such as RSVP extensions to Traffic-Engineering (RSVP-TE). The model covers the configuration, operational state, remote procedure calls, and event notifications data.

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 January 5, 2020.

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
1.1. Terminology . . . . .	3
1.2. Model Tree Diagram . . . . .	3
1.3. Prefixes in Data Node Names . . . . .	3
2. Model Overview . . . . .	3
2.1. Module(s) Relationship . . . . .	4
2.2. Design Considerations . . . . .	4
2.3. Model Notifications . . . . .	5
2.4. RSVP Base YANG Model . . . . .	5
2.4.1. Tree Diagram . . . . .	7
2.4.2. YANG Module . . . . .	11
2.5. RSVP Extended YANG Model . . . . .	31
2.5.1. Tree Diagram . . . . .	31
2.5.2. YANG Module . . . . .	33
3. IANA Considerations . . . . .	44
4. Security Considerations . . . . .	45
5. Acknowledgement . . . . .	46
6. Contributors . . . . .	46
7. Normative References . . . . .	46
Authors' Addresses . . . . .	49

## 1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) 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 that can be used to configure and manage the RSVP protocol [RFC2205]. This model covers RSVP protocol building blocks that can be augmented and used by other RSVP extension models- such as for signaling RSVP-TE MPLS (or other technology specific) Label Switched Paths (LSP)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. 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].

### 1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]
rt-type	ietf-routing-types	XX
key-chain	ietf-key-chain	XX

Table 1: Prefixes and corresponding YANG modules

## 2. Model Overview

The RSVP base YANG module augments the "control-plane-protocol" list in ietf-routing [RFC8349] module with specific RSVP parameters in an "rsvp" container. It also defines an extension identity "rsvp" of base "rt:routing-protocol" to identify the RSVP protocol.

The augmentation of the RSVP model by other models (e.g. RSVP-TE for MPLS or other technologies) are outside the scope of this document and are discussed in separate document(s), e.g. [I-D.ietf-teas-yang-rsvp-te].

## 2.1. Module(s) Relationship

This document divides the RSVP model into two modules: base and extended RSVP modules. Some RSVP features are categorized as core to the function of the protocol and are supported by most vendors claiming the support for RSVP protocol. Such features configuration and state are grouped in the RSVP base module.

Other extended RSVP features are categorized as either optional or providing ability to better tune the basic functionality of the RSVP protocol. The support for extended RSVP features by all vendors is considered optional. Such features are grouped in a separate RSVP extended module.

The relationship between the base and extended RSVP YANG model and the IETF routing YANG model is shown in Figure 1.

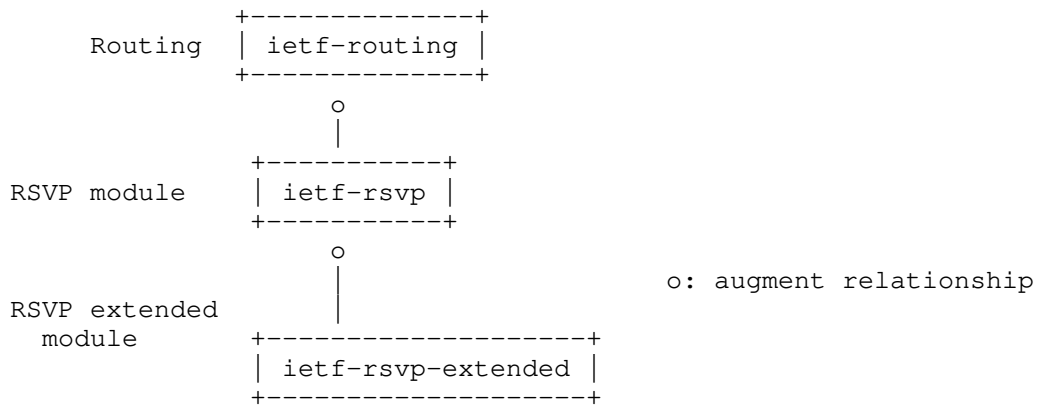


Figure 1: Relationship of RSVP and RSVP extended modules with other protocol modules

## 2.2. Design Considerations

The RSVP base model does not aim to be feature complete. The primary intent is to cover a set of standard core features that are commonly in use. For example:

- o Authentication ([RFC2747])
- o Refresh Reduction ([RFC2961])
- o Hellos ([RFC3209])
- o Graceful Restart ([RFC3473], [RFC5063])



The extended RSVP YANG model covers the configuration for optional features that are not must for basic RSVP protocol operation.

The defined data model supports configuration inheritance for neighbors, and interfaces. Data elements defined in 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). Vendors are expected to augment the above container(s) to provide the list of inheritance command for their implementations.

### 2.3. Model Notifications

Notifications data modeling is key in any defined data model.

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

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

### 2.4. RSVP Base YANG Model

The RSVP base YANG data model defines the container "rsvp" as the top level container in this data model. The presence of this container enables the RSVP protocol functionality.

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 globals
      .
      .
    +--rw interfaces
      .
      +-- ro <<derived state associated with interfaces>>
      .
      .
    +--rw neighbors
      .
      +-- ro <<derived state associated with the tunnel>>
      .
      .
    +--rw sessions
      .
      +-- ro <<derived state associated with the tunnel>>
      .
  rpcs:
    +--x clear-session
    +--x clear-neighbor
```

Figure 2: RSVP high-level tree model view

Configuration and state data are grouped to those applicable on per node (global), per interface, per neighbor, or per session.

#### Global Data:

The global data cover the configuration and state that is applicable the RSVP protocol behavior.

#### Interface Data:

The interface data configuration and state model relevant attributes applicable to one or all RSVP interfaces. Any data or state at the "interfaces" container level is equally applicable to all interfaces - unless overridden by explicit configuration or state under a specific interface.

#### Neighbor Data:

The neighbor data cover configuration and state relevant to RSVP neighbors. Neighbors can be dynamically discovered using RSVP signaling or explicitly configured.

## Session Data:

The sessions data branch covers configuration and state relevant to RSVP sessions. This is usually derived state that is result of signaling. This model defines attributes related to IP RSVP sessions as defined in [RFC2205].

## 2.4.1. Tree Diagram

Figure 3 shows the YANG tree representation for configuration and state data that is augmenting the RSVP basic module:

```

module: ietf-rsvp
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol:
      +--rw rsvp!
        +--rw globals
          +--rw sessions
            +--ro session-ip*
              [destination protocol-id destination-port]
              +--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-state?         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
            +--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 summary-refresh-sent?     yang:counter64
| | | +--ro summary-refresh-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 graceful-restart
| | +--rw enabled?                    boolean
+--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 summary-refresh-sent?           yang:counter64
| | | +--ro summary-refresh-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* [interface]
| +--rw interface                          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 summary-refresh-sent?            yang:counter64
    +--ro summary-refresh-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 neighbors
  +--rw neighbor* [address]
    +--rw address                        inet:ip-address
    +--rw epoch?                        uint32
    +--rw expiry-time?                  uint32
    +--rw graceful-restart
      +--rw enabled?                    boolean
      +--rw local-restart-time?         uint32
      +--rw local-recovery-time?        uint32
      +--rw neighbor-restart-time?      uint32
      +--rw neighbor-recovery-time?     uint32
    +--rw helper-mode
      +--rw enabled?                    boolean
      +--rw max-helper-restart-time?    uint32
      +--rw max-helper-recovery-time?   uint32
      +--rw neighbor-restart-time-remaining? uint32
      +--rw neighbor-recovery-time-remaining? uint32
    +--rw hello-status?                 enumeration
    +--rw interface?                   if:interface-ref
    +--rw neighbor-state?               enumeration
    +--rw refresh-reduction-capable?    boolean

```

```

        +---rw restart-count?                yang:counter32
        +---rw restart-time?                 yang:date-and-time

rpcs:
  +---x clear-session
  |   +---w input
  |   |   +---w routing-protocol-instance-name    leafref
  |   |   +---w (filter-type)
  |   |   |   +---:(match-all)
  |   |   |   |   +---w all                        empty
  |   |   +---:(match-one)
  |   |   |   +---w session-info
  |   |   |   |   +---w (session-type)
  |   |   |   |   |   +---:(rsvp-session-ip)
  |   |   |   |   |   |   +---w destination        leafref
  |   |   |   |   |   |   +---w protocol-id        uint8
  |   |   |   |   |   |   +---w destination-port    inet:ip-address
  |   +---x clear-neighbor
  |   |   +---w input
  |   |   |   +---w routing-protocol-instance-name    leafref
  |   |   |   +---w (filter-type)
  |   |   |   |   +---:(match-all)
  |   |   |   |   |   +---w all                        empty
  |   |   |   +---:(match-one)
  |   |   |   |   +---w neighbor-address              leafref

```

Figure 3: RSVP model tree diagram

#### 2.4.2. YANG Module

The ietf-rsvp module imports from the following modules:

- o ietf-interfaces defined in [RFC8343]
- o ietf-yang-types and ietf-inet-types defined in [RFC6991]
- o ietf-routing defined in [RFC8349]
- o ietf-key-chain defined in [RFC8177]

```

<CODE BEGINS> file "ietf-rsvp@2019-07-04.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";
}

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:Igor.Bryskin@huawei.com>

Editor: Himanshu Shah  
<mailto:hshah@ciena.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 "2019-07-04" {  
  description  
    "A YANG Data Model for Resource Reservation Protocol";  
  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-config {
  description
    "Base configuration parameters relating to RSVP
    Graceful-Restart";
  leaf enabled {
    type boolean;
    description
      "'true' if RSVP Graceful Restart is enabled.
      'false' if RSVP Graceful Restart is disabled.";
  }
}

grouping graceful-restart {
  description
    "RSVP graceful restart parameters grouping";
  container graceful-restart {
    description
      "RSVP graceful restart parameters container";
    uses graceful-restart-config;
  }
}

grouping refresh-reduction-config {
  description
```

```
    "Configuration parameters relating to RSVP
    refresh reduction";

    leaf enabled {
      type boolean;
      description
        "'true' if RSVP Refresh Reduction is enabled.
        'false' if RSVP Refresh Reduction is disabled.";
    }
  }

  grouping refresh-reduction {
    description
      "Top level grouping for RSVP refresh reduction
      parameters";
    container refresh-reduction {
      description
        "Top level container for RSVP refresh reduction
        parameters";
      uses refresh-reduction-config;
    }
  }

  grouping authentication-config {
    description
      "Configuration parameters relating to RSVP
      authentication";
    leaf enabled {
      type boolean;
      description
        "'true' if RSVP Authentication is enabled.
        'false' if RSVP Authentication is disabled.";
    }
    leaf authentication-key {
      type string;
      description
        "An authentication key string";
      reference
        "RFC 2747: RSVP Cryptographic Authentication";
    }
    leaf crypto-algorithm {
      type identityref {
        base key-chain:crypto-algorithm;
      }
      mandatory true;
      description
        "Cryptographic algorithm associated with key.";
    }
  }
```

```
}

grouping authentication {
  description
    "Top level grouping for RSVP authentication parameters";
  container authentication {
    description
      "Top level container for RSVP authentication
      parameters";
    uses authentication-config;
  }
}

grouping hellos-config {
  description
    "Configuration parameters relating to RSVP
    hellos";
  leaf enabled {
    type boolean;
    description
      "'true' if RSVP Hello is enabled.
      'false' if RSVP Hello is disabled.";
  }
}

grouping hellos {
  description
    "Top level grouping for RSVP hellos parameters";
  container hellos {
    description
      "Top level container for RSVP hello parameters";
    uses hellos-config;
  }
}

grouping signaling-parameters-config {
  description
    "Configuration parameters relating to RSVP
    signaling";
}

grouping signaling-parameters {
  description
    "Top level grouping for RSVP signaling parameters";
  uses signaling-parameters-config;
}

grouping session-attributes-state {
```

```
description
  "Top level grouping for RSVP session properties";
leaf destination-port {
  type inet:port-number;
  description "RSVP destination port";
  reference "RFC2205";
}
leaf protocol-id {
  type uint8;
  description "The IP protocol ID.";
  reference "RFC2205, section 3.2";
}
leaf source {
  type inet:ip-address;
  description "RSVP source address";
  reference "RFC2205";
}
leaf destination {
  type inet:ip-address;
  description "RSVP destination address";
  reference "RFC2205";
}
leaf session-name {
  type string;
  description
    "The signaled name of this RSVP session.";
}
leaf session-state {
  type enumeration {
    enum "up" {
      description
        "RSVP session is up";
    }
    enum "down" {
      description
        "RSVP session is down";
    }
  }
  description
    "Enumeration of RSVP session states";
}
leaf session-type {
  type identityref {
    base rsvp-session-type;
  }
  description "RSVP session type";
}
container psbs {
```

```
description "Path State Block container";
list psb {
  description "List of path state blocks";
  leaf source-port {
    type inet:port-number;
    description "RSVP source port";
    reference "RFC2205";
  }
  leaf expires-in {
    type uint32;
    units seconds;
    description "Time to reservation expiry (in seconds)";
  }
}
}
container rsbs {
  description "Reservation State Block container";
  list rsb {
    description "List of reservation state blocks";
    leaf source-port {
      type inet:port-number;
      description "RSVP source port";
      reference "RFC2205";
    }
    leaf reservation-style {
      type identityref {
        base reservation-style;
      }
      description "RSVP reservation style";
    }
    leaf expires-in {
      type uint32;
      units seconds;
      description "Time to reservation expiry (in seconds)";
    }
  }
}
}

grouping neighbor-attributes {
  description
    "Top level grouping for RSVP neighbor properties";
  leaf address {
    type inet:ip-address;
    description
      "Address of RSVP neighbor";
  }
}
```

```
leaf epoch {
    type uint32;
    description
        "Neighbor epoch.";
}

leaf expiry-time {
    type uint32;
    units seconds;
    description
        "Neighbor expiry time after which the neighbor state
        is purged if no states associated with it";
}

container graceful-restart {
    description
        "Graceful restart information.";

    leaf enabled {
        type boolean;
        description
            "'true' if graceful restart is enabled for the neighbor.";
    }

    leaf local-restart-time {
        type uint32;
        units seconds;
        description
            "Local node restart time";
    }

    leaf local-recovery-time {
        type uint32;
        units seconds;
        description
            "Local node recover time";
    }

    leaf neighbor-restart-time {
        type uint32;
        units seconds;
        description
            "Neighbor restart time";
    }

    leaf neighbor-recovery-time {
        type uint32;
        units seconds;
    }
}
```

```
    description
      "Neighbor recover time";
  }

  container helper-mode {
    description
      "Helper mode information ";

    leaf enabled {
      type boolean;
      description
        "'true' if helper mode is enabled.";
    }

    leaf max-helper-restart-time {
      type uint32;
      units seconds;
      description
        "The time the router or switch waits after it
        discovers that a neighboring router has gone down
        before it declares the neighbor down";
    }

    leaf max-helper-recovery-time {
      type uint32;
      units seconds;
      description
        "The amount of time the router retains the state of its
        RSVP neighbors while they undergo a graceful restart";
    }

    leaf neighbor-restart-time-remaining {
      type uint32;
      units seconds;
      description
        "Number of seconds remaining for neighbor to send
        Hello message after restart.";
    }

    leaf neighbor-recovery-time-remaining {
      type uint32;
      units seconds;
      description
        "Number of seconds remaining for neighbor to
        refresh.";
    }
  } // helper-mode
} // graceful-restart
```



```
leaf hello-status {
  type enumeration {
    enum "enabled" {
      description
        "Enabled";
    }
    enum "disabled" {
      description
        "Disabled";
    }
    enum "restarting" {
      description
        "Restarting";
    }
  }
  description
    "Hello status";
}

leaf interface {
  type if:interface-ref;
  description
    "Interface where RSVP neighbor was detected";
}

leaf neighbor-state {
  type enumeration {
    enum "up" {
      description
        "up";
    }
    enum "down" {
      description
        "down";
    }
    enum "hello-disable" {
      description
        "hello-disable";
    }
    enum "restarting" {
      description
        "restarting";
    }
  }
  description
    "Neighbor state";
}
```

```
leaf refresh-reduction-capable {
  type boolean;
  description
    "enables all RSVP refresh reduction message
    bundling, RSVP message ID, reliable message delivery
    and summary refresh";
  reference
    "RFC 2961 RSVP Refresh Overhead Reduction
    Extensions";
}

leaf restart-count {
  type yang:counter32;
  description
    "Number of times this neighbor restart";
}

leaf restart-time {
  type yang:date-and-time;
  description
    "Last restart time of the neighbor";
}

grouping packets-state {
  description
    "Packet statistics grouping";
  container packets {
    description
      "Packet statistics container";
    leaf sent {
      type yang:counter64;
      description
        "Packet sent count";
    }

    leaf received {
      type yang:counter64;
      description
        "Packet received count";
    }
  }
}

grouping protocol-state {
  description
    "RSVP protocol statistics grouping";
  container messages {
```

```
description
  "RSVP protocol statistics container";
leaf ack-sent {
  type yang:counter64;
  description
    "Hello sent count";
}

leaf ack-received {
  type yang:counter64;
  description
    "Hello received count";
}

leaf bundle-sent {
  type yang:counter64;
  description
    "Bundle sent count";
}

leaf bundle-received {
  type yang:counter64;
  description
    "Bundle received count";
}

leaf hello-sent {
  type yang:counter64;
  description
    "Hello sent count";
}

leaf hello-received {
  type yang:counter64;
  description
    "Hello received count";
}

leaf integrity-challenge-sent {
  type yang:counter64;
  description
    "Integrity Challenge sent count";
}

leaf integrity-challenge-received {
  type yang:counter64;
  description
    "Integrity Challenge received count";
}
```

```
    }

    leaf integrity-response-sent {
      type yang:counter64;
      description
        "Integrity Response sent count";
    }

    leaf integrity-response-received {
      type yang:counter64;
      description
        "Integrity Response received count";
    }

    leaf notify-sent {
      type yang:counter64;
      description
        "Notify sent count";
    }

    leaf notify-received {
      type yang:counter64;
      description
        "Notify received count";
    }

    leaf path-sent {
      type yang:counter64;
      description
        "Path sent count";
    }

    leaf path-received {
      type yang:counter64;
      description
        "Path received count";
    }

    leaf path-err-sent {
      type yang:counter64;
      description
        "Path error sent count";
    }

    leaf path-err-received {
      type yang:counter64;
      description
        "Path error received count";
    }
```

```
    }

    leaf path-tear-sent {
      type yang:counter64;
      description
        "Path tear sent count";
    }

    leaf path-tear-received {
      type yang:counter64;
      description
        "Path tear received count";
    }

    leaf resv-sent {
      type yang:counter64;
      description
        "Resv sent count";
    }

    leaf resv-received {
      type yang:counter64;
      description
        "Resv received count";
    }

    leaf resv-confirm-sent {
      type yang:counter64;
      description
        "Confirm sent count";
    }

    leaf resv-confirm-received {
      type yang:counter64;
      description
        "Confirm received count";
    }

    leaf resv-err-sent {
      type yang:counter64;
      description
        "Resv error sent count";
    }

    leaf resv-err-received {
      type yang:counter64;
      description
        "Resv error received count";
    }
```

```
    }

    leaf resv-tear-sent {
      type yang:counter64;
      description
        "Resv tear sent count";
    }

    leaf resv-tear-received {
      type yang:counter64;
      description
        "Resv tear received count";
    }

    leaf summary-refresh-sent {
      type yang:counter64;
      description
        "Summary refresh sent count";
    }

    leaf summary-refresh-received {
      type yang:counter64;
      description
        "Summary refresh received count";
    }

    leaf unknown-messages-received {
      type yang:counter64;
      description
        "Unknown packet received count";
    }
  }
}

grouping errors-state {
  description
    "Error statistics state grouping";
  container errors {
    description
      "Error statistics state container";
    leaf authenticate {
      type yang:counter64;
      description
        "The total number of packets received with an
        authentication failure.";
    }

    leaf checksum {
```

```
        type yang:counter64;
        description
            "The total number of 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-state {
    description "RSVP statistic attributes.";
    container statistics {
        config false;
        description
            "statistics state container";
        uses protocol-state;
        uses packets-state;
        uses errors-state;
    }
}

grouping neighbor-derived-state {
    description
        "Derived state at neighbor level.";
}

grouping global-attributes {
    description
        "Top level grouping for RSVP global properties";
    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-state;
        }
    }
}
```

```
    uses statistics-state;
  }

  grouping intf-attributes {
    description
      "Top level grouping for RSVP interface properties";
    uses signaling-parameters;
    uses refresh-reduction;
    uses hellos;
    uses authentication;
    uses statistics-state;
  }

  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 globals {
        description "RSVP global properties.";
        uses global-attributes;
        uses graceful-restart;
      }

      container interfaces {
        description
          "RSVP interfaces container";
        uses intf-attributes;

        list interface {
          key "interface";
          description
            "RSVP interfaces.";
          leaf interface {
            type if:interface-ref;
            description
              "RSVP interface.";
          }
          uses intf-attributes;
        }
      }
    }
  }
```



```
    container neighbors {
      description "RSVP neighbors container";
      list neighbor {
        key "address";
        description "List of RSVP neighbors";
        uses neighbor-attributes;
      }
    }
  }
}

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:globals" +
        "/rsvp:sessions/rsvp:session-ip/destination";
    }
    mandatory true;
    description "RSVP session";
  }
  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 {
  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 operation on";
                choice session-type {
                    mandatory true;
                    description "RSVP session type";
                    case rsvp-session-ip {
                        uses session-ref;
                    }
                }
            }
        }
    }
}

rpc clear-neighbor {
    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 "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 specific RSVP neighbor session";
            }
        }
    }
}
}
}
<CODE ENDS>

```

## 2.5. RSVP Extended YANG Model

The RSVP extended YANG model covers non-core RSVP feature(s). It also covers feature(s) that are not necessarily supported by all vendors, and hence, can be guarded with "if-feature" checks.

### 2.5.1. Tree Diagram

Figure 4 shows the YANG tree representation for configuration and state data that is augmenting the RSVP extended module:

```

module: ietf-rsvp-extended
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:graceful-restart:
      +--rw restart-time?    uint32
      +--rw recovery-time?   uint32
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals

```

```

        /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
        /rsvp:statistics/rsvp:messages:
augment /rt:routing/rt:control-plane-protocols
        /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
        /rsvp:statistics/rsvp:errors:
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?                  boolean
    +--rw patherr-state-removal?     empty
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 reliable-ack-hold-time?    uint32
    +--rw reliable-ack-max-size?     uint32
    +--rw reliable-retransmit-time?  uint32
    +--rw reliable-srefresh?         empty
    +--rw summary-max-size?          uint32
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?                  boolean
    +--rw patherr-state-removal?     empty

```

```

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 reliable-ack-hold-time?     uint32
      +--rw reliable-ack-max-size?     uint32
      +--rw reliable-retransmit-time?   uint32
      +--rw reliable-srefresh?          empty
      +--rw summary-max-size?          uint32
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 model tree diagram

### 2.5.2. YANG Module

The ietf-rsvp-extended module imports from the following modules:

- o ietf-rsvp defined in this document
- o ietf-routing defined in [RFC8349]
- o ietf-yang-types and ietf-inet-types defined in [RFC6991]
- o ietf-key-chain defined in [RFC8177]

Figure 5 shows the RSVP extended YANG module:

```

<CODE BEGINS> file "ietf-rsvp-extended@2019-07-04.yang"
module ietf-rsvp-extended {
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-rsvp-extended";

  prefix "rsvp-ext";

```

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

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

  Editor:     Xia Chen
```

<mailto:jescia.chenxia@huawei.com>

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

Editor: Bin Wen  
<mailto:Bin\_Wen@cable.comcast.com>;

#### description

"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 "2019-07-04" {  
  description  
    "A YANG Data Model for Extended Resource Reservation  
    Protocol";  
  reference  
    "RFCXXXX: A YANG Data Model for Resource Reservation Protocol  
    (RSVP)";  
}
```

```
/* RSVP features */  
feature authentication {  
  description  
    "Indicates support for RSVP authentication";  
}
```

```
feature error-statistics {  
  description
```

```
    "Indicates support for error statistics";
}

feature global-statistics {
  description
    "Indicates support for global statistics";
}

feature graceful-restart {
  description
    "Indicates support for RSVP graceful restart";
}

feature hellos {
  description
    "Indicates support for RSVP hellos (RFC3209).";
}

feature notify {
  description
    "Indicates support for RSVP notify message (RFC3473).";
}

feature refresh-reduction {
  description
    "Indicates support for RSVP refresh reduction (RFC2961).";
}

feature refresh-reduction-extended {
  description
    "Indicates support for RSVP refresh reduction (RFC2961).";
}

feature per-interface-statistics {
  description
    "Indicates support for per interface statistics";
}

grouping graceful-restart-extended-config {
  description
    "Configuration parameters relating to RSVP
    Graceful-Restart";
  leaf restart-time {
    type uint32;
    units seconds;
    description
      "Graceful restart time (seconds).";
    reference

```



```
        "RFC 5495: Description of the Resource
        Reservation Protocol - Traffic-Engineered
        (RSVP-TE) Graceful Restart Procedures";
    }
    leaf recovery-time {
        type uint32;
        units seconds;
        description
            "RSVP state recovery time";
    }
}

grouping authentication-extended-config {
    description
        "Configuration parameters relating to RSVP
        authentication";
    leaf lifetime {
        type uint32 {
            range "30..86400";
        }
        units seconds;
        description
            "Life time for each security association";
        reference
            "RFC 2747: RSVP Cryptographic
            Authentication";
    }
    leaf window-size {
        type uint32 {
            range "1..64";
        }
        description
            "Window-size to limit number of out-of-order
            messages.";
        reference
            "RFC 2747: RSVP Cryptographic
            Authentication";
    }
    leaf challenge {
        type empty;
        description
            "Enable challenge messages.";
        reference
            "RFC 2747: RSVP Cryptographic
            Authentication";
    }
    leaf retransmits {
        type uint32 {
```

```
        range "1..10000";
    }
    description
        "Number of retransmits when messages are
        dropped.";
    reference
        "RFC 2747: RSVP Cryptographic
        Authentication";
}
leaf key-chain {
    type key-chain:key-chain-ref;
    description
        "Key chain name to authenticate RSVP
        signaling messages.";
    reference
        "RFC 2747: RSVP Cryptographic
        Authentication";
}
}

grouping hellos-extended-config {
    description
        "Configuration parameters relating to RSVP
        hellos";
    leaf interface-based {
        type empty;
        description
            "Enable interface-based Hello adjacency if present.";
    }
    leaf hello-interval {
        type uint32;
        units milliseconds;
        description
            "Configure interval between successive Hello
            messages in milliseconds.";
        reference
            "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels.
            RFC 5495: Description of the Resource
            Reservation Protocol - Traffic-Engineered
            (RSVP-TE) Graceful Restart Procedures";
    }
    leaf hello-misses {
        type uint32 {
            range "1..10";
        }
        description
            "Configure max number of consecutive missed
            Hello messages.";
    }
}
```

```
        reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for
        LSP Tunnels RFC 5495: Description of the
        Resource Reservation Protocol - Traffic-
        Engineered (RSVP-TE) Graceful Restart
        Procedures";
    }
}

grouping signaling-parameters-extended-config {
    description
        "Configuration parameters relating to RSVP
        signaling";
    leaf refresh-interval {
        type uint32;
        description
            "Set interval between successive refreshes";
    }
    leaf refresh-misses {
        type uint32;
        description
            "Set max number of consecutive missed
            messages for state expiry";
    }
    leaf checksum {
        type boolean;
        description
            "Enable RSVP message checksum computation";
    }
    leaf patherr-state-removal {
        type empty;
        description
            "State-Removal flag in Path Error message
            if present.";
    }
}

grouping refresh-reduction-extended-config {
    description
        "Configuration parameters relating to RSVP
        refresh reduction";

    leaf bundle-message-max-size {
        type uint32 {
            range "512..65000";
        }
        description
            "Configure maximum size (bytes) of a
```

```
        single RSVP Bundle message.";
    }
    leaf reliable-ack-hold-time {
        type uint32;
        units milliseconds;
        description
            "Configure hold time in milliseconds for
            sending RSVP ACK message(s).";
    }
    leaf reliable-ack-max-size {
        type uint32;
        description
            "Configure max size of a single RSVP ACK
            message.";
    }
    leaf reliable-retransmit-time {
        type uint32;
        units milliseconds;
        description
            "Configure min delay in milliseconds to
            wait for an ACK before a retransmit.";
    }
    leaf reliable-srefresh {
        type empty;
        description
            "Configure use of reliable messaging for
            summary refresh if present.";
    }
    leaf summary-max-size {
        type uint32 {
            range "20..65000";
        }
        description
            "Configure max size (bytes) of a single
            RSVP summary refresh message.";
    }
}

grouping packets-extended-state {
    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 packet drop count";
    }

    leaf in-dropped {
        type yang:counter64;
        description
            "In packet drop count";
    }

    leaf out-errors {
        type yang:counter64;
        description
            "Out packet errors count";
    }

    leaf in-errors {
        type yang:counter64;
        description
            "In packet rx errors count";
    }
}

grouping protocol-extended-state {
    description "RSVP protocol statistics.";
}

grouping errors-extended-state {
    description
        "Error statistics.";
}

grouping extended-state {
    description "RSVP statistic attributes.";
    uses packets-extended-state;
    uses protocol-extended-state;
    uses errors-extended-state;
}

/**
 * RSVP extensions augmentations
 */
```

```
/* RSVP globals graceful restart*/
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:graceful-restart" {
    description
        "RSVP globals configuration extensions";
    uses graceful-restart-extended-config;
}

/* RSVP statistics augmentation */
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:packets" {
    description
        "RSVP packet stats extensions";
    uses packets-extended-state;
}
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:messages" {
    description
        "RSVP protocol message stats extensions";
    uses protocol-extended-state;
}
augment "/rt:routing/rt:control-plane-protocols/" +
    "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/" +
    "rsvp:statistics/rsvp:errors" {
    description
        "RSVP errors stats extensions";
    uses errors-extended-state;
}

/**
 * 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-config;
}

/* 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-config;
  }

  /* 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-config;
  }

  /* 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-config;
  }

  /**
   * RSVP 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-config;
  }

  /* 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-config;
  }

  /* 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-config;
}

/* 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-config;
}
}
<CODE ENDS>
```

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

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-extended  
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:	ietf-rsvp
reference:	RFCXXXX
name:	ietf-rsvp-extended
namespace:	urn:ietf:params:xml:ns:yang:ietf-rsvp-extended
prefix:	ietf-rsvp-extended
reference:	RFCXXXX



#### 4. 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 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:
```

The presence of this container enables the RSVP protocol functionality on a device. It also controls the configuration settings on data nodes pertaining to RSVP sessions, interfaces and neighbors. 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.

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.

## 5. Acknowledgement

The authors would like to thank Lou Berger for reviewing and providing valuable feedback on this document.

## 6. 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

## 7. Normative References

- [I-D.ietf-netconf-subscribed-notifications]  
Voit, E., Clemm, A., Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Event Notifications", draft-ietf-netconf-subscribed-notifications-26 (work in progress), May 2019.
- [I-D.ietf-netconf-yang-push]  
Clemm, A. and E. Voit, "Subscription to YANG Datastores", draft-ietf-netconf-yang-push-25 (work in progress), May 2019.

- [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-06 (work in progress), April  
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>>.
- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,  
DOI 10.17487/RFC3688, January 2004,  
<<https://www.rfc-editor.org/info/rfc3688>>.
- [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>>.

- [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>>.
- [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>>.

## Authors' Addresses

Vishnu Pavan Beeram  
Juniper Networks

Email: [vbeeram@juniper.net](mailto:vbeeram@juniper.net)

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  
Jabil

Email: [Xufeng\\_Liu@jabil.com](mailto:Xufeng_Liu@jabil.com)

Igor Bryskin  
Huawei Technologies

Email: [Igor.Bryskin@huawei.com](mailto:Igor.Bryskin@huawei.com)

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 8, 2020

V. Beeram  
T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems, Inc.  
X. Liu  
Volta Networks  
I. Bryskin  
Huawei Technologies  
H. Shah  
Ciena  
July 07, 2019

A YANG Data Model for RSVP-TE Protocol  
draft-ietf-teas-yang-rsvp-te-07

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 January 8, 2020.

#### 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
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 . . . . .	14
2.3. YANG Modules . . . . .	16
2.3.1. RSVP-TE YANG Module . . . . .	16
2.3.2. RSVP-TE MPLS YANG Module . . . . .	29
3. IANA Considerations . . . . .	42
4. Security Considerations . . . . .	42
5. Acknowledgement . . . . .	43
6. Contributors . . . . .	43
7. References . . . . .	43
7.1. Normative References . . . . .	44
7.2. Informative References . . . . .	45
Authors' Addresses . . . . .	46

#### 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.

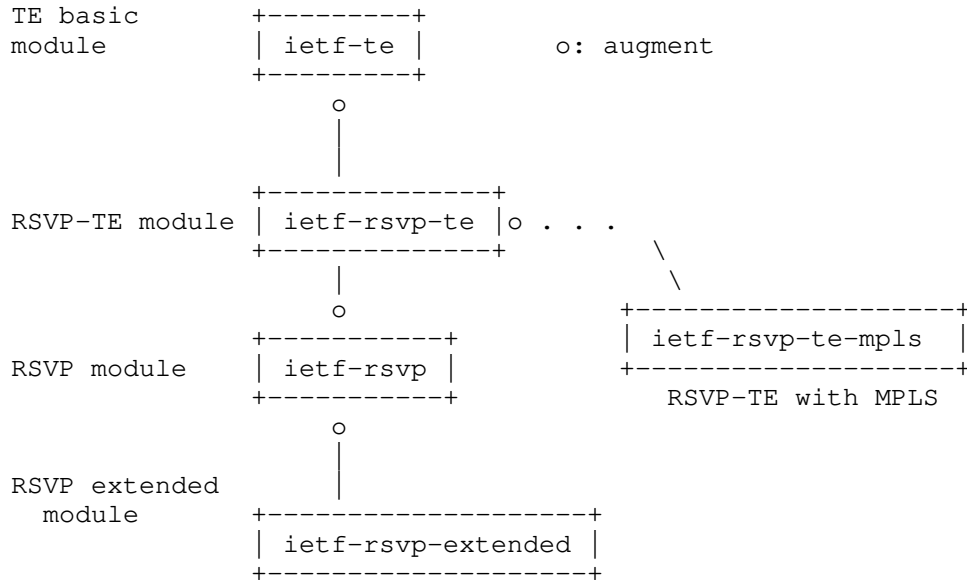


Figure 1: Relationship of RSVP and RSVP-TE modules with other protocol modules

## 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/rsvp:globals:
      +--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
      +--ro state
  augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces
      /rsvp:interface:

```

```

    +---rw rsvp-te-interface-attributes
        +---ro state
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:sessions/rsvp:session/rsvp:state/rsvp:psbs/rsvp:psb:
    +---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
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:globals
    /rsvp:sessions/rsvp:session/rsvp:state/rsvp:rsbs/rsvp:rsb:
    +---ro fspec-average-rate?    rt-types:bandwidth-ieee-float32
    +---ro fspec-size?            rt-types:bandwidth-ieee-float32
    +---ro fspec-peak-rate?       rt-types:bandwidth-ieee-float32
    +---ro min-policed-unit?      uint32
    +---ro max-packet-size?       uint32
augment /rt:routing/rt:control-plane-protocols
    /rt:control-plane-protocol/rsvp:rsvp/rsvp:neighbors:
augment /te:te/te:tunnels/te:tunnel:
    +---rw lsp-signaled-name?     string
    +---rw session-attribute*     identityref
    +---rw lsp-attribute*         identityref
    +---rw retry-timer?           uint16
augment /te:te/te:lsps-state/te:lsp:
    +---ro associated-rsvp-session?    leafref
    +---ro lsp-signaled-name?          string
    +---ro session-attribute*          identityref
    +---ro lsp-attribute*              identityref
    +---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:tunnels/te:tunnel/te:p2p-primary-paths
    /te:p2p-primary-path/te:lsps/te:lsp:
+---ro associated-rsvp-session?      leafref
+---ro lsp-signaled-name?            string
+---ro session-attribute*            identityref

```

```

+--ro lsp-attribute*                               identityref
+--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:tunnels/te:tunnel/te:p2p-primary-paths
    /te:p2p-primary-path/te:lsp-provisioning-error-infos
    /te:lsp-provisioning-error-info:
+---ro rsvp-message-type?      identityref
+---ro rsvp-error-code?        uint8
+---ro rsvp-error-subcode?     uint16
augment /te:te/te:tunnels/te:tunnel/te:p2p-primary-paths
    /te:p2p-primary-path/te:lsps/te:lsp
    /te:lsp-provisioning-error-infos
    /te:lsp-provisioning-error-info:
+---ro rsvp-message-type?      identityref
+---ro rsvp-error-code?        uint8
+---ro rsvp-error-subcode?     uint16
augment /te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths
    /te:p2p-secondary-path/te:lsps/te:lsp:
+---ro associated-rsvp-session?      leafref
+---ro lsp-signaled-name?            string
+---ro session-attribute*            identityref
+---ro lsp-attribute*                identityref
+---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:tunnels/te:tunnel/te:p2p-secondary-paths
    /te:p2p-secondary-path/te:lsp-provisioning-error-infos

```

```

        /te:lsp-provisioning-error-info:
    +--ro rsvp-message-type?    identityref
    +--ro rsvp-error-code?      uint8
    +--ro rsvp-error-subcode?   uint16
augment /te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths
        /te:p2p-secondary-path/te:lsps/te:lsp
        /te:lsp-provisioning-error-infos
        /te:lsp-provisioning-error-info:
    +--ro rsvp-message-type?    identityref
    +--ro rsvp-error-code?      uint8
    +--ro rsvp-error-subcode?   uint16
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 fast-reroute-local-revertive
      +--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:globals
    /rsvp:sessions/rsvp:session/rsvp:state:
  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-state/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:p2p-primary-paths
    /te:p2p-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:p2p-secondary-paths
    /te:p2p-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@2019-07-06.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 "draft-ietf-teas-yang-te-types: A YANG Data Model for
        Common Traffic Engineering Types";
}

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

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

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

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

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

    Editor:     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 "2019-07-06" {  
  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-config {
  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-config {
  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'";
    type identityref {
      base te-types:session-attributes-flags;
    }
  }
}
```

```
        description "RSVP session attributes flags";
        reference
            "RFC4859: Registry for RSVP-TE Session Flags";
    }
}

grouping lsp-properties-config {
    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-config;
    uses lsp-attributes-flags-config;
}

grouping tunnel-properties-config {
    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-config {
    description
        "Configuration for global RSVP-TE soft preemption";
    leaf soft-preemption-timeout {
        type uint16 {
            range 0..300;
        }
        default 0;
        description
            "Timeout value for soft preemption to revert
```

```
        to hard preemption";
    }
}

grouping global-soft-preemption {
    description
        "Top level group for RSVP-TE soft-preemption";
    container global-soft-preemption {
        presence "Enables soft preemption on a node.";
        description
            "Top level container for RSVP-TE soft-preemption";
        uses global-soft-preemption-config;
    }
}

/**** 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";
        container state {
            config false;
            description
                "State information associated with RSVP-TE
                bandwidth";
        }
    }
}

/**** End of RSVP-TE generic groupings ****/

/* RSVP-TE global properties */
augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals" {
    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:globals/"
    + "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-state;
  }

}

augment "/rt:routing/rt:control-plane-protocols/"
+ "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/"
+ "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:globals/"
+ "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:p2p-primary-paths/te:p2p-primary-path" +
        "/te:path-setup-protocol = '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-config;
    uses tunnel-properties-config;
}

/* 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";
  }
}

grouping rsvp-te-lsp-properties {
  description "RSVP-TE LSP properties grouping";
  leaf associated-rsvp-session {
    type leafref {
      path "/rt:routing/rt:control-plane-protocols/"
        + "rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/"
        + "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-config;
uses lsp-explicit-route-information-state;
uses lsp-record-route-information-state;
}

augment "/te:te/te:lsps-state/te:lsp" {
  when "/te:te/te:lsps-state/te:lsp" +
    "/te:path-setup-protocol = 'te-types:path-setup-rsvp'" {
    description
      "When the signaling protocol is RSVP-TE ";
  }
  description

```



```
    "RSVP-TE generic data augmentation pertaining to specific TE
    LSP";
    uses rsvp-te-lsp-properties;
}

augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
    "/te:p2p-primary-path/te:lsps/te:lsp" {
    when "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
        "/te:p2p-primary-path/te:lsps/te:lsp" +
        "/te:path-setup-protocol = 'te-types:path-setup-rsvp'" {
        description
            "When the signaling protocol is RSVP-TE ";
    }
    description
        "RSVP-TE generic data augmentation pertaining to specific TE
        LSP";
    uses rsvp-te-lsp-properties;
}

augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
    "/te:p2p-primary-path" +
    "/te:lsp-provisioning-error-infos" +
    "/te:lsp-provisioning-error-info" {
    description
        "Augmentation for RSVP-TE per LSP error reason";
    uses rsvp-te-lsp-error-info;
}

augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
    "/te:p2p-primary-path/te:lsps/te:lsp" +
    "/te:lsp-provisioning-error-infos" +
    "/te:lsp-provisioning-error-info" {
    when "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
        "/te:p2p-primary-path/te:lsps/te:lsp" +
        "/te:path-setup-protocol = 'te-types:path-setup-rsvp'" {
        description
            "When the signaling protocol is RSVP-TE ";
    }
    description
        "Augmentation for RSVP-TE per path error reason";
    uses rsvp-te-lsp-error-info;
}

augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
    "/te:p2p-secondary-path/te:lsps/te:lsp" {
    when "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
        "/te:p2p-secondary-path/te:lsps/te:lsp" +
        "/te:path-setup-protocol = 'te-types:path-setup-rsvp'" {
        description
            "When the signaling protocol is RSVP-TE ";
    }
```

```

    }
    description
      "RSVP-TE generic data augmentation pertaining to specific TE
      LSP";
    uses rsvp-te-lsp-properties;
  }
  augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
    "/te:p2p-secondary-path" +
    "/te:lsp-provisioning-error-infos" +
    "/te:lsp-provisioning-error-info" {
    description
      "Augmentation for RSVP-TE per path error reason";
    uses rsvp-te-lsp-error-info;
  }
  augment "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
    "/te:p2p-secondary-path/te:lsps/te:lsp" +
    "/te:lsp-provisioning-error-infos" +
    "/te:lsp-provisioning-error-info" {
    when "/te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths" +
      "/te:p2p-secondary-path/te:lsps/te:lsp" +
      "/te:path-setup-protocol = 'te-types:path-setup-rsvp'" {
    description
      "When the signaling protocol is RSVP-TE ";
    }
    description
      "Augmentation for RSVP-TE per LSP error reason";
    uses rsvp-te-lsp-error-info;
  }
}

/* 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@2019-07-06.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 "draft-ietf-teas-yang-te-types: A YANG Data Model for
              Common Traffic Engineering Types";
  }

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

  import ietf-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>

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

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

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

        Editor:     Tarek Saad
                   <mailto:tsaad.net@gmail.com>

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

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

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

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

    description
        "Latest update to MPLS RSVP-TE YANG data model.
        The model fully conforms to the Network Management Datastore
        Architecture (NMDA).

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

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
```

```
(https://trustee.ietf.org/license-info).
This version of this YANG module is part of RFC XXXX; see
the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove this
// note.

// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.

revision "2019-07-06" {
  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-config {
  description
    "Configuration parameters relating to RSVP-TE MPLS LSP
    attribute flags";
}

grouping lsp-session-attributes-obj-flags-mpls-config {
  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'";
    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-config {
  description
    "Top level grouping for LSP properties.";
  uses lsp-session-attributes-obj-flags-mpls-config;
  uses lsp-attributes-flags-mpls-config;
```

```
}

grouping lsp-properties-mpls {
  description
    "Top level grouping for LSP properties.";
  uses lsp-session-attributes-obj-flags-mpls-config;
  uses lsp-attributes-flags-mpls-config;
}
/* 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-state/te:lsp/"+
          "te:source";
      }
    }
    description
      "Tunnel sender address extracted from
      SENDER_TEMPLATE object";
  }
}
```

```
        reference "RFC3209";
    }
    leaf destination {
        type leafref {
            path "/te:te/te:lsps-state/te:lsp/"+
                "te:destination";
        }
        description
            "Tunnel endpoint address extracted from
            SESSION object";
        reference "RFC3209";
    }
    leaf tunnel-id {
        type leafref {
            path "/te:te/te:lsps-state/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-state/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-state/te:lsp/"+
                "te:extended-tunnel-id";
        }
        description
            "Extended Tunnel ID of the LSP.";
        reference "RFC3209";
    }
    leaf type {
        type leafref {
            path "/te:te/te:lsps-state/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-config {
  description
    "Interface bandwidth reservable configuration grouping";
  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.";
    }
}
}

grouping bandwidth-mpls-reservable {
    description
        "Packet reservable bandwidth";
    container bandwidth-mpls-reservable {
        description
            "Interface bandwidth reservable container";
        uses bandwidth-mpls-reservable-config;
    }
}

/* End of RSVP-TE interface groupings */

/* RSVP-TE FRR groupings */
grouping rsvp-te-frr-auto-tunnel-backup-config {
    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-config {
    description
        "Top level container for RSVP-TE FRR backup parameters";
    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-config;
  }
}
}

grouping rsvp-te-frr-backups {
  description
    "RSVP-TE facility backup grouping";
  container rsvp-te-frr-backups {
    description
      "RSVP-TE facility backup properties";
    uses rsvp-te-frr-backups-config;
  }
}

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;
    }
}

grouping fast-reroute-local-revertive-config {
    description "RSVP-TE FRR local revertive grouping";
    leaf rsvp-frr-local-revert-delay {
        type uint32;
        description
            "Time to wait after primary link is restored
            before node attempts local revertive
            procedures.";
    }
}

/**** End of RSVP-TE FRR backup information ****/

grouping fast-reroute-local-revertive {
    description
        "Top level grouping for globals properties";
    container fast-reroute-local-revertive {
        description "RSVP-TE FRR local revertive container";
        uses fast-reroute-local-revertive-config;
    }
}

/* 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 fast-reroute-local-revertive;
}
```

```
}

/* 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:globals/"
+ "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-config;
}

augment "/te:te/te:lsps-state/te:lsp" {
  when "/te:te/te:lsps-state/te:lsp" +
    "/te:path-setup-protocol = '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:p2p-primary-paths" +
  "/te:p2p-primary-path/te:lsps/te:lsp" {
  when "/te:te/te:tunnels/te:tunnel" +
    "/te:p2p-secondary-paths/te:p2p-secondary-path/" +
    "te:path-setup-protocol = '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:p2p-secondary-paths" +
  "/te:p2p-secondary-path/te:lsps/te:lsp" {
  when "/te:te/te:tunnels/te:tunnel" +
    "/te:p2p-secondary-paths/te:p2p-secondary-path/" +
    "te:path-setup-protocol = '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  
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-rsvp-te-mpls  
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-te  
prefix: ietf-rsvp  
reference: RFCXXXX

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

### 4. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC8341] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

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

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/globals: The data nodes defined defined in this

document and under this branch are applicable device-wide and can affect all RSVP established sessions. Unauthorized access to this container can potentially cause disruptive event(s) on all established sessions.

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/rsvp:globals/rsvp:sessions: The data nodes defined in this document and under this branch are applicable to one or all RSVP-TE session(s). Unauthorized access to this container can potentially affect the impacted RSVP session(s).

/rt:routing/rt:control-plane-protocols/rt:control-plane-protocol/rsvp:rsvp/rsvp:interfaces: The data nodes defined defined in this document and under this branch are applicable to one or all RSVP interfaces. Unauthorized access to this container can potentially affect established session(s) over impacted interface(s).

## 5. Acknowledgement

The authors would like to thank Lou Berger for reviewing and providing valuable feedback on this document.

## 6. Contributors

Xia Chen  
Huawei Technologies

Email: jescia.chenxia@huawei.com

Raqib Jones  
Brocade

Email: raqib@Brocade.com

Bin Wen  
Comcast

Email: Bin\_Wen@cable.comcast.com

## 7. References



## 7.1. Normative References

- [I-D.ietf-teas-yang-rsvp]  
Beeram, V., Saad, T., Gandhi, R., Liu, X., and I. Bryskin,  
"A YANG Data Model for Resource Reservation Protocol  
(RSVP)", draft-ietf-teas-yang-rsvp-11 (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-21 (work in  
progress), April 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-10 (work in progress), July 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. Ayyangarps, "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](mailto:vbeeram@juniper.net)

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)

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: May 6, 2020

X. Liu  
Volta Networks  
I. Bryskin  
Individual  
V. Beeram  
T. Saad  
Juniper Networks  
H. Shah  
Ciena  
S. Litkowski  
Cisco  
November 3, 2019

YANG Data Model for SR and SR TE Topologies  
draft-ietf-teas-yang-sr-te-topo-06

Abstract

This document defines a YANG data model for Segment Routing (SR) topology and Segment Routing (SR) traffic engineering (TE) topology.

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 May 6, 2020.

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
1.1. Terminology . . . . .	2
1.2. Tree Diagrams . . . . .	3
2. Modeling Considerations . . . . .	3
2.1. Segment Routing (SR) Topology . . . . .	3
2.2. Segment Routing (SR) TE Topology . . . . .	3
2.3. Relations to ietf-segment-routing . . . . .	4
2.4. Topology Type Modeling . . . . .	5
2.5. Topology Attributes . . . . .	5
2.6. Node Attributes . . . . .	5
2.7. Link Attributes . . . . .	6
3. Model Structure . . . . .	7
4. YANG Module . . . . .	9
5. IANA Considerations . . . . .	16
6. Security Considerations . . . . .	17
7. References . . . . .	18
7.1. Normative References . . . . .	18
7.2. Informative References . . . . .	19
Appendix A. Companion YANG Model for Non-NMDA Compliant Implementations . . . . .	21
A.1. SR Topology State Module . . . . .	21
Appendix B. Data Tree Example . . . . .	24
Appendix C. Contributors . . . . .	31
Authors' Addresses . . . . .	31

## 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

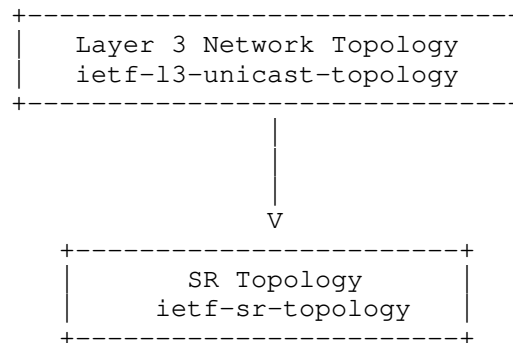
## 1.2. Tree Diagrams

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

## 2. Modeling Considerations

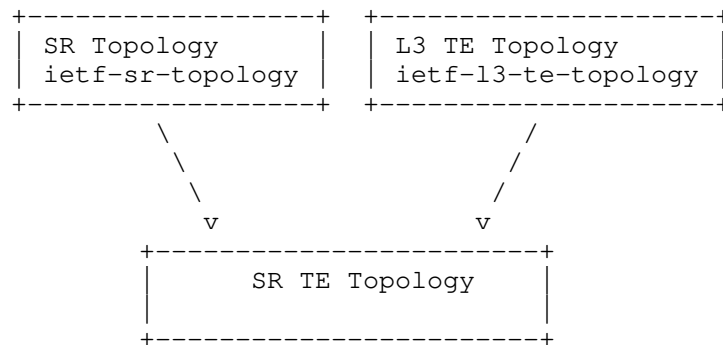
### 2.1. Segment Routing (SR) Topology

The Layer 3 network topology model is discussed in [RFC8346]. The Segment Routing (SR) topology model proposed in this document augments and uses the `ietf-l3-unicast-igp-topology` module defined in [RFC8346]. SR related attributes are covered in the `ietf-sr-topology` model.



### 2.2. Segment Routing (SR) TE Topology

When traffic engineering is enabled on an SR topology, there will be associations between objects in SR topologies and objects in TE topologies. An SR TE topology is both an SR topology and a layer 3 TE topology. Multiple inheritance is used to achieve such relations.



Each type of topologies is indicated by "network-types" defined in [RFC8345]. For the three types of topologies above, the data representations are:

L3 Topology:

```
/nd:networks/nd:network/nd:network-types/l3-unicast-topology
```

L3 TE Topology:

```
/nd:networks/nd:network/nd:network-types/l3-unicast-topology/l3-te
```

SR Topology:

```
/nd:networks/nd:network/nd:network-types/l3-unicast-topology/sr-mpls
```

SR TE Topology: (multiple inheritance)

```
/nd:networks/nd:network/nd:network-types/l3-unicast-topology/l3-te
/nd:networks/nd:network/nd:network-types/l3-unicast-topology/sr-mpls
```

### 2.3. Relations to ietf-segment-routing

[I-D.ietf-spring-sr-yang] defines ietf-segment-routing that is a model intended to be used on network elements to configure or operate segment routing; ietf-sr-topology defined in this document is intended to be used on a controller for the network-wide operations such as path computation.

SR topology model shares many modeling constructs defined in ietf-segment-routing. The module ietf-sr-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!
```

## 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. 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
        +--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.

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:

```

augment /nw:networks/nw:network/nw:node/l3t:l3-node-attributes:
  +--rw sr
    +--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
    +--ro node-capabilities
      |   +--ro transport-planes* [transport-plane]
      |   |   +--ro transport-plane    identityref
      |   +--ro entropy-readable-label-depth?    uint8
    +--rw msd?                                uint8 {msd}?
    +--ro information-source?                  enumeration
    +--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!
      +--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. The modeling structure is as follows:

```

augment /nw:networks/nw:network/nt:link/l3t:l3-link-attributes:
  +--rw sr!
    +--rw value-type?          enumeration
    +--rw sid                  uint32
    +--rw advertise-protection? enumeration
    +--rw is-local?            boolean
    +--rw msd?                 uint8 {msd}?
    +--rw address-family?      enumeration
    +--rw is-backup?            boolean
    +--rw is-part-of-set?       boolean
    +--rw is-persistent?        boolean
    +--rw is-on-lan?            boolean
    +--ro information-source?    enumeration
    +--ro information-source-state
      +--ro credibility-preference? uint16

```

The usage of the leaf "advertise-protection" is described in [I-D.ietf-spring-sr-yang].

Both IGP and BGP 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 [I-D.ietf-teas-yang-te-topo]

### 3. Model Structure

The model tree structure of the Segment Routing (SR) topology module is as shown below:

```

module: ietf-sr-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
          +--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
          +--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
+--ro node-capabilities
    |   +--ro transport-planes* [transport-plane]
    |   |   +--ro transport-plane  identityref
    |   +--ro entropy-readable-label-depth?  uint8
+--rw msd?                               uint8 {msd}?
+--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!
    +--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!
    +--rw value-type?                 enumeration
    +--rw sid                         uint32
    +--rw advertise-protection?       enumeration
    +--rw is-local?                   boolean
    +--rw msd?                        uint8 {msd}?
    +--rw address-family?             enumeration
    +--rw is-backup?                   boolean
    +--rw is-part-of-set?              boolean
    +--rw is-persistent?              boolean
    +--rw is-on-lan?                  boolean
    +--ro information-source?          enumeration
    +--ro information-source-instance?  string
    +--ro information-source-state
        +--ro credibility-preference?  uint16

```

## 4. YANG Module

```
<CODE BEGINS> file "ietf-sr-topology@2019-11-02.yang"
module ietf-sr-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sr-topology";
  prefix "srt";

  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
      "I-D.ietf-spring-sr-yang: 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.

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 2019-11-02 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for SR and SR TE Topologies";
}

feature msd {
  description
    "Support of signaling MSD (Maximum SID Depth) in IGP.";
}

grouping sr-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 topology type.";
  }
}

augment "/nw:networks/nw:network/nw:network-types/"
+ "l3t:l3-unicast-topology" {
  description
```

```
    "Defines the SR topology type.";
    uses sr-topology-type;
}

augment "/nw:networks/nw:network/l3t:l3-topology-attributes" {
    when "../nw:network-types/l3t:l3-unicast-topology/srt:sr-mpls" {
        description "Augment only for SR topology.";
    }
    description "Augment topology configuration";
    uses sr-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node/l3t:l3-node-attributes" {
    when "../nw:network-types/l3t:l3-unicast-topology/"
        + "srt:sr-mpls" {
        description "Augment only for SR 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/"
        + "srt:sr-mpls" {
        description "Augment only for SR 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/"
        + "srt:sr-mpls" {
        description "Augment only for SR topology.";
    }
    description "Augment link configuration";
    uses sr-link-attributes;
}

grouping sr-topology-attributes {
    description "SR topology scope attributes.";
    container sr {
        description
            "Containing SR attributes.";
        uses sr-cmn:srgb;
    } // sr
} // sr-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.";
      }
      enum "ospfv3" {
        description "OSPFv3.";
      }
      enum "isis" {
        description "ISIS.";
      }
      enum "bgp-ls" {
        description "BGP-LS.";
        reference
          "RFC 7752: North-Bound Distribution of Link-State and
          Traffic Engineering (TE) Information Using BGP";
      }
      enum "system-processed" {
        description "System processed entity.";
      }
      enum "other" {
        description "Other source.";
      }
    }
    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 {
        description
            "Containing SR attributes.";
        uses sr-cmn:srgb;
        uses sr-cmn:srlb;
        uses sr-cmn:node-capabilities;
        leaf msd {
            if-feature "msd";
            type uint8;
            description
                "Node MSD is the lowest MSD supported by the node.";
        }
        // 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 {
        presence "Presence indicates SR is enabled.";
        description
            "Containing SR attributes for a prefix.";
        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.";
}
} // sr
} // sr-node-prefix-attributes

grouping sr-link-attributes {
    description "SR link scope attributes";
    container sr {
        presence "Presence indicates SR is enabled.";
        description
            "Containing SR attributes.";
        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 advertise-protection {
            type enumeration {
                enum "single" {
                    description
                        "A single Adj-SID is associated
                        with the adjacency and reflects
                        the protection configuration.";
                }
                enum "dual" {
                    description
                        "Two Adj-SIDs will be associated
                        with the adjacency if interface
                        is protected. In this case
                        one will be enforced with
                        backup flag set, the other
```

```
        will be enforced to backup flag unset.
        In case, protection is not configured,
        a single Adj-SID will be advertised
        with backup flag unset.";
    }
}
default "single";
description
    "If set, the Adj-SID refers to an
    adjacency being protected.";
}
leaf is-local {
    type boolean;
    default false;
    description
        "'true' if the SID is local.";
}
leaf msd {
    if-feature "msd";
    type uint8;
    description
        "SID depth of the interface associated with the link.";
}
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-backup {
    type boolean;
    default false;
    description
        "'true' if the SID is a backup.";
}
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.";
    }
    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-topology
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
-----
```

```
-----
URI: urn:ietf:params:xml:ns:yang:ietf-sr-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-topology
namespace:     urn:ietf:params:xml:ns:yang:ietf-sr-topology
prefix:        srt
reference:     RFC XXXX
-----
```

```
-----
name:          ietf-sr-topology-state
namespace:     urn:ietf:params:xml:ns:yang:ietf-sr-topology-state
prefix:        srt-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:network-types/l3t:l3-unicast-topology/sr-mpls`

This subtree specifies the SR topology type. Modifying the configurations can make SR 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 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:network-types/l3t:l3-unicast-topology/sr-mps

Unauthorized access to this subtree can disclose the SR 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>>.
- [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>>.

## 7.2. Informative References

- [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>>.
- [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>>.
- [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-spring-sr-yang]  
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-13 (work in progress), July 2019.



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

The YANG module `ietf-sr-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-topology-state`, is defined as state model, which mirrors the module `ietf-sr-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-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 Topology State Module

```
<CODE BEGINS> file "ietf-sr-topology-state@2019-11-02.yang"
module ietf-sr-topology-state {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sr-topology-state";
  prefix "srt-s";

  import ietf-sr-topology {
    prefix "srt";
  }
  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
      "I-D.ietf-spring-sr-yang: 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, 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 2019-11-02 {
  description "Initial revision";
  reference
```

```
    "RFC XXXX: YANG Data Model for SR and SR TE Topologies";
}

augment "/nw-s:networks/nw-s:network/nw-s:network-types/"
+ "l3t-s:l3-unicast-topology" {
  description
    "Defines the SR topology type.";
  uses srt:sr-topology-type;
}

augment "/nw-s:networks/nw-s:network/"
+ "l3t-s:l3-topology-attributes" {
  when "../nw-s:network-types/l3t-s:l3-unicast-topology/"
+ "srt-s:sr-mpls" {
    description "Augment only for SR topology.";
  }
  description "Augment topology configuration";
  uses srt:sr-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/"
+ "srt-s:sr-mpls" {
    description "Augment only for SR topology.";
  }
  description "Augment node configuration.";
  uses srt: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/"
+ "srt-s:sr-mpls" {
    description "Augment only for SR topology.";
  }
  description "Augment node prefix.";
  uses srt: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/"
+ "srt-s:sr-mpls" {
    description "Augment only for SR topology.";
  }
  description "Augment link configuration";
  uses srt:sr-link-attributes;
}
```

```

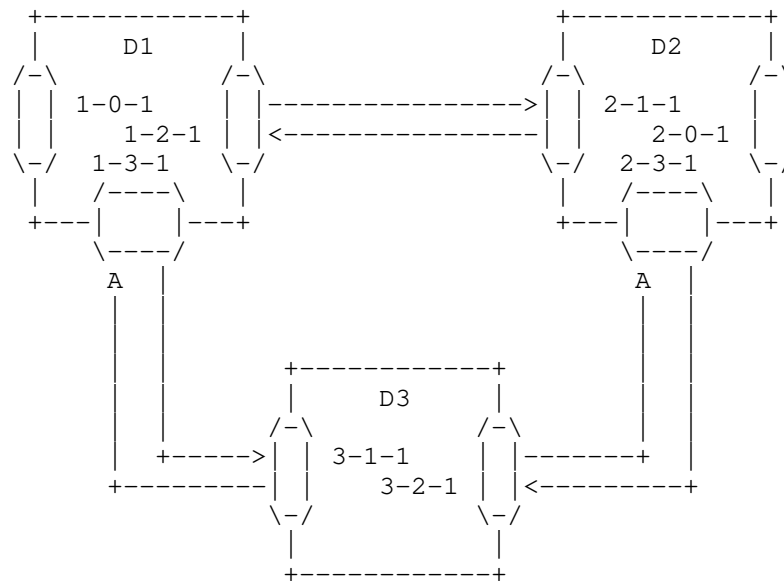
    }

    grouping sr-topology-attributes {
      description "SR topology scope attributes.";
      container sr {
        description
          "Containing SR attributes.";
        uses sr-cmn:srgb;
      } // sr
    } // sr-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-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-types": {
          "ietf-l3-unicast-topology:l3-unicast-topology": {
            "ietf-sr-topology:sr-mpls": {}
          }
        },
        "network-id": "sr-topo-example",
        "ietf-l3-unicast-topology:l3-topology-attributes": {
          "ietf-sr-topology:sr": {
            "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-topology:sr": {
            "start-sid": 101,
            "range": 1,
            "is-local": false,
            "is-node": true
        }
    },
    "ietf-sr-topology:sr": {
        "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-topology:sr": {
          "start-sid": 102,
          "range": 1,
          "is-local": false,
          "is-node": true
        }
      }
    ],
    "ietf-sr-topology:sr": {
      "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-topology:sr": {
            "start-sid": 101,
            "range": 1,
            "is-local": false,
            "is-node": true
        }
    },
    ],
    "ietf-sr-topology:sr": {
        "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": {
            "metricl": "100",
            "ietf-sr-topology:sr": {
                "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-topology:sr": {
            "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-topology:sr": {
            "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-topology:sr": {
            "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-topology:sr": {
        "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-topology:sr": {
        "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  
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

Stephane Litkowski  
Cisco  
  
EMail: slitkows.ietf@gmail.com

TEAS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 5, 2020

T. Saad  
Juniper Networks  
R. Gandhi  
Cisco Systems Inc  
X. Liu  
Volta Networks  
V. Beeram  
Juniper Networks  
I. Bryskin  
Individual  
November 02, 2019

A YANG Data Model for MPLS Traffic Engineering Tunnels  
draft-ietf-teas-yang-te-mpls-02

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 May 5, 2020.

## 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
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 . . . . .	17
4. Security Considerations . . . . .	18
5. Contributors . . . . .	18
6. Normative References . . . . .	18
Authors' Addresses . . . . .	20

## 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]

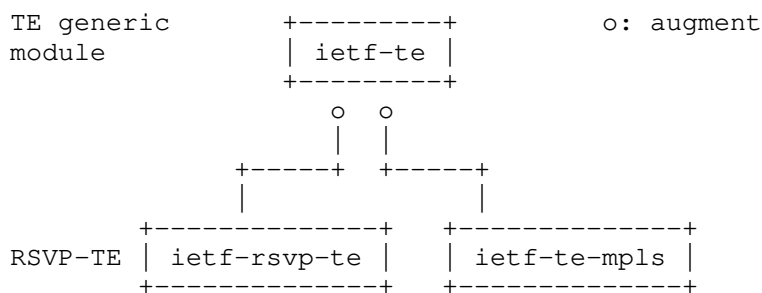


Figure 1: Relationship of MPLS TE module with TE generic and RSVP-TE YANG modules

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:p2p-primary-paths
    /te:p2p-primary-path:
    +--rw static-lsp-name?  mpls-static:static-lsp-ref
augment /te:te/te:tunnels/te:tunnel/te:p2p-secondary-paths
    /te:p2p-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:p2p-primary-paths
    /te:p2p-primary-path/te:lsps/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@2019-11-02.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 "2019-11-02" {
  description "Latest update to MPLS TE YANG module.";
  reference
    "RFCXXXX: A YANG Data Model for MPLS-TE Tunnels and LSP(s)";
}

/* 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:p2p-primary-paths/te:p2p-primary-path" {
    when "/te:te/te:tunnels/te:tunnel" +
        "/te:p2p-primary-paths/te:p2p-primary-path" +
        "/te:path-setup-protocol = 'te-types:path-setup-static'" {
        description
            "When the path is statically provisioned";
    }
}
```

```
    description "MPLS TE LSP augmentation";
    leaf static-lsp-name {
        type mpls-static:static-lsp-ref;
        description "Static LSP name";
    }
}

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

augment "/te:te/te:globals/te:named-path-constraints/" +
    "te:named-path-constraint" {
    description "foo";
    uses te-path-bandwidth_top;
}

augment "/te:te/te:tunnels/te:tunnel/te:p2p-primary-paths" +
    "/te:p2p-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-11 (work in progress),  
July 2019.
- [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-21 (work in  
progress), April 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-11 (work in progress), October 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

Vishnu Pavan Beeram  
Juniper Networks

Email: vbeeram@juniper.net

Igor Bryskin  
Individual

Email: i\_bryskin@yahoo.com



CCAMP Working Group  
Internet Draft  
Intended status: Standards Track  
Expires: May 2020

Yi Lin  
Huawei Technologies  
November 3, 2019

RSVP-TE Extensions in Support of Proactive Protection  
draft-lin-ccamp-gmpls-proactive-protection-00.txt

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), 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 May 3, 2020.

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 describes protocol-specific procedures and extensions for Generalized Multi-Protocol Label Switching (GMPLS) Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE) signaling to support Label Switched Path (LSP) Proactive Protection, which create the protection LSP after a failure is predicted and before it becomes a real failure.

## Table of Contents

1. Introduction .....	2
2. Conventions used in this document .....	3
3. Overview of Predicted Failure and Related Recovery Methods ....	3
3.1. Predicted Failure .....	3
3.2. Proactive Protection .....	4
4. Modified PROTECTION Object Format .....	5
5. Extension to ERROR_SPEC Object .....	6
5.1. New Error Code / Sub-code .....	6
5.2. New TLV in ERROR_SPEC Object .....	6
6. End-to-end Proactive Protection .....	7
6.1. Creation of the Protected LSP .....	7
6.2. Notification of Predicted Failure Event .....	7
6.3. Tearing Down of the Protection LSP .....	8
7. Proactive Segment Protection .....	8
7.1. Creation of the Protected LSP .....	8
7.2. Notification of Predicted Failure Event .....	9
7.3. Tearing Down of the Segment Recovery LSP .....	9
7.4. Priority and Resource Pre-emption .....	10
8. Consideration of Backward Compatibility .....	11
9. Security Considerations .....	11
10. IANA Considerations .....	11
11. References .....	12
11.1. Normative References .....	12
11.2. Informative References .....	12
12. Authors' Addresses .....	12

## 1. Introduction

[RFC4872] and [RFC4873] describe protocol-specific procedures and extensions for GMPLS RSVP-TE signaling to support end-to-end LSP

recovery (including protection and restoration) and segment LSP recovery, respectively.

Traditional protection solution (e.g., 1+1 or 1:1 protection) could have very fast protection switch after failure happens, but takes twice of resource in the network during the whole lifetime of the LSP. On the other hand, the traditional restoration solution has much higher resource use, but the recovery of the LSP is much slower, due to the additional signaling time to create the restoration LSP.

In order to reduce the recovery resource while keeping the very fast protection switch, an approach is to use the failure prediction technologies and to create 1+1 or 1:1 protection only when a potential failure is predicted. This approach refers to "Proactive Protection" in this document.

This document extends the RSVP-TE protocol to support the control of the Proactive Protection.

## 2. 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.

## 3. Overview of Predicted Failure and Related Recovery Methods

### 3.1. Predicted Failure

In most cases, there will be some indications before a physical failure happens in a network. For example, abnormal fluctuation of noise of a lightpath, BER (Bit Error Rate) (before error correction) rising, temperature rising of a transponder.

Therefore, by monitoring on certain physical parameters and analyzing the change tendency using, for example, Machine Learning (ML) or other technologies, a node is possible to predict whether failure will happen in an upcoming period of time.

Note that a predicted failure is different from a Signal Degrade in that:

- When Signal Degrade happens to a connection, the connection is still available but the quality of the signal carried by this

connection has declined and is lower than the predetermined threshold. For example, the BER of a connection rises and is out of tolerance.

- When a predicted failure of a connection is inferred, no failure nor degradation happens at present, but there is a trend that after a period of time, failure will probably happen, which will cause Signal Fail or Signal Degrade.

The methods to predict failures are outside the scope of this document.

### 3.2. Proactive Protection

The "Proactive Protection" refers to an LSP protection approach which create the protection LSP after a failure is predicted and before it becomes a real failure. Both end-to-end protection (defined in [RFC4872]) and segment protection (defined in [RFC4873]) are applicable for the Proactive Protection.

The main procedure of Proactive Protection is shown in Figure 1:

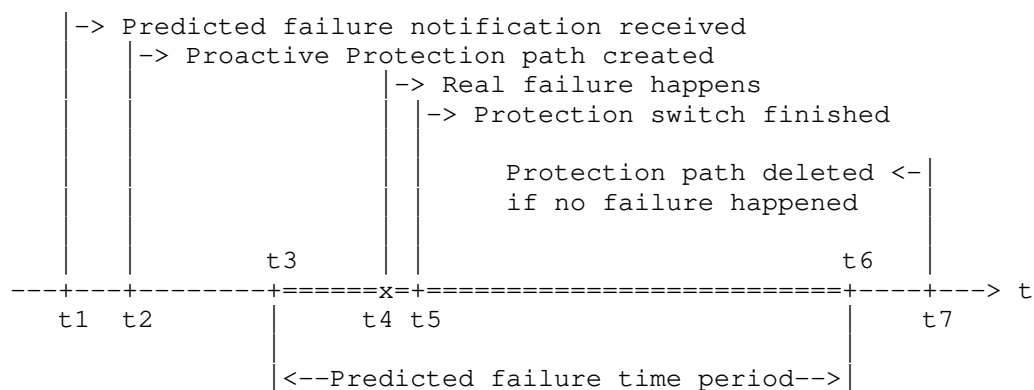


Figure 1: Overview of Proactive Protection

- t1: The protection source node of an LSP is notified that a failure will probably happen during t3~t6, so it starts to create 1+1 or 1:1 protection of the connection. Here the protection source node can be the source node of the LSP (for end-to-end protection case), or a branch node located between the source node and the predicted failure point of the LSP (for segment protection case).

t2: The 1+1 or 1:1 protecting path is created between the protection source node and the protection destination node. Here the protection destination node can be the destination node of the LSP (for end-to-end protection case), or a merge node located between the predicted failure point and the destination node of the LSP (for segment protection case).

- t4: If real failure happens as predicted, the 1+1 or 1:1 protection switch will be triggered.
- t5: Protection switch finished and the service in the connection is recovered.
- t7: If in fact the predicted failure didn't happen, and no further predicted failure notification received, the protection source node MAY tear down the protecting path after t6, in order to save the network resource.

#### 4. Modified PROTECTION Object Format

This document modifies the PROTECTION object (C-Type=2) by adding two new bits T and A in reserved fields, as shown in Figure 2 below:

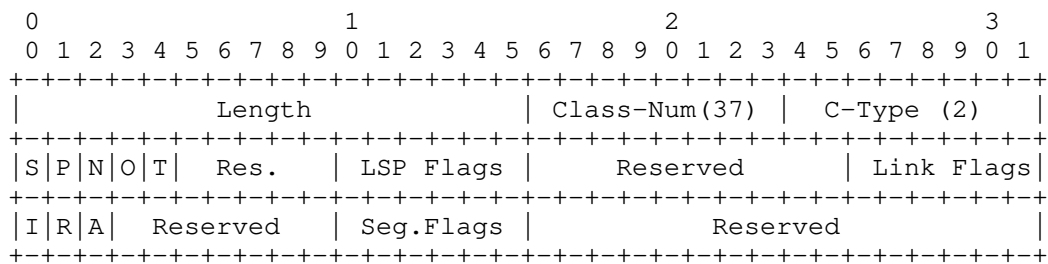


Figure 2: The modified PROTECTION object (C-Type=2)

- T (Triggered End-to-end Proactive Protection): 1 bit, when set (1), it indicates that the end-to-end Proactive Protection are required.

Note that if T bit is set (1), the LSP Flags SHOULD be one of:

0x04 1:N Protection with Extra-Traffic

0x08 1+1 Unidirectional Protection

```
0x10      1+1 Bidirectional Protection
```

- A (proActive Segment Protection): 1 bit, when set (1), it indicates that the Proactive Segment Protection are required.

Note that If A bit is set (1), the Seg. Flags SHOULD be one of:

```

0x04    1:N Protection with Extra-Traffic
0x08    1+1 Unidirectional Protection
0x10    1+1 Bidirectional Protection

```

See [RFC4872] and [RFC4873] for the definition of other fields.

## 5. Extension to ERROR\_SPEC Object

### 5.1. New Error Code / Sub-code

A new Error Sub-code under Error Code "25 - Notify Error" is defined in this document, which is used to notify the event of a predicted failure:

Error Code = 25: "Notify Error" (see [RFC3209])

Error Sub-code = TBA: "Notify Error/LSP Local Predicted Failure"

### 5.2. New TLV in ERROR\_SPEC Object

When predicting a failure, a certain time before which the failure may happen may also be predicted. This time information is useful for the source node to know how long it should wait for the predicted failure to become a real failure, and to decide when it's safe to tear down the protection LSP if the predicted failure didn't happen.

A new TLV in IPv4/IPv6 IF\_ID ERROR\_SPEC Object is defined in this document, which is used to indicate the time before which the predicted failure will probably become real failure. The format of this new TLV is shown in Figure 3 below:

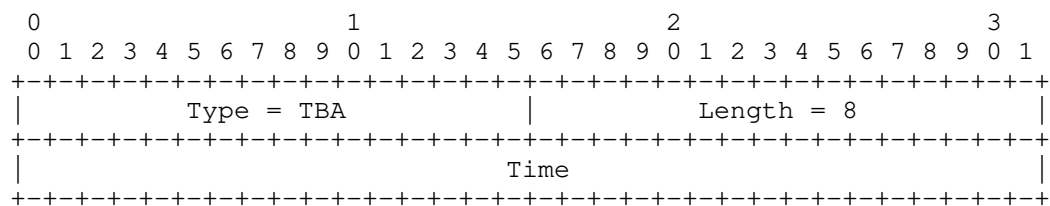


Figure 3: New TLV (type=TBA) in ERROR\_SPEC Object

- Type: TBA
- Length: 8

- Time: A relative time measured in second, which indicates within how many seconds (from the current time) the predicted failure will probably become real failure.

## 6. End-to-end Proactive Protection

### 6.1. Creation of the Protected LSP

To create an LSP with recovery type of "End-to-end Proactive Protection", the source node of the LSP generates a Path message with a PROTECTION object included. The T bit in the PROTECTION object MUST be set to 1 (End-to-end Proactive Protection), so that all other nodes along the LSP can start the failure prediction function on related links/nodes.

Note that the N bit in the PROTECTION object is used to indicate whether the control plane message exchange is only used for notification or for protection-switching purpose after real failure happens, see [RFC4872]. In other words, the N bit have nothing to do with the notification of a predicted failure before real failure happens.

To allow the notification of predicted failure event to the source node by the Notify message, the NOTIFY REQUEST object MUST also be included in the Path message (see [RFC3473]), where the "Notify Node Address" SHOULD be the address of the source node of the LSP.

### 6.2. Notification of Predicted Failure Event

When an intermediate node on an LSP infers that a failure will happen and will affect the LSP, a Notify message will be sent to the source node of the LSP, to inform such predicted failure event. A new error code/sub-code "Notify Error/LSP Local Predicted Failure" is used in the ERROR\_SPEC object or IF\_ID\_ERROR\_SPEC object in the Notify message.

The Notify message MAY also include a TLV (type = TBA) in the IPv4 or IPv6 IF\_ID\_ERROR\_SPEC object, to indicate the time before which the predicted failure will probably become real failure.

On receiving the Notify message with error code/sub-code "Notify Error/LSP Local Predicted Failure", the source node of the LSP SHOULD trigger the procedure to create the protection LSP, according to the protection type indicated in the "LSP Flags" field of the PROTECTION object in the Path message for the protected LSP. The procedures of creating the protection LSP and the protection switching after real failure happens are described in [RFC4872].

### 6.3. Tearing Down of the Protection LSP

After the protected LSP is created, the source node MAY start a timer `T_wait` and wait for the predicted failure to become a real failure. If no real failure happens and no more notification of predicted failure is received till `T_wait`, the source node MAY trigger the procedure to tear down the protection LSP, according to local policy. See [RFC4872] about the process of tearing down a protection LSP.

Implementations SHOULD allow this policy to be configured to provide a default across all LSPs on a node, but SHOULD also allow it to be configured per LSP.

Note that the `T_wait` MUST longer than the time indicated in the TLV (type=TBA) in the `ERROR_SPEC` object in the Notify message, if the TLV exists.

Note also that the value of `T_wait` is a local matter of the source node, and is outside the scope of this document.

## 7. Proactive Segment Protection

### 7.1. Creation of the Protected LSP

To create an LSP with recovery type of "Proactive Segment Protection", the source node of the LSP generates a Path message, where:

- A `PROTECTION` object is included, where the `A` bit MUST be set to 1 (Proactive Segment Protection), so that all nodes along the protected LSP can start the failure prediction function on related links/nodes if supported. The "Seg. Flags" are used to indicate the protection type of the Proactive Segment Protection.
- One or more `SERO` objects MAY included (i.e., explicit Proactive Segment Protection), indicating the branch node and the merge node of each segment recovery LSP. If no `SERO` object is included, it indicates that the dynamic Proactive Segment Protection method is used.
- A `NOTIFY REQUEST` object is included, where the `Notify Node Address` SHOULD be the address of the source node of the LSP.

For explicit Proactive Segment Protection, when a branch node receives a Path message with `A` bit set to 1 in the `PROTECTION` object, the branch node follows [RFC4873] to process the Path



message, except that the Path message for the recovery LSP will not be generated and be sent at this stage. Also, one more NOTIFY REQUEST object SHOULD be added to the Path message of the protected LSP, which carries the address of this branch node.

For dynamic Proactive Segment Protection, when an intermediate node receives a Path message with A bit set to 1 in the PROTECTION object, the node will determine if it has the ability to be a branch node, as described in Section 6.2 of [RFC4873]. If yes, it follows the same procedure as what a branch node does in the case of explicit Proactive Segment Protection, as described above. If not, the node only follows the standard procedure to create the protected LSP.

## 7.2. Notification of Predicted Failure Event

When an intermediate node between a pair of branch and merge nodes on an LSP infers that a failure will happen and will affect the LSP, a Notify message will be sent to the nearest branch node on the upstream direction of the LSP, to inform such predicted failure event. The error code/sub-code "Notify Error/LSP Local Predicted Failure" is used in the ERROR\_SPEC object or IF\_ID\_ERROR\_SPEC object in the Notify message.

Similar to End-to-end Proactive Protection, the time before which the predicted failure may occur MAY also be included in the Notify message.

On receiving the Notify message with error code/sub-code "Notify Error/LSP Local Predicted Failure", the branch node on the protected LSP SHOULD generate a new Path message, and send this new Path message along the recovery LSP between the branch and the merge nodes. The procedures of generating new Path message and creating the recovery LSP are the same as what is described in [RFC4873], except that the A bit in the PROTECTION object of this new Path message MUST set to 1.

## 7.3. Tearing Down of the Segment Recovery LSP

After the segment recovery LSP is created, the branch node MAY start a timer T\_wait and wait for the predicted failure to become a real failure. If no real failure happen and no more notification of predicted failure is received till T\_wait, the branch node MAY trigger the procedure to tear down the segment recovery LSP, according to local policy. See [RFC4873] about the process of tearing down a segment recovery LSP.

Implementations SHOULD allow this policy to be configured to provide a default across all LSPs on a node, but SHOULD also allow it to be configured per LSP.

Note that the T\_wait MUST longer than the time indicated in the TLV (type=TBA) in the ERROR\_SPEC object in the Notify message, if the TLV exists.

Note also that the value of T\_wait is a local matter of the branch node, and is outside the scope of this document.

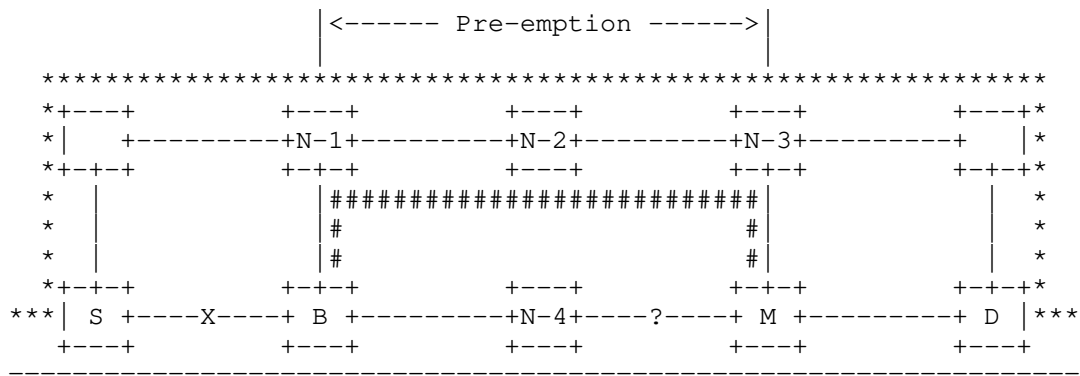
#### 7.4. Priority and Resource Pre-emption

It's possible that after recovery LSP is created and before the predicted failure becomes a real failure, another real failure happens on the LSP outside the protected segment. In this case, the source node (or an intermediate node in the upstream direction of the real failure) may start a restoration procedure to recover the LSP. For the same protected LSP, since recovering from a real failure always has higher priority than protecting against a predicted failure which still hasn't happened, the restoration LSP can pre-empt the resource of the segment recovery LSP.

As shown in Figure 4, assume that node B (branch node) was notified of a predicted failure event between N-4 and M (merge node), and has created the segment recovery LSP along B, N-1, N-2, N-3 and M. If another failure between S (source node) and B happens before the predicted failure becomes a real failure, node S will try to create the restoration LSP. Since that resource is limited, the restoration LSP can pre-empt the resource of the segment recovery LSP between N-1 and N-3.

The nodes along the segment recovery LSP has enough information to determine whether pre-emption is allowed. This is because these nodes know that:

- The current segment recovery LSP is used for Proactive Segment Protection through the A bit in the PROTECTION object;
- The segment recovery LSP and the restoration LSP are protecting the same LSP through the association relationship.



S: Source node            D: Destination node  
 B: Branch node           M: Merge node  
 X: Real failure        ?: Predicted failure (haven't happened yet)

=====: Protected LSP  
 #####: Segment Recovery LSP  
 \*\*\*\*\*: Restoration LSP

Figure 4: Resource pre-emption by restoration LSP

## 8. Consideration of Backward Compatibility

TBD.

[Editor's note]: will add some description about interwork with legacy nodes which do not support the function of failure prediction and reporting.

## 9. Security Considerations

TBD.

## 10. IANA Considerations

IANA assigns values to RSVP protocol parameters. Within the current document, a new Error code/sub-code value is defined:

Error Code = 25: "Notify Error" (see [RFC3209])

- o "Notify Error/LSP Local Predicted Failure" (TBA)

## 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.
- [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.
- [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.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017.

### 11.2. Informative References

- [RFC4426] Lang, J., Ed., Rajagopalan, B., Ed., and D. Papadimitriou, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification," RFC 4426, March 2006.

## 12. Authors' Addresses

Yi Lin  
Huawei Technologies  
F3 R&D Center, Huawei Industrial Base,  
Bantian, Longgang District,  
Shenzhen 518129 P.R.China  
Email: yi.lin@huawei.com



Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

J. Dang  
B. Liu  
Huawei Technologies  
G. Yang  
China Telecom  
K. Lee  
LG U+  
November 4, 2019

Instant Congestion Assessment Network (iCAN) for Traffic Engineering  
draft-liu-ican-01

Abstract

This draft proposes a new technology named iCAN (instant Congestion Assessment Network), which represents a set of mechanisms running directly on network nodes. These mechanisms allow the nodes adjusting the flows' paths based on real-time measurement of the candidate paths. The measurement is to reflect the congestion situation of each path, so that the nodes could decide which flows need to be switched from a path to another.

This is something that current TE technologies can hardly achieve. In current TE, the paths are usually planned in a centralized controller, which is far from the data plane, thus neither be able to assess the real-time congestion situation of each path, nor able to assure the data plane always go as expected (especially in SRv6 scenarios). In a result, traditional TE is not able to adjust the flow paths in real-time to fit for the change of traffic instantly.

iCAN can work with traditional TE perfectly: the controller plans multi-path transmission in relatively long period (e.g. minutes), and iCAN does the flow path optimization in a much shorter interval (e.g. milliseconds).

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 May 7, 2020.

#### 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 . . . . .	3
2. Problem Statement . . . . .	4
2.1. Background Problems . . . . .	4
2.1.1. Latency issue . . . . .	4
2.1.2. Microburst issue . . . . .	4
2.2. Gap Analysis . . . . .	4
2.2.1. Load balancing . . . . .	4
2.2.2. SLA assurance . . . . .	5
2.2.3. High availability . . . . .	5
3. iCAN Architecture and Key Technical Requirements . . . . .	5
3.1. Architecture . . . . .	5
3.2. Key technical requirements . . . . .	7
3.2.1. Path quality assessment . . . . .	7
3.2.2. Recognition and statistic of flows in devices . . . . .	9
3.2.3. Flow switching between paths . . . . .	9
4. Use Cases and Scenarios . . . . .	10
4.1. Network load balancing . . . . .	10
4.1.1. Multiple-access in backbone networks . . . . .	10
4.1.2. Multiple paths in metro network . . . . .	11
4.2. SLA assurance . . . . .	11
4.3. Fine-Granularity reliability . . . . .	11
5. Implementation with Underlay Technologies . . . . .	11
5.1. iCAN with SRv6 . . . . .	11
5.2. iCAN with VxLAN . . . . .	12
5.3. iCAN with MPLS/MPLS-TE . . . . .	12
6. Standardization Requirements . . . . .	12

7. Security Considerations . . . . .	12
8. IANA Considerations . . . . .	12
9. Acknowledgements . . . . .	12
10. References . . . . .	13
10.1. Normative References . . . . .	13
10.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

Traditional IP routing is shortest path based on static metrics, which can fulfil basic requirement of connectivity. MPLS-TE brings the capability of utilizing non-shortest paths, thus traffic dispatch is doable; however, MPLS-TE is only a complementary mechanism because of the scalability issue. Segment routing provides even more flexibility that paths could be easily programmed; and along with the controller, it could be scaled.

However, the above mentioned mechanisms all run in the control plane, which implies that they are not able to sense the data plane situation in real-time, thus they are mostly for relative static planning/controlling (minutes, hours or even day-level) of network traffic and not able to adapt to the microscopic traffic change in real-time (e.g. mili-second level). So, in real bearer networks (metro, backbones etc.), it is always underload so that the redundant resources could tolerant the traffic burst, results in a significant waste of network resources.

This draft proposes the iCAN (Instant Congestion Assessment Network) architecture to achieve autonomous adapt to traffic changes in real-time in terms of switching flows between multiple forwarding paths. iCAN includes following mechanisms:

- o A mechanism between ingress and egress nodes to assess the path congestion situation in RTT level speed, to recognize which paths are underload and which are heavy loaded.
- o Recognizing big flows and small flows in the device, in real time
- o Ingress node dispatches flows to multiple paths, to make load balance, or to guarantee SLA for specific flows

Use cases, scenarios and implementation candidates of iCAN are also discussed in this draft.



## 2. Problem Statement

### 2.1. Background Problems

#### 2.1.1. Latency issue

New services like AR/VR would require strictly low latency which would be a big challenge to current network. Other than fixed latency caused by the forwarding devices' internal processing and transmission time on wire, the prominent factor of latency is the queuing time caused by congestion. Thus, to control the latency of a certain path, is mostly to control the congestion.

#### 2.1.2. Microburst issue

The network users/services are so comprehensive that the traffic model is always uncertain, which results in high bandwidth peak-to-average ratio. In other words, real traffic could often change dramatically in second or even millisecond level. Thus, even if the bandwidth of paths seem all good in a network management system, there might be congestion happening in real forwarding plane that just could not be detected by the management system.

### 2.2. Gap Analysis

#### 2.2.1. Load balancing

In real networks, the traffic is usually un-balanced. Some links might be idle while some might be heavily congested. The partial congestion in the network has affected the bandwidth planning of the network. Ideally, the bandwidth planning of the network generally depends on the average bandwidth of the link, however, in reality the bandwidth planning more tends to peak bandwidth of the link in order to guarantee the business experience. Especially for low-latency service, since it is very sensitive to congestion, the result is that operators would have to increase investment for network expansion.

Although there are mechanisms against load balancing issue, the real result is usually not as expected.

##### - Device-level Load Balance (e.g. ECMP)

1) Not recognize flows' amount. ECMP is mostly deployed in a per-flow manner. Since the devices cannot recognize each flow's amount, they just fork the flows based on the numbers of flows, not the exact amount of flows. Thus, the result of ECMP could be unbalanced.

2) Not consider congestion status of E2E paths. ECMP only cares about the next hop, thus, if one remote link that is on the path is already highly congested, the device would still fork flows to the path due to the ECMP local decision.

- Network-level load balance (e.g. UCMP)

1) Lack of data plane mechanisms to ensure the real sharing ratio between multiple paths. Again, since there is no mechanisms to recognize flows' amount, the controller just could not make sure whether the traffic is forked exactly as it expected.

2) Slow reaction: The global path optimization architecture of SDN can sense traffic changes by measuring protocol, but the overall feedback speed is relatively slow.

#### 2.2.2. SLA assurance

- QoS mechanism

When congestion occurs, QoS scheduling will be triggered. The QoS mechanism in a network element selects a portion of the packets into the buffer. The purpose of delaying the retransmission is to increase the network capacity. Even if low-latency services are placed in high-priority queues to avoid additional delays, high-priority queues are still in buffers. Once there are more high-priority services, the packets will still enter the buffer for a period of time.

#### 2.2.3. High availability

Current networks highly rely on BFD for high availability. The problems of BFD are:

1) Complex configurations

2) Can only detect path on/off, not able to detect path quality deterioration

3) Cannot distinguish individual paths in multi paths

### 3. iCAN Architecture and Key Technical Requirements

#### 3.1. Architecture

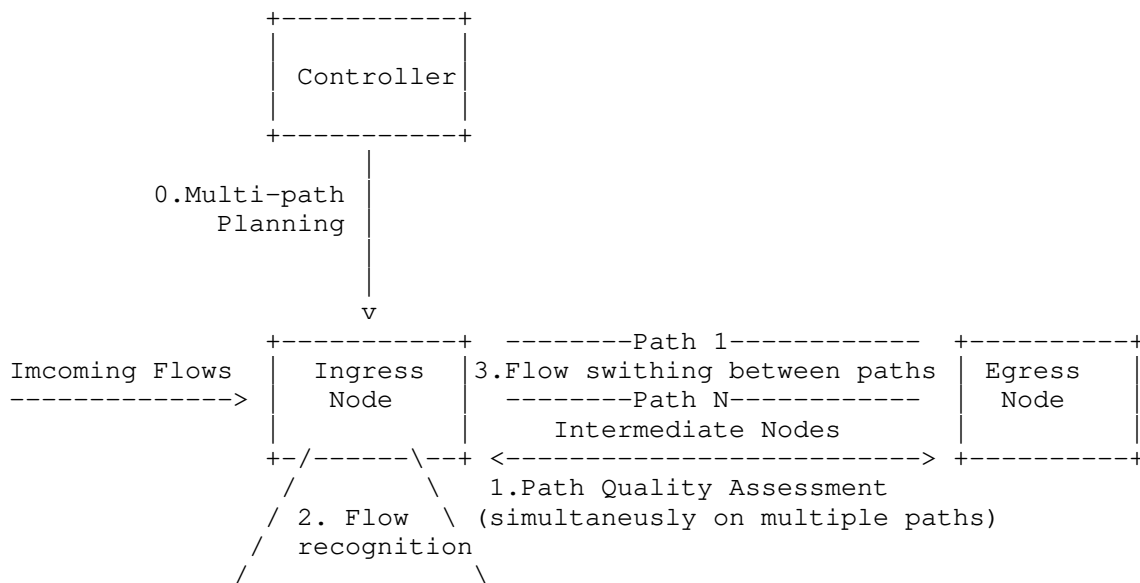


Figure-1: iCAN architecture

As above figure shows, there are 3 entities:

1. Controller

- Responsible for planning multiple paths for a set of flows that could be aggregated to a pair of Ingress/Egress routers.
- After delivering the planned paths to the ingress router, the controller would need nothing to do.

2. Ingress node:

- Serves as a local "controller" for the iCAN system.
- Responsible for triggering the path congestion assessment, which is coordinated with the egress router through a measurement protocol.
- After getting the assessment results, the ingress router would calculate which flows need to be switched to a different path, in order to make the paths load balanced or to assure the transport quality of a certain of important flows.

- In order to do the path switching calculation, the ingress router needs to recognize the TopN flow passing by it, since switching the big flows would make the most effort.

3. Egress node:

- Only needs to coordinate with the ingress router to do the path assessment.

4. Intermediate nodes (optional)

- If the intermediate nodes are allowed to participate in iCAN, they can provide useful information (e.g. link utilization) for better measurement of path quality. (TBD)

3.2. Key technical requirements

3.2.1. Path quality assessment

- o Req-1: the assessment MUST reflex the congestion status of the paths.
- o Req-2: the assessment SHOULD be done within a RTT timeslot. Since iCAN is to adapt the traffic change in real-time, the assessment needs to be done very fast.
- o Req-3: the assessment MUST be done for multiple paths between the same ingress/egress routes simultaneously.

Candidate solutions:

- o For Req-1:

In the draft [I-D.dang-ippm-congestion], a new metric is proposed to indicate the congestion degree of a certain path. A specific value could be calculated according to a certain method (described below), so that the congestion situation of different paths could be quantified and compared.

- o For Req-2 & 3:

In the draft [I-D.dang-ippm-multiple-path-measurement], a specific method is proposed to calculate the congestion degree as described above. The proposed method supports measuring multiple paths simultaneously, which needs a special mechanism to align measurements of different paths to the same time slots.

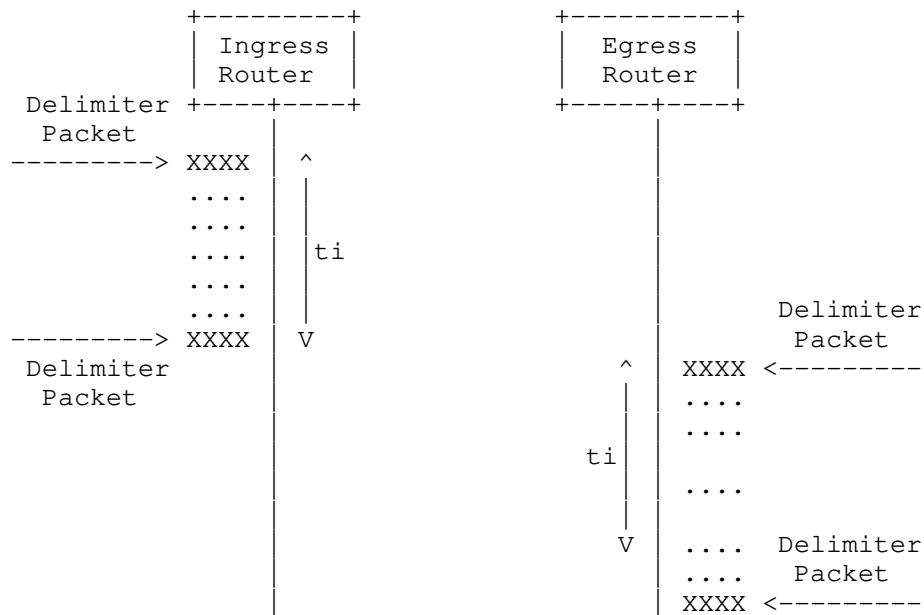


Figure-2 Path congestion assesment between ingress/egress routes

As the figure shows above, the ingress router inserts the delimiter packet in a fixed interval "ti" (e.g. 3.3ms); at the same time, the ingress router would count how many packets are sent during the interval. When the egress routers recieves the same delimiter packet, it also starts count the packet number during the same interval "ti". Because of the path congestion, the gap between the receiving packets might probably differs from the gap between sending packets, the difference is just the key infomation for estimating the congestion degree of the path. The egress router would ignore the gap difference, and just return the packet number to the ingress router when the "ti" time is up.

The ingress/egress router could also count the packets number between two delimiter packes, so that the counts could be compared for detecting packet loss. If packet loss happens, it should be specially taken into consideration by the path congestion degree calculation.

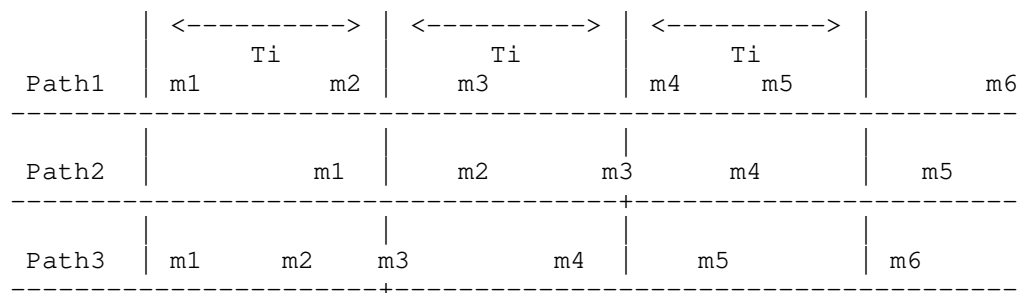


Figure-3 Multiple path measurement allignment

As the figure shows above, m1-mN indicates every measurent made on each path, as every "ti" interval, the egress would return the packet count to the ingress router for path congestion degree measurement (as described above). Due to path congestion, the packets sent by the egress router would arrive at the ingress router at different intervals other than "ti" time. Thus, for path congestion degree comparison between multipla paths, it needs a longer "Ti" interval (e.g. 10ms) to make sure that there would be at least one measurement completed most of the time.

### 3.2.2. Recognition and statistic of flows in devices

- o Req-1: the device SHOULD be able to recognize TopN big flows within a timeslot.
- o Req-2: the device MAY need to statistic all flows' amount within a timeslot.

### 3.2.3. Flow switching between paths

- o Req-1: the device SHOULD be able to recognize flow let. The flow switching is done from the next flow let.
- o Req-2: the device MAY need to actively generate gap to artificially create flow let. If the flow needs to be switched immediately, then the device would need to make the gap, to avoid out-of-order packets arriving to the destination through multiple paths.
- o Req-3: the device SHOULD avoid oscillation of frequently switching flows from one to another.
- o Req-4: multiple ingress devices SHOULD be able to coordinate so that they won't switch flows to the shared path at the same time, to avoid potential congestion in the shared path.

## 4. Use Cases and Scenarios

### 4.1. Network load balancing

Background problem: traffic is not balanced in current metro network.

While some links are heavily loaded, others might be still lightly loaded: unbalance could lows down the service quality (e.g. SLA could not be guaranteed in the heavily loaded links/paths); unbalance could lows down the network utilization ratio (normally with 30%, e.g. a 100G physical capacity network can only bear at most 30G traffic, a huge waste of network infrastructure).

iCAN could be used for load balance among the multiple paths between a pair of ingress/egress nodes. Once the network is balanced, the real throughput of the network could be elevated significantly. A couple of targeted scenarios are described below.

#### 4.1.1. Multiple-access in backbone networks

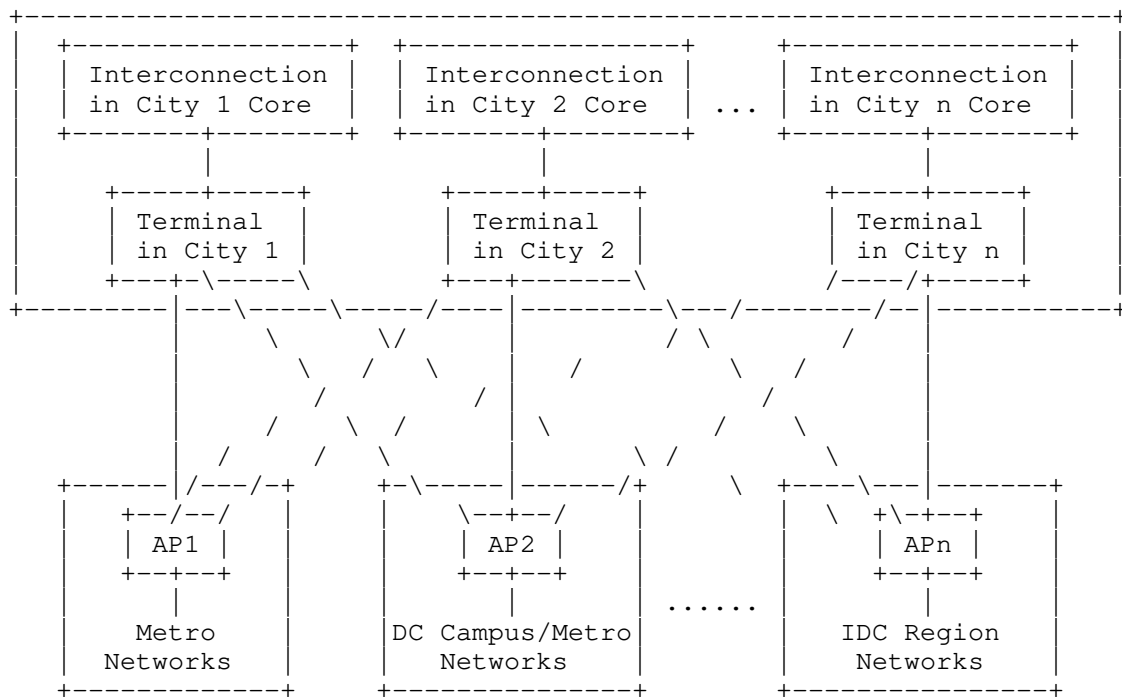


Figure-4: Multiple-access in backbone networks

As Figure-4 shows above, at the edge of the backbone, there are APs (Access Points) that attached with different networks. The APs connect to the TPs (Terminal Points), via which the traffic from the APs could be exchanged to other networks (e.g. other ISPs). In order to get a high quality connection, each AP could attach to multiple TPs, thus, the IP tunnels are constructed in a mesh manner.

Between APs and TPs, BGP is running for traffic routing. The BGP policies are always static, since it is very error-prone for manually adjusting BGP policies. Thus, the traffic running among the paths between APs and TPs are always un-balanced. With iCAN runs as on top of the mesh tunnels, the un-balanced issue could be solved in principle.

#### 4.1.2. Multiple paths in metro network

Similar as the iCAN architecture showed as Figure-1, in metro networks, there might be multiple paths between ingress/egress router pairs. Thus, iCAN could be used to increase the throughput without hardware expansion.

#### 4.2. SLA assurance

Since iCAN could switch flow in real-time, it can guarantee a set of important flows. Once the path which carries the important flows is to be congested, the other flows could be switched to alternative paths, and the important flows would stably running in the original path.

(More content TBD)

#### 4.3. Fine-Granularity reliability

Traditional reliability protocols such as BFD, can only assess the link on or off. With the path congestion assessment ability, iCAN could also assess the quality degradation.

(More content TBD)

### 5. Implementation with Underlay Technologies

#### 5.1. iCAN with SRv6

- SR Multiple Explicit Paths

For example, there are 3 paths between the ingress and egress nodes, and the multi-path is defined as a SR-List containing LSP1/2/3.



The probe message detects the congestion status of the three SR-list paths. The edge device adjusts the load balancing between the three paths according to the congestion status of the three SR-lists, and switch the flows from the path with a high congestion to the path with a low congestion.

- SR Multiple Explicit+Loose Paths

In loose path scenario, there needs to be an additional approach to probe the specific paths of a SR tunnel. After that, operations on the probed paths are the same as explicit path scenario.

5.2. iCAN with VxLAN

TBD.

5.3. iCAN with MPLS/MPLS-TE

TBD.

6. Standardization Requirements

1. Multi-path Planning (North Interface between Controller and devices)
2. Path Congestion Assesment (Horizontal Interface between devices), mostly regarding to Req-1&2&3 described in Section 3.2.1 .
3. Flow Switching Negotiation (Horizontal Interface between devices), mostly regarding to Req-3&4 described in Section 3.2.3 .

(More content TBD.)

7. Security Considerations

TBD.

8. IANA Considerations

TBD.

9. Acknowledgements

Very valuable comments were from Shunsuke Homma, Mikael Abrahamsson and Bruno Decraene.

A commercial router hardware based prototype had been implemented to prove the mechanisms discussed in the document are workable.

Conventions: [RFC2119][RFC2629]

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <<https://www.rfc-editor.org/info/rfc2629>>.

### 10.2. Informative References

- [I-D.dang-ippm-congestion]  
Dang, J. and J. Wang, "A One-Path Congestion Metric for IPPM", draft-dang-ippm-congestion-01 (work in progress), March 2019.
- [I-D.dang-ippm-multiple-path-measurement]  
Dang, J. and J. Wang, "A Multi-Path Concurrent Measurement Protocol for IPPM", draft-dang-ippm-multiple-path-measurement-01 (work in progress), March 2019.

## Authors' Addresses

Joanna Dang  
Huawei Technologies  
Q27, Huawei Campus  
No.156 Beiqing Road  
Beijing 100095  
P.R. China

Email: [dangjuanna@huawei.com](mailto:dangjuanna@huawei.com)

Bing Liu  
Huawei Technologies  
Q27, Huawei Campus  
No.156 Beiqing Road  
Hai-Dian District, Beijing 100095  
P.R. China

Email: leo.liubing@huawei.com

Guangming Yang  
China Telecom  
109 Zhongshan W Ave  
Guangzhou 510630  
P.R. China

Email: yangguangm@chinatelecom.cn

Kyungtae Lee  
LG U+  
71, Magokjungang 8-ro, Gangseo-gu  
Seoul  
Republic of Korea

Email: coolee@lguplus.co.kr

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

X. Liu  
Volta Networks  
J. Tantsura  
Apstra Networks  
I. Bryskin  
Individual  
L. Contreras  
Telefonica  
Q. Wu  
Huawei  
November 4, 2019

Transport Network Slice YANG Data Model  
draft-liu-teas-transport-network-slice-yang-00

Abstract

This document describes a YANG data model for managing and controlling transport network slices.

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 May 7, 2020.

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
1.1. Terminology . . . . .	3
1.2. Tree Diagrams . . . . .	3
2. Modeling Considerations . . . . .	3
2.1. Relationships to Related Topology Models . . . . .	3
2.2. Network Slice with TE . . . . .	4
2.3. ACTN for Network Slicing . . . . .	5
3. Model Applicability . . . . .	5
3.1. Network Slicing by Virtualization . . . . .	5
3.2. Network Slicing by TE Overlay . . . . .	7
4. Model Tree Structure . . . . .	9
5. YANG Module . . . . .	9
6. IANA Considerations . . . . .	15
7. Security Considerations . . . . .	16
8. Acknowledgements . . . . .	17
9. References . . . . .	17
9.1. Normative References . . . . .	17
9.2. Informative References . . . . .	19
Appendix A. Data Tree for the Example in Section 3.1. . . . .	21
A.1. Native Topology . . . . .	21
A.2. Network Slice Blue . . . . .	25
Authors' Addresses . . . . .	31

## 1. Introduction

This document defines a YANG [RFC7950] data model for representing, managing, and controlling transport network slices.

The defined data model is an interface between clients and providers for configurations and state retrievals, so as to support transport network slicing as a service. Through this model, a client can learn the slicing capabilities and the available resources of the provider. A client can request or negotiate with a transport network slicing provider to create an instance. The client can incrementally update its requirements on individual topology elements in the slice instance, and retrieve the operational states of these elements. With the help of other mechanisms and data models defined in IETF, the telemetry information can be published to the client.

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

### 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

A transport network slice is modeled as network topology defined in [RFC8345], with augmentations. A new network type "network-slice" is defined in this document. When a network topology data instance contains the network-slice network type, it represents an instance of a transport network slice.

### 2.1. Relationships to Related Topology Models

There are several related YANG data models that have been defined in IETF. Some of these are:

Network Topology Model:  
Defined in [RFC8345].

OTN Topology Model:  
Defined in [I-D.ietf-ccamp-otn-topo-yang].

L2 Topology Model:  
Defined in [I-D.ietf-i2rs-yang-l2-network-topology].

L3 Topology Model:  
Defined in [RFC8346].

TE Topology Model:

Defined in [I-D.ietf-teas-yang-te-topo].

Figure 1 shows the relationships among these models. The box of dotted lines denotes the model defined in this document.

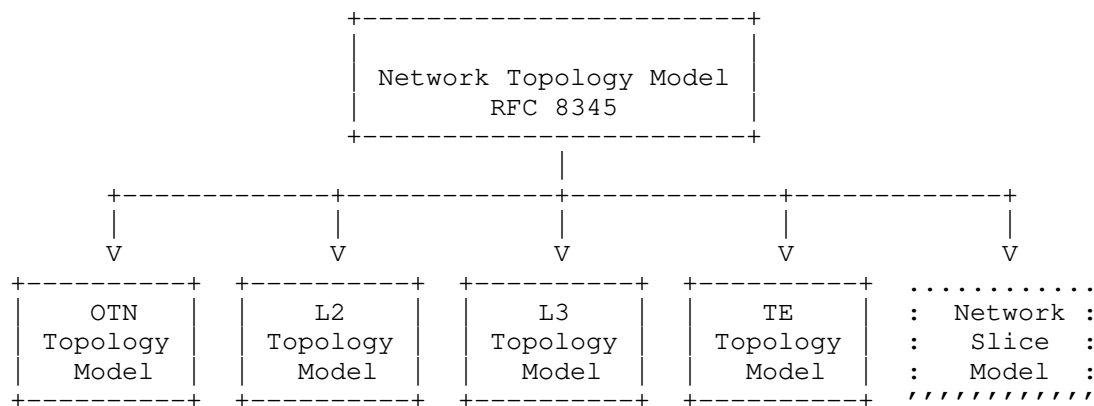


Figure 1: Model Relationships

## 2.2. Network Slice with TE

In many situations, a transport network slice needs to have TE (Traffic Engineering) capabilities to achieve certain network characteristics. The TE Topology Model defined in [I-D.ietf-teas-yang-te-topo] can be used to make a transport network slice TE capable. To achieve this, a transport network slice instance will be configured to have both "network-slice" and "te-topology" network types, taking advantage of the multiple inheritance capability featured by the network topology model [RFC8345]. The following diagram shows their relations.

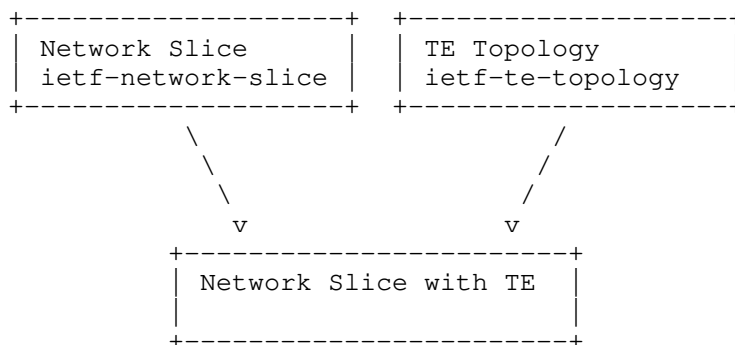


Figure 2: Network Slice with TE

This method can be applied to other types of network topology models too. For example, when a network topology instance is configured to have the types of "network-slice" defined in this document, "te-topology" defined in [I-D.ietf-teas-yang-te-topo], and "l3-unicast-topology" defined in [RFC8346], this network topology instance becomes a transport network slice instance that can perform layer 3 traffic engineering.

### 2.3. ACTN for Network Slicing

Since ACTN topology data models are based on the network topology model defined in [RFC8345], the augmentations defined in this document are effective augmentations to the ACTN topology data models, resulting in making the ACTN framework [RFC8453] and data models [I-D.ietf-teas-actn-yang] capable of slicing networks with the required network characteristics.

## 3. Model Applicability

There are many technologies to achieve transport network slicing. The data model defined in this document can be applied to a wide ranges of cases. This section describes how this data model is applied to a few cases.

### 3.1. Network Slicing by Virtualization

In the case shown in Figure 3, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1. Each of the virtual routers takes a portion of the resources such as ports and memory in the physical router. Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.

As an example, Appendix A. shows the JSON encoded data instances of the native topology and the customized topology for Network Slice Blue.



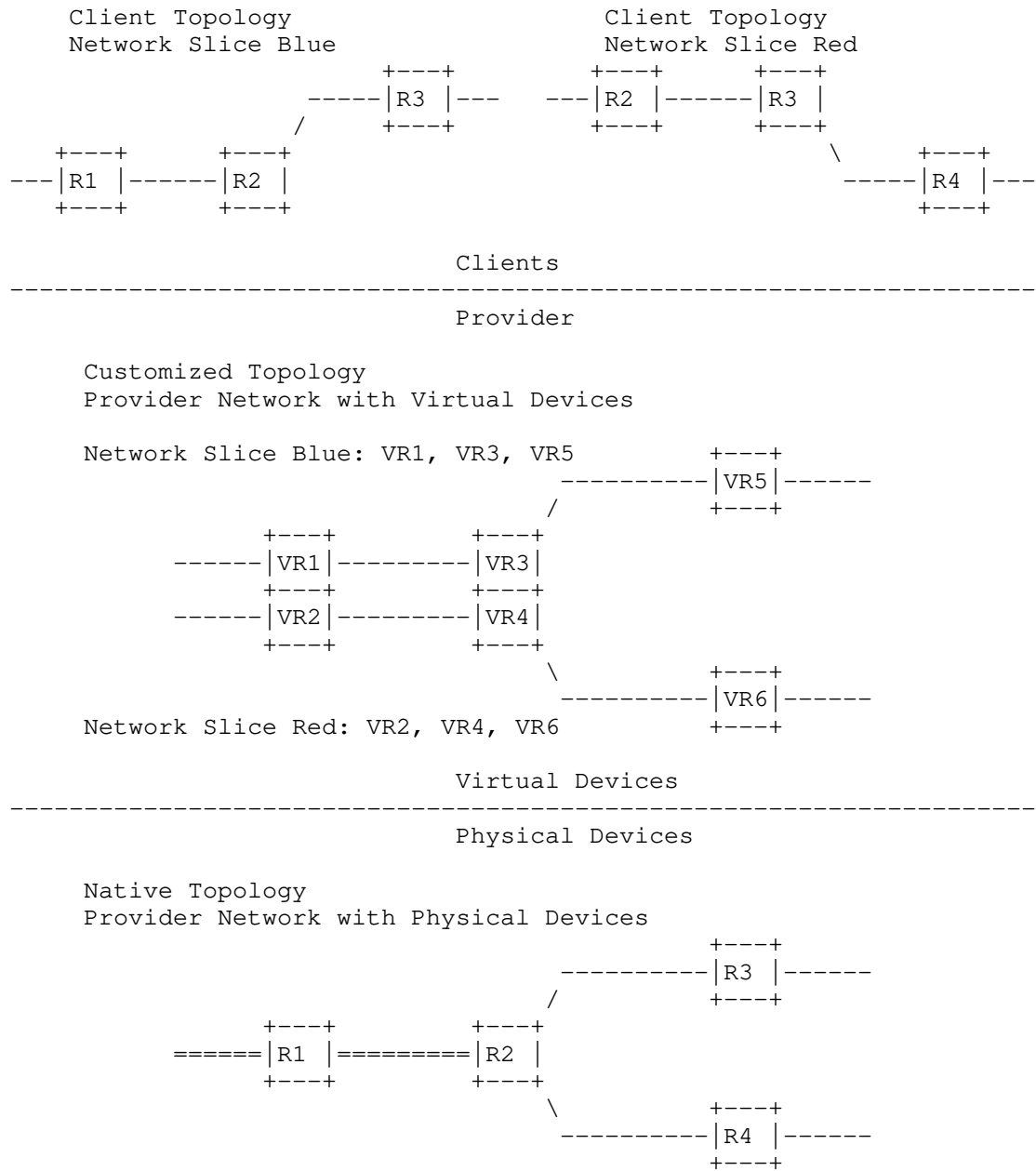


Figure 3: Network Slicing by Virtualization

### 3.2. Network Slicing by TE Overlay

Figure 5 shows a case where TE (Traffic Engineering) overlay is applied to achieve logically separated client transport network slices. In the underlay TE capable network, TE tunnels are established to support the TE links in the overlay network. These links and tunnels maintain the characteristics required by the clients. The provider selects the proper logical nodes and links in the overlay network, assigns them to specific transport network slices, and uses the data model defined in this document to send the results to the clients.

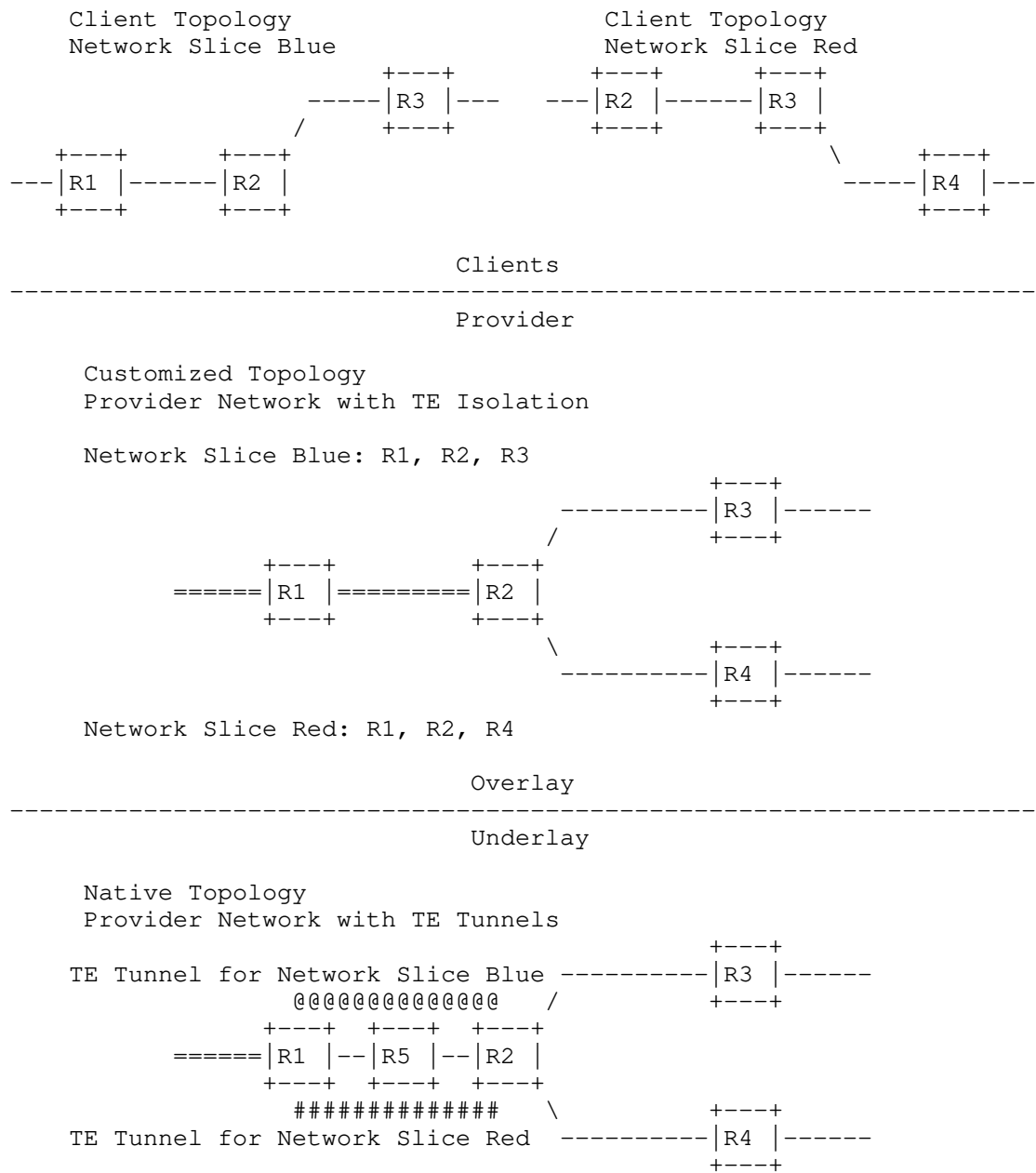


Figure 4: Network Slicing by TE Overlay

#### 4. Model Tree Structure

```
module: ietf-network-slice
  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-slice!
  augment /nw:networks/nw:network:
    +--rw network-slice
      +--rw optimization-criterion?  identityref
      +--rw delay-tolerance?         boolean
      +--rw periodicity*             uint64
      +--rw isolation-level?         identityref
  augment /nw:networks/nw:network/nw:node:
    +--rw network-slice
      +--rw isolation-level?  identityref
      +--rw compute-node-id? string
      +--rw storage-id?      string
  augment /nw:networks/nw:network/nt:link:
    +--rw network-slice
      +--rw delay-tolerance?  boolean
      +--rw periodicity*     uint64
      +--rw isolation-level?  identityref
```

#### 5. YANG Module

This module references [RFC8345], [I-D.ietf-teas-yang-te-types], and [GSMA-NS-Template]

```
<CODE BEGINS> file "ietf-network-slice@2019-10-15.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix "ns";

  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-te-types {
    prefix "te-types";
  }
```

```
reference
  "I-D.ietf-teas-yang-te-types: Traffic Engineering Common YANG
  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:     Xufeng Liu
               <mailto:xufeng.liu.ietf@gmail.com>

  Editor:     Jeff Tantsura
               <mailto:jefftant.ietf@gmail.com>

  Editor:     Igor Bryskin
               <mailto:i_bryskin@yahoo.com>

  Editor:     Luis Miguel Contreras Murillo
               <mailto:luismiguel.contrerasmurillo@telefonica.com>

  Editor:     Qin Wu
               <mailto:bill.wu@huawei.com>

  ";

description
  "YANG data model for representing and managing network
  slices.

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

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

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

revision 2019-10-15 {
  description "Initial revision";
```

```
    reference
      "RFC XXXX: YANG Data Model for Network Slices";
  }

  /*
   * Identities
   */
  identity isolation-level {
    description
      "Base identity for the isolation-level.";
    reference
      "GSMA-NS-Template: Generic Network Slice Template,
       Version 1.0.";
  }
  identity no-isolation {
    base isolation-level;
    description
      "Network slices are not separated.";
  }
  identity physical-isolation {
    base isolation-level;
    description
      "Network slices are physically separated (e.g. different rack,
       different hardware, different location, etc.).";
  }
  identity logical-isolation {
    base isolation-level;
    description
      "Network slices are logically separated.";
  }
  identity process-isolation {
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity physical-memory-isolation {
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity physical-network-isolation {
    base physical-isolation;
    description
      "Process and threads isolation.";
  }
  identity virtual-resource-isolation {
    base logical-isolation;
    description
```

```
    "A network slice has access to specific range of resources
    that do not overlap with other network slices
    (e.g. VM isolation).";
}
identity network-functions-isolation {
    base logical-isolation;
    description
        "NF (Network Function) is dedicated to the network slice, but
        virtual resources are shared.";
}
identity service-isolation {
    base logical-isolation;
    description
        "NSC data are isolated from other NSCs, but virtual
        resources and NFs are shared.";
}

/*
 * Groupings
 */
grouping network-slice-topology-attributes {
    description "Network Slice topology scope attributes.";
    container network-slice {
        description
            "Containing Network Slice attributes.";
        leaf optimization-criterion {
            type identityref {
                base te-types:objective-function-type;
            }
            description
                "Optimization criterion applied to this topology.";
        }
        leaf delay-tolerance {
            type boolean;
            description
                "'true' if is not too critical how long it takes to deliver
                the amount of data.";
            reference
                "GSMA-NS-Template: Generic Network Slice Template,
                Version 1.0.";
        }
        leaf-list periodicity {
            type uint64;
            units seconds;
            description
                "A list of periodicities supported by the network slice.";
            reference
                "GSMA-NS-Template: Generic Network Slice Template,
```

```
        Version 1.0.";
    }
    leaf isolation-level {
        type identityref {
            base isolation-level;
        }
        description
            "A network slice instance may be fully or partly, logically
            and/or physically, isolated from another network slice
            instance. This attribute describes different types of
            isolation:";
    }
} // network-slice
} // network-slice-topology-attributes

grouping network-slice-node-attributes {
    description "Network Slice node scope attributes.";
    container network-slice {
        description
            "Containing Network Slice attributes.";
        leaf isolation-level {
            type identityref {
                base isolation-level;
            }
            description
                "A network slice instance may be fully or partly, logically
                and/or physically, isolated from another network slice
                instance. This attribute describes different types of
                isolation:";
        }
        leaf compute-node-id {
            type string;
            description
                "Reference to a compute node instance specified in
                a data model specifying the computing resources.";
        }
        leaf storage-id {
            type string;
            description
                "Reference to a storage instance specified in
                a data model specifying the storage resources.";
        }
    }
} // network-slice
} // network-slice-node-attributes

grouping network-slice-link-attributes {
    description "Network Slice link scope attributes";
    container network-slice {
```



```
description
  "Containing Network Slice attributes.";
leaf delay-tolerance {
  type boolean;
  description
    "'true' if is not too critical how long it takes to deliver
    the amount of data.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 1.0.";
}
leaf-list periodicity {
  type uint64;
  units seconds;
  description
    "A list of periodicities supported by the network slice.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template,
    Version 1.0.";
}
leaf isolation-level {
  type identityref {
    base isolation-level;
  }
  description
    "A network slice instance may be fully or partly, logically
    and/or physically, isolated from another network slice
    instance. This attribute describes different types of
    isolation:";
}
} // network-slice
} // network-slice-link-attributes

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Defines the Network Slice topology type.";
  container network-slice {
    presence "Indicates Network Slice topology";
    description
      "Its presence identifies the Network Slice type.";
  }
}

augment "/nw:networks/nw:network" {
  when "nw:network-types/ns:network-slice" {
```

```
        description "Augment only for Network Slice topology.";
    }
    description "Augment topology configuration and state.";
    uses network-slice-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
    when "../nw:network-types/ns:network-slice" {
        description "Augment only for Network Slice topology.";
    }
    description "Augment node configuration and state.";
    uses network-slice-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
    when "../nw:network-types/ns:network-slice" {
        description "Augment only for Network Slice topology.";
    }
    description "Augment link configuration and state.";
    uses network-slice-link-attributes;
}
}
<CODE ENDS>
```

## 6. 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-network-slice
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-network-slice  
prefix:        ns  
reference:     RFC XXXX  
-----
```

## 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 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/ns:network-slice  
This subtree specifies the network slice type. Modifying the configurations can make network slice type invalid and cause interruption to transport network slices.

/nw:networks/nw:network/ns:network-slice  
This subtree specifies the topology-wide configurations. Modifying the configurations here can cause traffic characteristics changed in this transport network slice and related networks.

/nw:networks/nw:network/nw:node/ns:network-slice  
This subtree specifies the configurations of the nodes in a transport network slice. Modifying the configurations in this subtree can change the traffic characteristics on this node and the related networks.

/nw:networks/nw:network/nt:link/ns:network-slice

This subtree specifies the configurations of the links in a transport network slice. Modifying the configurations in this subtree can change the traffic characteristics on this link and the related networks.

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/ns:network-slice  
Unauthorized access to this subtree can disclose the network slice type.

/nw:networks/nw:network/ns:network-slice  
Unauthorized access to this subtree can disclose the topology-wide states.

/nw:networks/nw:network/nw:node/ns:network-slice  
Unauthorized access to this subtree can disclose the operational state information of the nodes in a transport network slice.

/nw:networks/nw:network/nt:link/ns:network-slice  
Unauthorized access to this subtree can disclose the operational state information of the links in a transport network slice.

## 8. Acknowledgements

The TEAS Network Slicing Design Team (NSDT) members included Aijun Wang, Dong Jie, Eric Gray, Jari Arkko, Jeff Tantsura, John E Drake, Luis M. Contreras, Rakesh Gandhi, Ran Chen, Reza Rokui, Ricard Vilalta, Ron Bonica, Sergio Belotti, Tomonobu Niwa, Xuesong Geng, and Xufeng Liu.

## 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>>.

- [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>>.

[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-11 (work in progress), October 2019.

[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.

[GSMA-NS-Template]  
GSM Association, "Generic Network Slice Template, Version 1.0", NG.116, May 2019.

## 9.2. Informative References

[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>>.

[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>>.

[I-D.ietf-ccamp-otn-topo-yang]  
Zheng, H., Guo, A., Busi, I., Sharma, A., Liu, X., Belotti, S., Xu, Y., Wang, L., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-08 (work in progress), September 2019.

[I-D.ietf-i2rs-yang-l2-network-topology]  
Dong, J., Wei, X., WU, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer-2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-12 (work in progress), October 2019.

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

Lee, Y., Zheng, H., Ceccarelli, D., Yoon, B., Dios, O.,  
Shin, J., and S. Belotti, "Applicability of YANG models  
for Abstraction and Control of Traffic Engineered  
Networks", draft-ietf-teas-actn-yang-04 (work in  
progress), August 2019.

## Appendix A. Data Tree for the Example in Section 3.1.

## A.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the native topology depicted in Figure 3.

```
{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-native-topology",
        "network-types": {
        },
        "node": [
          {
            "node-id": "R1",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "1-0-1"
              },
              {
                "tp-id": "1-0-2"
              },
              {
                "tp-id": "1-2-1"
              },
              {
                "tp-id": "1-2-2"
              }
            ]
          },
          {
            "node-id": "R2",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "2-1-1"
              },
              {
                "tp-id": "2-1-2"
              },
              {
                "tp-id": "2-3-1"
              },
              {
                "tp-id": "2-4-1"
              }
            ]
          }
        ]
      }
    ]
  }
}
```



```
    ]
  },
  {
    "node-id": "R3",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "3-0-1"
      },
      {
        "tp-id": "3-2-1"
      }
    ]
  },
  {
    "node-id": "R4",
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "4-0-1"
      },
      {
        "tp-id": "4-2-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "R1,1-0-1,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-1"
    }
  },
  {
    "link-id": ",,R1,1-0-1",
    "destination": {
      "dest-node": "R1",
      "dest-tp": "1-0-1"
    }
  },
  {
    "link-id": "R1,1-0-2,,",
    "source": {
      "source-node": "R1",
      "source-tp": "1-0-2"
    }
  },
  {

```

```
    "link-id":",,R1,1-0-2",
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-0-2"
    }
  },
  {
    "link-id":"R1,1-2-1,R2,2-1-1",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-1"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-1"
    }
  },
  {
    "link-id":"R2,2-1-1,R1,1-2-1",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-1"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-1"
    }
  },
  {
    "link-id":"R1,1-2-2,R2,2-1-2",
    "source": {
      "source-node":"R1",
      "source-tp":"1-2-2"
    },
    "destination": {
      "dest-node":"R2",
      "dest-tp":"2-1-2"
    }
  },
  {
    "link-id":"R2,2-1-2,R1,1-2-2",
    "source": {
      "source-node":"R2",
      "source-tp":"2-1-2"
    },
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-2-2"
    }
  }
```

```
    }  
  },  
  {  
    "link-id": "R2,2-3-1,R3,3-2-1",  
    "source": {  
      "source-node": "R2",  
      "source-tp": "2-3-1"  
    },  
    "destination": {  
      "dest-node": "R3",  
      "dest-tp": "3-2-1"  
    }  
  },  
  {  
    "link-id": "R3,3-2-1,R2,2-3-1",  
    "source": {  
      "source-node": "R3",  
      "source-tp": "3-2-1"  
    },  
    "destination": {  
      "dest-node": "R2",  
      "dest-tp": "2-3-1"  
    }  
  },  
  {  
    "link-id": "R2,2-4-1,R4,4-2-1",  
    "source": {  
      "source-node": "R2",  
      "source-tp": "2-4-1"  
    },  
    "destination": {  
      "dest-node": "R4",  
      "dest-tp": "4-2-1"  
    }  
  },  
  {  
    "link-id": "R4,4-2-1,R2,2-4-1",  
    "source": {  
      "source-node": "R4",  
      "source-tp": "4-2-1"  
    },  
    "destination": {  
      "dest-node": "R2",  
      "dest-tp": "2-4-1"  
    }  
  },  
  {  
    "link-id": "R3,3-0-1,,",
```

```

        "source": {
            "source-node": "R3",
            "source-tp": "3-0-1"
        },
        {
            "link-id": ",,R3,3-0-1",
            "destination": {
                "dest-node": "R3",
                "dest-tp": "3-0-1"
            }
        },
        {
            "link-id": "R4,4-0-1,,",
            "source": {
                "source-node": "R4",
                "source-tp": "4-0-1"
            }
        },
        {
            "link-id": ",,R4,4-0-1",
            "destination": {
                "dest-node": "R4",
                "dest-tp": "4-0-1"
            }
        }
    ]
}
]
}
}

```

#### A.2. Network Slice Blue

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network-slice" for the topology customized for Network Slice Blue depicted in Figure 3.

```

{
  "ietf-network:networks": {
    "network": [
      {
        "network-id": "example-customized-blue-topology",
        "network-types": {
          "ietf-network-slice:network-slice": {

```

```
    }
  },
  "supporting-network": [
    {
      "network-ref": "example-native-topology"
    }
  ],
  "node": [
    {
      "node-id": "VR1",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R1"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "1-0-1"
        },
        {
          "tp-id": "1-3-1"
        }
      ]
    },
    {
      "node-id": "VR3",
      "supporting-node": [
        {
          "network-ref": "example-native-topology",
          "node-ref": "R2"
        }
      ],
      "ietf-network-slice:network-slice": {
        "isolation-level":
          "ietf-network-slice:physical-memory-isolation"
      },
      "ietf-network-topology:termination-point": [
        {
          "tp-id": "3-1-1"
        },
        {
          "tp-id": "3-5-1"
        }
      ]
    }
  ]
}
```

```
    ]
  },
  {
    "node-id": "VR5",
    "supporting-node": [
      {
        "network-ref": "example-native-topology",
        "node-ref": "R3"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id": "5-3-1"
      },
      {
        "tp-id": "5-0-1"
      }
    ]
  }
],
"ietf-network-topology:link": [
  {
    "link-id": "VR1,1-0-1,,",
    "source": {
      "source-node": "VR1",
      "source-tp": "1-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R1,1-0-1,,"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR1,1-0-1",
    "destination": {
      "dest-node": "VR1",
      "dest-tp": "1-0-1"
    }
  },

```

```
"supporting-link": [  
  {  
    "network-ref": "example-native-topology",  
    "link-ref": ",,R1,1-0-1"  
  }  
],  
"ietf-network-slice:network-slice": {  
  "isolation-level":  
    "ietf-network-slice:physical-network-isolation"  
}  
},  
{  
  "link-id": "VR1,1-3-1,VR3,3-1-1",  
  "source": {  
    "source-node": "VR1",  
    "source-tp": "1-3-1"  
  },  
  "destination": {  
    "dest-node": "VR3",  
    "dest-tp": "3-1-1"  
  },  
  "supporting-link": [  
    {  
      "network-ref": "example-native-topology",  
      "link-ref": "R1,1-2-1,R2,2-1-1"  
    }  
  ],  
  "ietf-network-slice:network-slice": {  
    "isolation-level":  
      "ietf-network-slice:physical-network-isolation"  
  }  
},  
{  
  "link-id": "VR3,3-1-1,VR1,1-3-1",  
  "source": {  
    "source-node": "VR3",  
    "source-tp": "3-1-1"  
  },  
  "destination": {  
    "dest-node": "R1",  
    "dest-tp": "1-3-1"  
  },  
  "supporting-link": [  
    {  
      "network-ref": "example-native-topology",  
      "link-ref": "R2,2-1-1,R1,1-2-1"  
    }  
  ],  
}
```

```
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR3, 3-5-1, VR5, 5-3-1",
    "source": {
      "source-node": "VR3",
      "source-tp": "3-5-1"
    },
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-3-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R2, 2-3-1, R3, 3-2-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": "VR5, 5-3-1, VR3, 3-5-1",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-3-1"
    },
    "destination": {
      "dest-node": "VR3",
      "dest-tp": "3-5-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3, 3-2-1, R2, 2-3-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
```



```

    "link-id": "VR5,5-0-1,,",
    "source": {
      "source-node": "VR5",
      "source-tp": "5-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": "R3,3-0-1,, "
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  },
  {
    "link-id": ",,VR5,5-0-1",
    "destination": {
      "dest-node": "VR5",
      "dest-tp": "5-0-1"
    },
    "supporting-link": [
      {
        "network-ref": "example-native-topology",
        "link-ref": ",,R3,3-0-1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
  }
],
"ietf-network-slice:network-slice": {
  "optimization-criterion":
    "ietf-te-types:of-minimize-cost-path",
  "isolation-level":
    "ietf-network-slice:physical-isolation"
}
}
]
}
}

```

Authors' Addresses

Xufeng Liu  
Volta Networks

EMail: xufeng.liu.ietf@gmail.com

Jeff Tantsura  
Apstra Networks

EMail: jefftant.ietf@gmail.com

Igor Bryskin  
Individual

EMail: i\_bryskin@yahoo.com

Luis Miguel Contreras Murillo  
Telefonica

EMail: luismiguel.contrerasmurillo@telefonica.com

Qin Wu  
Huawei

EMail: bill.wu@huawei.com

TEAS Working Group  
Internet Draft  
Intended status: Informational

Fabio Peruzzini  
TIM  
Italo Busi  
Huawei  
Daniel King  
Old Dog Consulting  
Sergio Belotti  
Nokia  
Gabriele Galimberti  
Cisco

Expires: April 2020

October 31, 2019

Applicability of Abstraction and Control of Traffic Engineered  
Networks (ACTN) to Packet Optical Integration (POI)

draft-peru-teas-actn-poi-applicability-02.txt

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), 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 April 31, 2020.

## 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 considers the applicability of ACTN to Packet Optical Integration (POI) and IP and Optical DWDM domain internetworking, and specifically the YANG models being defined by the IETF to support this deployment architecture.

In this document we highlight the IETF protocols and data models that may be used for the ACTN and control of POI networks, with particular focus on the interfaces between the MDSC (Multi-Domain Service Coordinator) and the underlying Packet and Optical Domain Controllers (P-PNC and O-PNC) to support Packet Optical Integration (POI) use cases.

## Table of Contents

1. Introduction.....	3
2. Reference Scenario.....	4
2.1. Generic Assumptions.....	6
3. Scenario 1 - Multi-Layer Topology Coordination.....	7
3.1. Discovery of existing Och, ODU, IP links, IP tunnels and IP services.....	7
3.1.1. Common YANG models used at the MPIs.....	7
3.1.1.1. YANG models used at the Optical MPIs.....	8
3.1.1.2. Required YANG models at the Packet MPIs.....	8
3.1.2. Inter-domain link Discovery.....	9
3.2. Provisioning of an IP Link/LAG over DWDM.....	10
3.2.1. YANG models used at the MPIs.....	10
3.2.1.1. YANG models used at the Optical MPIs.....	10
3.2.1.2. Required YANG models at the Packet MPIs.....	11

3.2.2. IP Link Setup Procedure.....	11
3.3. Provisioning of an IP link/LAG over DWDM with path constraints.....	12
3.3.1. YANG models used at the MPIs.....	12
3.4. Provisioning of an additional link member to an existing LAG with or without path constraints.....	12
3.4.1. YANG models used at the MPIs.....	13
4. Multi-Layer Recovery Coordination.....	13
4.1. Ensuring Network Resiliency during Maintenance Events....	13
4.2. Router port failure.....	13
5. Security Considerations.....	14
6. Operational Considerations.....	14
7. IANA Considerations.....	15
8. References.....	15
8.1. Normative References.....	15
8.2. Informative References.....	16
9. Acknowledgments.....	16
10. Authors' Addresses.....	16

## 1. Introduction

Packet Optical Integration (POI) is an advanced use case of traffic engineering. In wide area networks, a packet network based on the Internet Protocol (IP) and possibly Multiprotocol Label Switching (MPLS) is typically realized on top of an optical transport network that uses Dense Wavelength Division Multiplexing (DWDM). In many existing network deployments, the packet and the optical networks are engineered and operated independently of each other. There are technical differences between the technologies (e.g., routers vs. optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP vs. dealing with physical impairments in DWDM, or very different time scales). In addition, customers and customer needs can be different between a packet and an optical network, and it is not uncommon to use different vendors in both domains. Last but not least, state-of-the-art packet and optical networks use sophisticated but complex technologies, and for a network engineer it may not be trivial to be a full expert in both areas. As a result, packet and optical networks are often operated in technical and organizational silos.

This separation is inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by better integrating the packet and the optical network. Multi-layer online topology insight can speed up troubleshooting (e.g., alarm correlation) and network operation (e.g., coordination of maintenance events), multi-layer offline

topology inventory can improve service quality (e.g., detection of diversity constraint violations) and multi-layer traffic engineering can use the available network capacity more efficiently (e.g., coordination of restoration). In addition, provisioning workflows can be simplified or automated as needed across layers (e.g, to achieve bandwidth on demand, or to perform maintenance events).

Fully leveraging these benefits requires an integration between the management and control of the packet and the optical network. The Abstraction and Control of TE Networks (ACTN) framework defines functions and interfaces between a Multi-Domain Service Coordinator (MDSC) and Provisioning Network Controllers (PNCs) that can be used for coordinating the packet and optical layers.

In this document, key use cases for Packet Optical Integration (POI) are described both from the point of view of the optical and the packet layer. The objective is to explain the benefit and the impact for both the packet and the optical layer, and to identify the required interaction between both layers. Precise definitions of use cases can help with achieving a common understanding across different disciplines. The focus of the use cases are IP networks operated as client of optical DWDM networks. The use cases are ordered by increasing level of integration and complexity. For each multi-layer use case, the document analyzes how to use the interfaces and data models of the ACTN architecture.

Understanding the level of standardization and the gaps will help to better assess the feasibility of integration between IP and Optical DWDM domain, in an end-to-end multi-vendor service provisioning perspective.

## 2. Reference Scenario

This document is considering a network scenario with multiple Optical domains and multiple Packet domains.

Figure 1 shows this scenario in case of two Optical domains and two Packet domains:

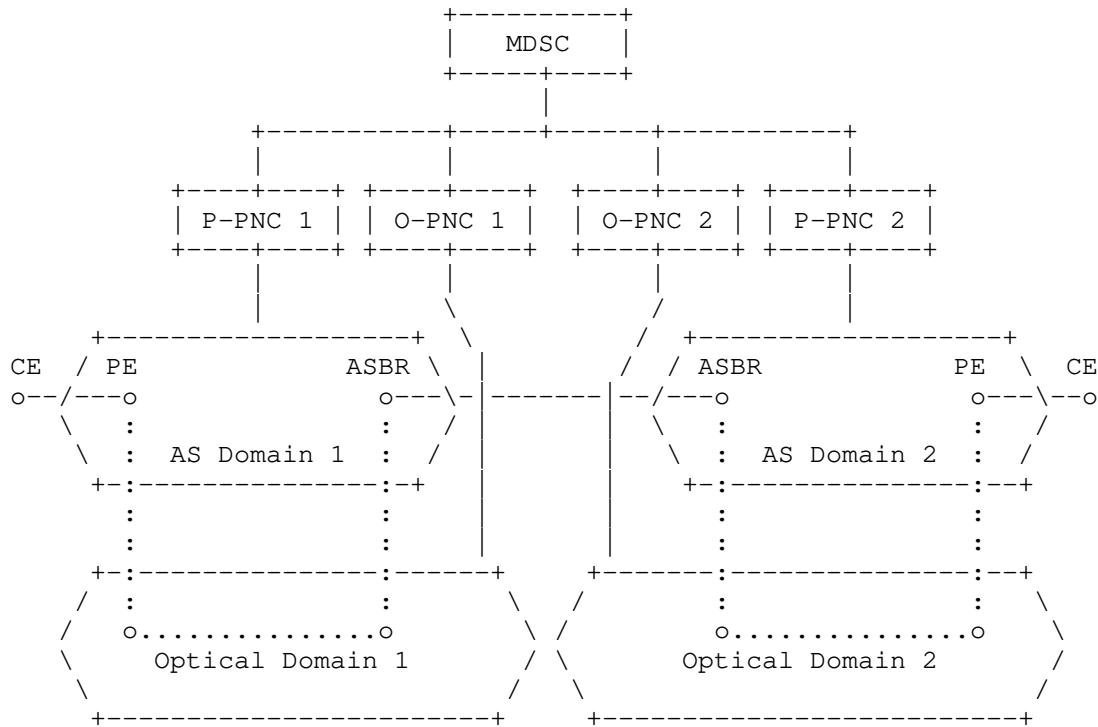


Figure 1 - Reference Scenario

The ACTN architecture, defined in [RFC8453], is used to control this multi-domain network where each Packet PNC (P-PNC) is responsible for controlling its IP domain (AS), and each Optical PNC (O-PNC) is responsible for controlling its Optical Domain. The MDSC is responsible for coordinating the whole multi-domain multi-layer (Packet and Optical) network. A specific standard interface (MPI) permits MDSC to interact with the different Provisioning Network Controller (O/P-PNCs). The MPI interface presents an abstracted topology to MDSC hiding technology-specific aspects of the network and hiding topology details depending on the policy chosen regarding the level of abstraction supported. The level of abstraction can be obtained based on P-PNC and O-PNC configuration parameters (e.g. provide the potential connectivity between any PE and any ABSR in an MPLS-TE network).

The MDSC in Figure 1 is responsible for multi-domain and multi-layer coordination across multiple Packet and Optical domains, as well as

to provide IP services to different CNCs at its CMIs (e.g., using L2SM, L3SM).

The multi-domain coordination mechanisms for the IP tunnels supporting these IP services are outside the scope of this document and described in [ACTN-VPN]. In some cases, the MDSC could also rely on the multi-layer Packet Optical Integration mechanisms, described in this draft, to support multi-layer optimizations for these IP services and tunnels.

In the network scenario of Figure 1, it is assumed that:

- o The domain boundaries between the IP and Optical domains are congruent. In other words, one Optical domain supports connectivity between Routers in one and only one Packet Domain.
- o Inter-domain links exist only between Packet domains (i.e., between ASBR routers) and between Packet and Optical domains (i.e., between routers and ROADMs). In other words, there are no inter-domain links between Optical domains
- o The interfaces between the routers and the ROADM's are "Ethernet" physical interfaces
- o The interfaces between the ASBR routers are "Ethernet" physical interfaces

## 2.1. Generic Assumptions

This section describes general assumptions which are applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, and also to all the scenarios discussed in this document.

The data models used on these interfaces are assumed to use the YANG 1.1 Data Modeling Language, as defined in [RFC7950].

The RESTCONF protocol, as defined in [RFC8040], using the JSON representation, defined in [RFC7951], is assumed to be used at these interfaces.

As required in [RFC8040], the "ietf-yang-library" YANG module defined in [RFC8525] is used to allow the MDSC to discover the set of YANG modules supported by each PNC at its MPI.



### 3. Scenario 1 - Multi-Layer Topology Coordination

In this scenario, the MSDC needs to discover the network topology, at both WDM and IP layers, in terms of nodes (NEs) and links, including inter AS domain links as well as cross-layer links.

Each PNC provides to the MSDC an abstract topology view of the WDM or of the IP topology of the domain it controls. This topology is abstracted in the sense that some detailed NE information is hidden at the MPI, and all or some of the NEs and related physical links are exposed as abstract nodes and logical (virtual) links, depending on the level of abstraction the user want. This detailed information is key to understand both the inter-AS domain links (seen by each controller as UNI interfaces but as I-NNI interfaces by the MSDC) as well as the cross-layer mapping between IP and WDM layer.

The MSDC also maintains an up-to-date network inventory of both IP and WDM layers through the use of IETF notifications through MPI with the PNCs.

For the cross-layer links it is key for MSDC to be able to correlate automatically the information about the physical ports from the routers (single link or bundle links for LAG) to client ports in the ROADMs.

#### 3.1. Discovery of existing OCh, ODU, IP links, IP tunnels and IP services

In this scenarios MSDC must be able to automatically discover network topology of both WDM and IP layers (links and NE, links between two domains).

- o An abstract view of the WDM and IP topology must be available.
- o MSDC must keep an up-to-date network inventory of both IP and WDM layers and it should be possible to correlate such information (e.g.: which port, lambda/OTSi, direction is used by a specific IP service on the WDM equipment).
- o It should be possible at MSDC level to easily correlate WDM and IP layers alarms to speed-up troubleshooting.

##### 3.1.1. Common YANG models used at the MPIs

Both Optical and Packet PNCs use the following common topology YANG models at the MPI to report their abstract topologies:

- o The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345]
- o The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model
- o The TE Topology Model, defined in the "ietf-te-topology" YANG module of [TE-TOPO], which augments the Base Network Topology Model

These common YANG models are generic and augmented by technology-specific YANG modules as described in the following sections.

#### 3.1.1.1. YANG models used at the Optical MPIs

The Optical PNC also uses at least the following technology-specific topology YANG models, providing WDM and Ethernet technology-specific augmentations of the generic TE Topology Model:

- o The WSON Topology Model, defined in the "ietf-wson-topology" YANG modules of [WSON-TOPO], or the Flexi-grid Topology Model, defined in the "ietf-flexi-grid-topology" YANG module of [Flexi-TOPO].
- o The Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO]

The WSON Topology Model or, alternatively, the Flexi-grid Topology model is used to report the DWDM network topology (e.g., ROADMs and links) depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The Ethernet Topology is used to report the access links between the IP routers and the edge ROADMs.

#### 3.1.1.2. Required YANG models at the Packet MPIs

The Packet PNC also uses at least the following technology-specific topology YANG models, providing IP and Ethernet technology-specific augmentations of the generic Topology Models:

- o The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG modules of [RFC8346], which augments the Base Network Topology Model

- o The Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO], which augments the TE Topology Model

The Ethernet Topology Model is used to report the access links between the IP routers and the edge ROADMs as well as the inter-domain links between ASBRs, while the L3 Topology Model is used to report the IP network topology (e.g., IP routers and links).

### 3.1.2. Inter-domain link Discovery

In the reference network of Figure 1, there are two types of inter-domain links:

- o Links between two IP domains (ASes)
- o Links between an IP router and a ROADM

Both types of links are Ethernet physical links.

The inter-domain link information is reported to the MDSC by the two adjacent PNCs, controlling the two ends of the inter-domain link.

The MDSC can understand how to merge these inter-domain links together using the plug-id attribute defined in the TE Topology Model [TE-TOPO], as described in as described in section 4.3 of [TE-TOPO].

A more detailed description of how the plug-id can be used to discover inter-domain link is also provided in section 5.1.4 of [TNBI].

Both types of inter-domain links are discovered using the plug-id attributes reported in the Ethernet Topologies exposed by the two adjacent PNCs. The MDSC can also discover an inter-domain IP link/adjacency between the two IP LTPs, reported in the IP Topologies exposed by the two adjacent P-PNCs, supported by the two ETH LTPs of an Ethernet Link discovered between these two P-PNCs.

Two options are possible to discover these inter-domain links:

1. Static configuration
2. LLDP [IEEE 802.1AB] automatic discovery

Since the static configuration requires an administrative burden to configure network-wide unique identifiers, the automatic discovery solution based on LLDP is preferable when LLDP is supported.

As outlined in [TNBI], the encoding of the plug-id namespace as well as of the LLDP information within the plug-id value is implementation specific and needs to be consistent across all the PNCs.

### 3.2. Provisioning of an IP Link/LAG over DWDM

In this scenario, the MSDC needs to coordinate the creation of an IP link, or a LAG, between two routers through a DWDM network.

It is assumed that the MDSC has already discovered the whole network topology as described in section 3.1.

#### 3.2.1. YANG models used at the MPIs

##### 3.2.1.1. YANG models used at the Optical MPIs

The Optical PNC uses at least the following YANG models:

- o The TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL]
- o The WSON Tunnel Model, defined in the "ietf-wson-tunnel" YANG modules of [WSON-TUNNEL], or the Flexi-grid Media Channel Model, defined in the "ietf-flexi-grid-media-channel" YANG module of [Flexi-MC]
- o The Ethernet Client Signal Model, defined in the "ietf-eth-tran-service" YANG module of [CLIENT-SIGNAL]

The TE Tunnel model is generic and augmented by technology-specific models such as the WSON Tunnel Model and the Flexi-grid Media Channel Model.

The WSON Tunnel Model or, alternatively, the Flexi-grid Media Channel Model are used to setup connectivity within the DWDM network depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The Ethernet Client Signal Model is used to configure the steering of the Ethernet client traffic between Ethernet access links and TE Tunnels, which in this case could be either WSON Tunnels or Flexi-Grid Media Channels. This model is generic and applies to any technology-specific TE Tunnel: technology-specific attributes are provided by the technology-specific models which augment the generic TE-Tunnel Model.

#### 3.2.1.2. Required YANG models at the Packet MPIs

The Packet PNC uses at least the following topology YANG models:

- o The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345] (see section 3.1.1)
- o The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345] (see section 3.1.1)
- o The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG modules of [RFC8346] (see section 3.1.1.1)

If, as discussed in section 3.2.2, IP Links created over DWDM can be automatically discovered by the P-PNC, the IP Topology is needed only to report these IP Links after being discovered by the P-PNC.

The IP Topology can also be used to configure the IP Links created over DWDM.

#### 3.2.2. IP Link Setup Procedure

The MDSC requires the O-PNC to setup a WDM Tunnel (either a WSON Tunnel or a Flexi-grid Tunnel) within the DWDM network between the two Optical Transponders (OTs) associated with the two access links.

The Optical Transponders are reported by the O-PNC as Trail Termination Points (TTPs), defined in [TE-TOPO], within the WDM Topology. The association between the Ethernet access link and the WDM TTP is reported by the Inter-Layer Lock (ILL) identifiers, defined in [TE-TOPO], reported by the O-PNC within the Ethernet Topology and WDM Topology.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two access Ethernet Links over the WDM Tunnel.

After the WDM Tunnel has been setup and the client traffic steering configured, the two IP routers can exchange Ethernet packets between themselves, including LLDP messages.

If LLDP [IEEE 802.1AB] is used between the two routers, the P-PNC can automatically discover the IP Link being setup by the MDSC. The IP LTPs terminating this IP Link are supported by the ETH LTPs terminating the two access links.

Otherwise, the MDSC needs to require the P-PNC to configure an IP Link between the two routers: the MDSC also configures the two ETH LTPs which support the two IP LTPs terminating this IP Link.

### 3.3. Provisioning of an IP link/LAG over DWDM with path constraints

MDSC must be able to provision an IP link with a fixed maximum latency constraint, or with the minimum latency available constraint within each domain but as well inter-domain when required (e.g. by monitoring traffic KPIs trends for this IP link). Through the O-PNC fixed latency path/minimum latency path is chosen between PE and ASBR in each optical domain. Then MDSC needs to select the inter-AS domain with less latency (in case we have several interconnection links) to have the right low latency constraint fulfilled end-to-end across domains.

MDSC must be able to automatically create two IP links between two routers, over DWDM network, with physical path diversity (avoiding SRLGs communicated by O-PNCs to the MDSC).

MDSC must be responsible to route each of this IP links through different inter-AS domain links so that end-to-end IP links are fully disjoint.

Optical connectivity must be set up accordingly by MDSC through O-PNCs.

#### 3.3.1. YANG models used at the MPIS

This is for further study

### 3.4. Provisioning of an additional link member to an existing LAG with or without path constraints

For adding an additional link member to a LAG between two routers with or without path latency/diversity constraint. MDSC must be able

to force additional optical connection to use the same physical path in the optical domain where the LAG capacity increase is required.

#### 3.4.1. YANG models used at the MPIs

This is for further study

### 4. Multi-Layer Recovery Coordination

#### 4.1. Ensuring Network Resiliency during Maintenance Events

Before planned maintenance operation on DWDM network takes place, IP traffic should be moved hitless to another link.

MDSC must reroute IP traffic before the events takes place. It should be possible to lock IP traffic to the protection route until the maintenance event is finished, unless a fault occurs on such path.

#### 4.2. Router port failure

The focus is on client-side protection scheme between IP router and reconfigurable ROADM. Scenario here is to define only one port in the routers and in the ROADM muxponder board at both ends as back-up ports to recover any other port failure on client-side of the ROADM (either on router port side or on muxponder side or on the link between them). When client-side port failure occurs, alarms are raised to MDSC by IP-PNC and O-PNC (port status down, LOS etc.). MDSC checks with OP-PNC(s) that there is no optical failure in the optical layer.

There can be two cases here:

- a) LAG was defined between the two end routers. MDSC, after checking that optical layer is fine between the two end ROADMs, triggers the ROADM configuration so that the router back-up port with its associated muxponder port can reuse the OCh that was already in use previously by the failed router port and adds the new link to the LAG on the failure side.

While the ROADM reconfiguration takes place, IP/MPLS traffic is using the reduced bandwidth of the IP link bundle, discarding lower priority traffic if required. Once backup port has been reconfigured to reuse the existing OCh and new link has been added to the LAG then original Bandwidth is recovered between the end routers.

Note: in this LAG scenario let assume that BFD is running at LAG level so that there is nothing triggered at MPLS level when one of the link member of the LAG fails.

- b) If there is no LAG then the scenario is not clear since a router port failure would automatically trigger (through BFD failure) first a sub-50ms protection at MPLS level :FRR (MPLS RSVP-TE case) or TI-LFA (MPLS based SR-TE case) through a protection port. At the same time MDSC, after checking that optical network connection is still fine, would trigger the reconfiguration of the back-up port of the router and of the ROADM muxponder to reuse the same OCh as the one used originally for the failed router port. Once everything has been correctly configured, MDSC Global PCE could suggest to the operator to trigger a possible re-optimisation of the back-up MPLS path to go back to the MPLS primary path through the back-up port of the router and the original OCh if overall cost, latency etc. is improved. However, in this scenario, there is a need for protection port PLUS back-up port in the router which does not lead to clear port savings.

## 5. Security Considerations

Several security considerations have been identified and will be discussed in future versions of this document.

## 6. Operational Considerations

Telemetry data, such as the collection of lower-layer networking health and consideration of network and service performance from POI domain controllers, may be required. These requirements and capabilities will be discussed in future versions of this document.



## 7. IANA Considerations

This document requires no IANA actions.

## 8. References

### 8.1. Normative References

- [RFC7950] Bjorklund, M. et al., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, August 2016.
- [RFC8040] Bierman, A. et al., "RESTCONF Protocol", RFC 8040, January 2017.
- [RFC8345] Clemm, A., Medved, J. et al., "A Yang Data Model for Network Topologies", RFC8345, March 2018.
- [RFC8346] Clemm, A. et al., "A YANG Data Model for Layer 3 Topologies", RFC8346, March 2018.
- [RFC8453] Ceccarelli, D., Lee, Y. et al., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC8453, August 2018.
- [RFC8525] Bierman, A. et al., "YANG Library", RFC 8525, March 2019.
- [IEEE 802.1AB] IEEE 802.1AB-2016, "IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery", March 2016.
- [TE-TOPO] Liu, X. et al., "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo, work in progress.
- [WSON-TOPO] Lee, Y. et al., " A YANG Data Model for WSON (Wavelength Switched Optical Networks)", draft-ietf-ccamp-wson-yang, work in progress.
- [Flexi-TOPO] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.
- [CLIENT-TOPO] Zheng, H. et al., "A YANG Data Model for Client-layer Topology", draft-zheng-ccamp-client-topo-yang, 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.

[WSON-TUNNEL] Lee, Y. et al., "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model, work in progress.

[Flexi-MC] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang, work in progress.

[CLIENT-SIGNAL] Zheng, H. et al., "A YANG Data Model for Transport Network Client Signals", draft-ietf-ccamp-client-signal-yang, work in progress.

## 8.2. Informative References

[TNBI] Busi, I., Daniel, K. et al., "Transport Northbound Interface Applicability Statement", draft-ietf-ccamp-transport-nbi-app-statement, work in progress.

[ACTN-VPN] Lee, Y. et al., "Applicability of ACTN to Support Packet and Optical Integration", draft-lee-teas-actn-poi-applicability, work in progress.

## 9. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

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

## 10. Authors' Addresses

Fabio Peruzzini  
TIM

Email: fabio.peruzzini@telecomitalia.it

Italo Busi  
Huawei

Email: Italo.busi@huawei.com

Daniel King  
Old Dog Consulting  
Email: daniel@olddog.co.uk

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

Gabriele Galimberti  
Cisco  
Email: ggalimbe@cisco.com

Zheng Yanlei  
China Unicom  
Email: zhengyanlei@chinaunicom.cn

Washington Costa Pereira Correia  
TIM Brasil  
Email: wcorreia@timbrasil.com.br

Jean-Francois Bouquier  
Vodafone  
Email: jeff.bouquier@vodafone.com



BESS Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 3, 2020

M. Wang  
Q. Wu  
R. Even  
Huawei  
B. Wen  
Comcast  
C. Liu  
China Unicom  
H. Xu  
China Telecom  
October 31, 2019

A YANG Model for Network and VPN Service Performance Monitoring  
draft-www-bess-yang-vpn-service-pm-04

Abstract

The data model defined in [RFC8345] introduces vertical layering relationships between networks that can be augmented to cover network/service topologies. This document defines a YANG model for both Network Performance Monitoring and VPN Service Performance Monitoring that can be used to monitor and manage network performance on the topology at higher layer or the service topology between VPN sites. This model is an augmentation to the network topology YANG data model defined in [RFC8345].

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 May 3, 2020.

## 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. Conventions used in this document . . . . .	3
2.1. Tree Diagrams . . . . .	3
3. Network and VPN service assurance module . . . . .	3
4. Layering relationship between multiple layers of topology . .	4
5. Model Usage Guideline . . . . .	5
5.1. Performance Monitoring Data Source . . . . .	5
5.2. Retrieval via I2RS Pub/Sub [RFC7923] . . . . .	5
5.3. On demand Retrieval via RPC model . . . . .	5
6. Design of the Data Model . . . . .	5
6.1. Network Level . . . . .	6
6.2. Node Level . . . . .	6
6.3. Link and Termination Point Level . . . . .	7
7. Example of I2RS Pub/Sub Retrieval [RFC7923] . . . . .	8
8. Example of RPC model based Retrieval . . . . .	10
9. Network and VPN Service Assurance YANG Module . . . . .	11
10. Security Considerations . . . . .	20
11. IANA Considerations . . . . .	21
12. Normative References . . . . .	22
Authors' Addresses . . . . .	23

## 1. Introduction

[RFC8345] defines an abstract YANG data model for network/service topologies and inventories. Service topology described in [RFC8345] includes the a virtual topology for a service layer above the L1, L2, and L3 layers. This service topology has the generic topology elements of node, link, and terminating point. One typical example of a service topology is described in figure 3 of [RFC8345], two VPN service topologies instantiated over a common L3 topology. Each VPN

service topology is mapped onto a subset of nodes from the common L3 topology.

In [RFC8299], 3 types of VPN service topologies are defined for the L3VPN service data model: any to any; hub and spoke; and hub and spoke disjoint. These VPN topology types can be used to describe how VPN sites communicate with each other.

This document defines a YANG Model for both Network performance monitoring and VPN Service Performance Monitoring that can be used to monitor and manage network Performance on the topology at higher layer or the service topology between VPN sites and it is an augmentation to the network topology YANG data model defined in [RFC8345].

## 2. Conventions used in this document

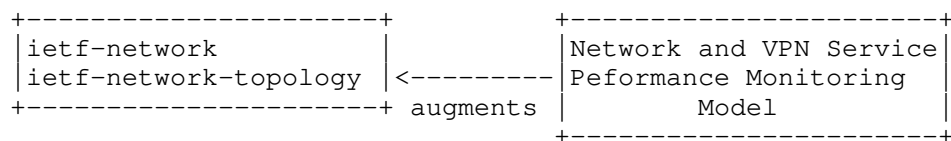
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

### 2.1. Tree Diagrams

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

## 3. Network and VPN service assurance module

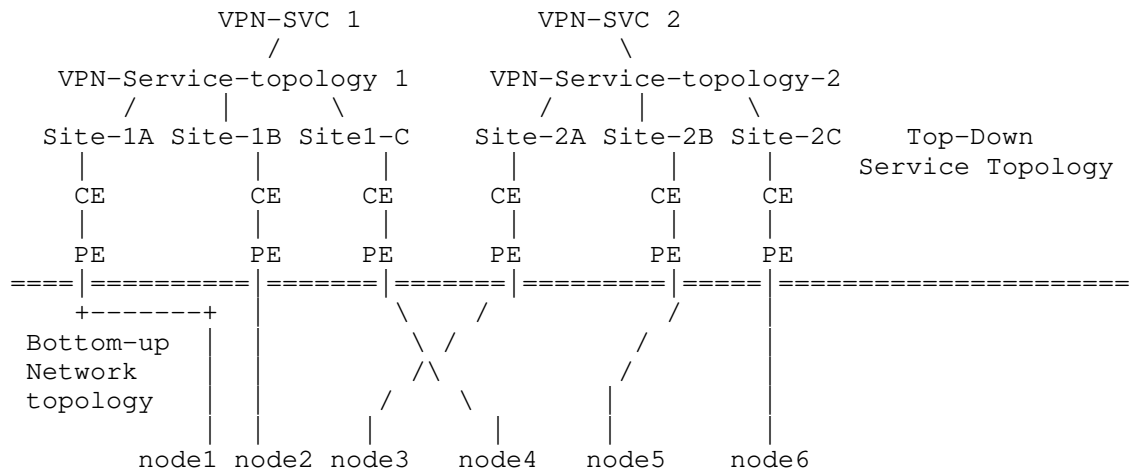
This module defined in this document is a Network and VPN Service assurance module that can be used to monitor and manage the network Performance on the topology at higher layer layer or the service topology between VPN sites and it is an augmentation to the "ietf-network" and "ietf-network-topology" YANG data model [RFC8345]. The performance monitoring data is augmented to service topology.



#### 4. Layering relationship between multiple layers of topology

The data model defined in [RFC8345] can describe vertical layering relationships between networks. That model can be augmented to cover network/service topologies.

Figure 1 describes an example on topology mapping between the VPN service topology and the underlying network:



Example of topology mapping between VPN Service Topo and Underlying network

As shown in Figure 1, Site-1A, Site-1B, and Site-1C are mapped to nodes 1, 2, and 3, while Site-2A, Site-2B, and Site-2C are mapped to nodes 4, 5, and 6 in the underlying physical network. In this figure, two VPN services topologies are both built on top of one common underlying physical network.

VPN-SVC 1: supporting hub-spoke communication for Customer 1 connecting the customers access at 3 sites

VPN-SVC 2: supporting hub-spoke disjoint communication for Customer 2 connecting the customers access at 3 sites

VPN service topology 1 is hub and spoke topology while VPN service topology 2 is hub and spoke disjoint topology. In VPN service topology 1, Site-1 A plays the role of hub while Site-2 B and C plays the role of spoke. In VPN service topoogy 2, Site-2 A and B play the role of hub while Site-2 C plays the role of spoke.



## 5. Model Usage Guideline

An SP must be able to manage the capabilities and characteristics of their Network/VPN services when Network connection is established or VPN sites are setup to communicate with each other. VPN service topology such as hub and spoke describes how these VPN sites are communicating with each other.

### 5.1. Performance Monitoring Data Source

As described in Section 4, once the mapping between VPN Service topology and underlying physical network has been setup, the performance monitoring data per link in the underlying network can be collected using network performance measurement method such as MPLS Loss and Delay Measurement [RFC6374]. The performance monitoring information reflecting the quality of the Network or VPN service such as end to end network performance data between source node and destination node in the network or between VPN sites can be aggregated or calculated using PCEP solution [RFC5440] or LMAP solution [RFC8194]. The information can be fed into data source such as the management system or network devices. The measurement interval and report interval associated with these performance data usually depends on configuration parameters.

### 5.2. Retrieval via I2RS Pub/Sub [RFC7923]

Some applications such as service-assurance applications, which must maintain a continuous view of operational data and state, can use subscription model [I-D.ietf-netconf-yang-push] to subscribe to the Network performance data or VPN service performance data they are interested in, at the data source.

The data source can then use the Network and VPN service assurance model defined in this document and push model [I-D.ietf-netconf-yang-push] to distribute specific telemetry data to target recipients.

### 5.3. On demand Retrieval via RPC model

To obtain a snapshot of a large amount of performance data from the network element, service-assurance applications can also use polling based solution such as RPC model to fetch performance data on demand.

## 6. Design of the Data Model

This document defines the YANG module "ietf-network-vpn-pm", which has the following structure

### 6.1. Network Level

```
module: ietf-network-vpn-pm
  augment /nw:networks/nw:network/nw:network-types:
    +--rw network-technology-type*  identityref
  augment /nw:networks/nw:network:
    +--rw vpn-topo-attributes
      +--rw vpn-topo?  identityref
```

#### Network Level View of the hierarchies

For VPN service performance monitoring, this model defines only the following minimal set of Network level network topology attributes:

- o Network-technology-type: Indicate the network technology type such as L3VPN, L2VPN, ISIS, OSPF. If the network-technology-type is VPN type, e.g., L3VPN, L2VPN, the VPN-topo should be set.
- o vpn-topo: The type of VPN service topology, this model supports any-to-any, Hub and Spoke (where Hubs can exchange traffic), and "Hub and Spoke disjoint" (where Hubs cannot exchange traffic).

For network performance monitoring, the attributes of "Network Level" that defined in [RFC8345] do not need to be extended.

### 6.2. Node Level

```
augment /nw:networks/nw:network/nw:node:
  +--rw node-attributes
    +--rw node-type?  identityref
    +--rw site-id?    string
    +--rw site-role?  Identityref
```

#### Node Level View of the hierarchies

The Network and VPN service performance monitoring model defines only the following minimal set of Node level network topology attributes and constraints:

- o Node-type (Attribute): Indicate the type of the node, such as PE or ASBR.
- o Site-id (Constraint): Uniquely identifies the site within the overall network infrastructure.
- o Site-role (Constraint): Defines the role of the site in a particular VPN topology.

### 6.3. Link and Termination Point Level

```

augment /nw:networks/nw:network/nt:link:
  +--rw link-type?                               identityref
  +--ro link-telemetry-attributes
    +--ro loss-statistics
      +--ro direction                             identityref
      +--ro packet-loss-count?                    uint32
      +--ro loss-ratio?                           percentage
      +--ro packet-reorder-count?                 uint32
      +--ro packets-out-of-seq-count?             uint32
      +--ro packets-dup-count?                    uint32
    +--ro delay-statistics
      +--ro direction?                           identityref
      +--ro min-delay-value?                      uint32
      +--ro max-delay-value?                      uint32
      +--ro average-delay-value?                  uint32
    +--ro jitter-statistics
      +--ro direction?                           identityref
      +--ro min-jitter-value?                     uint32
      +--ro max-jitter-value?                     uint32
      +--ro average-jitter-value?                 uint32
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--ro tp-telemetry-attributes
    +--ro in-octets?                              uint32
    +--ro inbound-unicast?                        uint32
    +--ro inbound-nunicast?                      uint32
    +--ro inbound-discards?                      uint32
    +--ro inbound-errors?                        uint32
    +--ro inunknown-protos?                      uint32
    +--ro out-octets?                             uint32
    +--ro outbound-unicast?                      uint32
    +--ro outbound-nunicast?                    uint32
    +--ro outbound-discards?                    uint32
    +--ro outbound-errors?                      uint32
    +--ro outbound-qlen?                         uint32

```

#### Link and Termination point Level View of the hierarchies

The Network and VPN service performance monitoring model defines only the following minimal set of Link level network topology attributes:

- o Link-type (Attribute): Indicate the type of the link, such as GRE, IP in IP.
- o Loss Statistics: A set of loss statistics attributes that are used to measure end to end loss between VPN sites.

- o Delay Statistics: A set of delay statistics attributes that are used to measure end to end latency between VPN sites.
- o Jitter Statistics: A set of jitter statistics attributes that are used to measure end to end jitter between VPN sites.

The Network and VPN service performance monitoring defines the following minimal set of Termination point level network topology attributes:

- o Inbound statistics: A set of inbound statistics attributes that are used to measure the inbound statistics of the termination point, such as "the total number of octets received on the termination point", "The number of inbound packets which were chosen to be discarded", "The number of inbound packets that contained errors", etc.
- o Outbound statistics: A set of outbound statistics attributes that are used to measure the outbound statistics of the termination point, such as "the total number of octets transmitted out of the termination point", "The number of outbound packets which were chosen to be discarded", "The number of outbound packets that contained errors", etc.

#### 7. Example of I2RS Pub/Sub Retrieval [RFC7923]

This example shows the way for a client to subscribe for the Performance monitoring information between node A and node B in the L3 network topology built on top of the underlying network . The performance monitoring parameter that the client is interested in is end to end loss attribute.

```
<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-subscribed-notifications">
    <stream-subtree-filter>
      <networks xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topo">
        <network>
          <network-id>l3-network</network-id>
          <network-technology-type xmlns="urn:ietf:params:xml:ns:yang:iet
f-network-vpn-pm">
            L3VPN
          </network-technology-type>
          <node>
            <node-id>A</node-id>
            <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-netw
ork-vpn-pm">
              <node-type>pe</node-type>
            </node-attributes>
```

```

work-topology">
    <termination-point xmlns="urn:ietf:params:xml:ns:yang:ietf-ne
etf-network-vpn-pm">
        <tp-id>1-0-1</tp-id>
        <tp-telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:i
            <in-octets>100</in-octets>
            <out-octets>150</out-octets>
        </tp-telemetry-attributes>
        </termination-point>
    </node>
    <node>
        <node-id>B</node-id>
        <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-netw
ork-vpn-pm">
            <node-type>pe</node-type>
        </node-attributes>
        <termination-point xmlns="urn:ietf:params:xml:ns:yang:ietf-ne
work-topology">
            <tp-id>2-0-1</tp-id>
            <tp-telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:i
etf-network-vpn-pm">
                <in-octets>150</in-octets>
                <out-octets>100</out-octets>
            </tp-telemetry-attributes>
            </termination-point>
        </node>
        <link xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topology
">
            <link-id>A-B</link-id>
            <source>
                <source-node>A</source-node>
            </source>
            <destination>
                <dest-node>B</dest-node>
            </destination>
            <link-type>mpls-te</link-type>
            <link-telemetry-attributes
                xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm">
                <loss-statistics>
                    <packet-loss-count>100</packet-loss-count>
                </loss-statistics>
            </link-telemetry-attributes>
            </link>
        </network>
    </networks>
</stream-subtree-filter>
<period xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">500</pe
riod>
</establish-subscription>
</rpc>

```

## 8. Example of RPC model based Retrieval

This example shows the way for the client to use RPC model to fetch performance data on demand, e.g., the client requests packet-loss-count between PE1 in site 1 and PE2 in site 2 belonging to the same VPN1.

```

<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  message-id="1">
  <report xmlns="urn:ietf:params:xml:ns:yang:example-service-pm-report">
    <networks xmlns="urn:ietf:params:xml:ns:yang:ietf-network-topo">
      <network>
        <network-id>vpn1</network-id>
        <node>
          <node-id>A</node-id>
          <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-p
m">
            <node-type>pe</node-type>
            </node-attributes>
            <termination-point xmlns="urn:ietf:params:xml:ns:yang:ietf-network-top
ology">
              <tp-id>1-0-1</tp-id>
              <tp-telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-netw
ork-vpn-pm">
                <in-octets>100</in-octets>
                <out-octets>150</out-octets>
              </tp-telemetry-attributes>
            </termination-point>
          </node>
          <node>
            <node-id>B</node-id>
            <node-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-network-vpn-p
m">
              <node-type>pe</node-type>
              </node-attributes>
              <termination-point xmlns="urn:ietf:params:xml:ns:yang:ietf-network-top
ology">
                <tp-id>2-0-1</tp-id>
                <tp-telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-netw
ork-vpn-pm">
                  <in-octets>150</in-octets>
                  <out-octets>100</out-octets>
                </tp-telemetry-attributes>
              </termination-point>
            </node>
            <link-id>A-B</link-id>
            <source>
              <source-node>A</source-node>
            </source>
            <destination>
              <dest-node>B</dest-node>
            </destination>
            <link-type>mpls-te</link-type>
            <telemetry-attributes xmlns="urn:ietf:params:xml:ns:yang:ietf-network-
pm">

```

```
        <loss-statistics>
          <packet-loss-count>120</packet-loss-count>
        </loss-statistics>
      </telemetry-attributes>
    </link>
  </network>
</report>
</rpc>
```

## 9. Network and VPN Service Assurance YANG Module

```
<CODE BEGINS> file "ietf-network-vpn-pm.yang"
module ietf-network-vpn-pm {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm";
  prefix nvp;

  import ietf-network {
    prefix nw;
  }
  import ietf-network-topology {
    prefix nt;
  }
  import ietf-l3vpn-svc {
    prefix l3vpn-svc;
  }

  organization
    "IETF BESS Working Group";
  contact
    "Zitao Wang: wangzitao@huawei.com
     Qin Wu: bill.wu@huawei.com";
  description
    "This module defines a model for the VPN Service Performance monitoring.";

  revision 2019-03-01 {
    description
      "Initial revision.";
    reference
      "foo";
  }

  identity network-type {
    description
      "Base type for Overlay network topology";
  }

  identity l3vpn {
```

```
    base network-type;
    description
        "Identity for layer3 vpn network type.";
}

identity l2vpn {
    base network-type;
    description
        "Identity for layer2 vpn network type.";
}

identity ospf {
    base network-type;
    description
        "Identity for OSPF network type.";
}

identity isis {
    base network-type;
    description
        "Identity for ISIS network type.";
}

identity node-type {
    description
        "Base identity for node type";
}

identity pe {
    base node-type;
    description
        "Identity for PE type";
}

identity ce {
    base node-type;
    description
        "Identity for CE type";
}

identity asbr {
    base node-type;
    description
        "Identity for ASBR type";
}

identity p {
    base node-type;
    description
```



```
    "Identity for P type";
}

identity link-type {
    description
        "Base identity for link type,e.g.,GRE, MPLS TE, VXLAN.";
}
identity gre {
    base link-type;
    description
        "Base identity for GRE Tunnel.";
}
identity VXLAN {
    base link-type;
    description
        "Base identity for VXLAN Tunnel.";
}
identity ip-in-ip {
    base link-type;
    description
        "Base identity for IP in IP Tunnel.";
}
identity direction {
    description
        "Base Identity for measurement direction including
        one way measurement and two way measurement.";
}

identity oneway {
    base direction;
    description
        "Identity for one way measurement.";
}

identity twoway {
    base direction;
    description
        "Identity for two way measurement.";
}
typedef percentage {
    type decimal64 {
        fraction-digits 5;
        range "0..100";
    }
    description
        "Percentage.";
}
```

```
grouping link-error-statistics {
  description
    "Grouping for per link error statistics";
  container loss-statistics {
    description
      "Per link loss statistics.";
    leaf direction {
      type identityref {
        base direction;
      }
      default "oneway";
      description
        "Define measurement direction including one way
        measurement and two way measurement.";
    }
    leaf packet-loss-count {
      type uint32 {
        range "0..4294967295";
      }
      default "0";
      description
        "Total received packet drops count.
        The value of count will be set to zero (0)
        on creation and will thereafter increase
        monotonically until it reaches a maximum value
        of 2^32-1 (4294967295 decimal), when it wraps
        around and starts increasing again from zero.";
    }
    leaf loss-ratio {
      type percentage;
      description
        "Loss ratio of the packets. Express as percentage
        of packets lost with respect to packets sent.";
    }
    leaf packet-reorder-count {
      type uint32 {
        range "0..4294967295";
      }
      default "0";
      description
        "Total received packet reordered count.
        The value of count will be set to zero (0)
        on creation and will thereafter increase
        monotonically until it reaches a maximum value
        of 2^32-1 (4294967295 decimal), when it wraps
        around and starts increasing again from zero.";
    }
    leaf packets-out-of-seq-count {
```

```
    type uint32 {
      range "0..4294967295";
    }
    description
      "Total received out of sequence count.
      The value of count will be set to zero (0)
      on creation and will thereafter increase
      monotonically until it reaches a maximum value
      of 2^32-1 (4294967295 decimal), when it wraps
      around and starts increasing again from zero..";
  }
  leaf packets-dup-count {
    type uint32 {
      range "0..4294967295";
    }
    description
      "Total received packet duplicates count.
      The value of count will be set to zero (0)
      on creation and will thereafter increase
      monotonically until it reaches a maximum value
      of 2^32-1 (4294967295 decimal), when it wraps
      around and starts increasing again from zero.";
  }
}

grouping link-delay-statistics {
  description
    "Grouping for per link delay statistics";
  container delay-statistics {
    description
      "Link delay summarised information. By default,
      one way measurement protocol (e.g., OWAMP) is used
      to measure delay.";
    leaf direction {
      type identityref {
        base direction;
      }
      default "oneway";
      description
        "Define measurement direction including one way
        measurement and two way measurement.";
    }
    leaf min-delay-value {
      type uint32;
      description
        "Minimum delay value observed.";
    }
  }
}
```

```
    leaf max-delay-value {
        type uint32;
        description
            "Maximum delay value observed.";
    }
    leaf average-delay-value {
        type uint32;
        description
            "Average delay is calculated on all the packets of a sample
            and is a simple computation to be performed for single marking method
            .";
    }
}

grouping link-jitter-statistics {
    description
        "Grouping for per link jitter statistics";
    container jitter-statistics {
        description
            "Link jitter summarised information. By default,
            jitter is measured using IP Packet Delay Variation
            (IPDV) as defined in RFC3393.";
        leaf direction {
            type identityref {
                base direction;
            }
            default "oneway";
            description
                "Define measurement direction including one way
                measurement and two way measurement.";
        }
        leaf min-jitter-value {
            type uint32;
            description
                "Minimum jitter value observed.";
        }
        leaf max-jitter-value {
            type uint32;
            description
                "Maximum jitter value observed.";
        }
        leaf average-jitter-value {
            type uint32;
            description
                "Average jitter is calculated on all the packets of a sample
                and is a simple computation to be performed for single marking method
                .";
        }
    }
}
```

```
}

grouping tp-svc-telemetry {

  leaf in-octets {
    type uint32;
    description
      "The total number of octets received on the
       interface, including framing characters.";
  }
  leaf inbound-unicast {
    type uint32;
    description
      "Inbound unicast packets were received, and delivered
       to a higher layer during the last period.";
  }
  leaf inbound-nunicast {
    type uint32;
    description
      "The number of non-unicast (i.e., subnetwork-
       broadcast or subnetwork-multicast) packets
       delivered to a higher-layer protocol.";
  }
  leaf inbound-discards {
    type uint32;
    description
      "The number of inbound packets which were chosen
       to be discarded even though no errors had been
       detected to prevent their being deliverable to a
       higher-layer protocol.";
  }
  leaf inbound-errors {
    type uint32;
    description
      "The number of inbound packets that contained
       errors preventing them from being deliverable to a
       higher-layer protocol.";
  }
  leaf inunknown-protos {
    type uint32;
    description
      "The number of packets received via the interface
       which were discarded because of an unknown or
       unsupported protocol";
  }
  leaf out-octets {
    type uint32;
    description
```

```
        "The total number of octets transmitted out of the
          interface, including framing characters";
    }
    leaf outbound-unicast {
        type uint32;
        description
            "The total number of packets that higher-level
             protocols requested be transmitted to a
             subnetwork-unicast address, including those that
             were discarded or not sent.";
    }
    leaf outbound-nunicast {
        type uint32;
        description
            "The total number of packets that higher-level
             protocols requested be transmitted to a non-
             unicast (i.e., a subnetwork-broadcast or
             subnetwork-multicast) address, including those
             that were discarded or not sent.";
    }
    leaf outbound-discards {
        type uint32;
        description
            "The number of outbound packets which were chosen
             to be discarded even though no errors had been
             detected to prevent their being transmitted. One
             possible reason for discarding such a packet could
             be to free up buffer space.";
    }
    leaf outbound-errors {
        type uint32;
        description
            "The number of outbound packets that contained
             errors preventing them from being deliverable to a
             higher-layer protocol.";
    }
    leaf outbound-qlen {
        type uint32;
        description
            " Length of the queue of the interface from where
             the packet is forwarded out. The queue depth could
             be the current number of memory buffers used by the
             queue and a packet can consume one or more memory buffers
             thus constituting device-level information.";
    }
    description
        "Grouping for interface service telemetry";
}
```

```
augment "/nw:networks/nw:network/nw:network-types" {
  description
    "Augment the network-types with service topology types";
  leaf-list network-technology-type {
    type identityref {
      base network-type;
    }
    description
      "Identify the network technology type, e.g., L3VPN, L2VPN, ISIS, OSPF.";
  }
}
augment "/nw:networks/nw:network" {
  description
    "Augment the network with service topology attributes";
  container overlay-topo-attributes {
    leaf vpn-topology {
      type identityref {
        base l3vpn-svc:vpn-topology;
      }
      description
        "VPN service topology, e.g. hub-spoke, any-to-any, hub-spoke-disjoint,
etc";
    }
    description
      "Container for vpn services";
  }
}
augment "/nw:networks/nw:network/nw:node" {
  description
    "Augment the network node with overlay topology attributes";
  container node-attributes {
    leaf node-type {
      type identityref {
        base node-type;
      }
      description
        "Node type, e.g. PE, P, ASBR, etc";
    }
    leaf site-id {
      type string;
      description
        "Asscoiated vpn site";
    }
    leaf site-role {
      type identityref {
        base l3vpn-svc:site-role;
      }
      default "l3vpn-svc:any-to-any-role";
      description

```

```
        "Role of the site in the VPN.";
    }
    description
        "Container for overlay topology attributes";
}
}
augment "/nw:networks/nw:network/nt:link" {
    description
        "Augment the network topology link with overlay topology attributes";
    leaf link-type {
        type identityref {
            base link-type;
        }
        description
            "Link type, e.g. GRE,VXLAN,IP in IP, etc";
    }
    container link-telemetry-attributes {
        config false;
        uses link-error-statistics;
        uses link-delay-statistics;
        uses link-jitter-statistics;
        description
            "Container for service telemetry attributes";
    }
}
augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
    description
        "Augment the network topology termination point with vpn service attribute
s";
    container tp-telemetry-attributes {
        config false;
        uses tp-svc-telemetry;
        description
            "Container for termination point service telemetry attributes.";
    }
}
}
}
<CODE ENDS>
```

## 10. Security Considerations

The YANG modules defined in this document MAY be accessed via the RESTCONF protocol [RFC8040] or NETCONF protocol ([RFC6241]). The lowest RESTCONF or NETCONF layer requires that the transport-layer protocol provides both data integrity and confidentiality, see Section 2 in [RFC8040] and [RFC6241]. 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 [RFC5246].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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

- o /nw:networks/nw:network/svc-topo:svc-telemetry-attributes
- o /nw:networks/nw:network/nw:node/svc-topo:node-attributes

## 11. IANA Considerations

This document registers a URI 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-network-vpn-pm

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 [RFC6020].

---

Name:	ietf-network-vpn-pm
Namespace:	urn:ietf:params:xml:ns:yang:ietf-network-vpn-pm
Prefix:	nvp
Reference:	RFC xxxx

---

## 12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [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>>.
- [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>>.
- [RFC6370] Bocci, M., Swallow, G., and E. Gray, "MPLS Transport Profile (MPLS-TP) Identifiers", RFC 6370, DOI 10.17487/RFC6370, September 2011, <<https://www.rfc-editor.org/info/rfc6370>>.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", RFC 6374, DOI 10.17487/RFC6374, September 2011, <<https://www.rfc-editor.org/info/rfc6374>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7923] Voit, E., Clemm, A., and A. Gonzalez Prieto, "Requirements for Subscription to YANG Datastores", RFC 7923, DOI 10.17487/RFC7923, June 2016, <<https://www.rfc-editor.org/info/rfc7923>>.

- [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>>.
- [RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <<https://www.rfc-editor.org/info/rfc7952>>.
- [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>>.
- [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>>.

## Authors' Addresses

Michael Wang  
Huawei Technologies, Co., Ltd  
101 Software Avenue, Yuhua District  
Nanjing 210012  
China

Email: wangzitao@huawei.com

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: bill.wu@huawei.com

Roni Even  
Huawei Technologies, Co., Ltd  
Tel Aviv  
Israel

Email: roni.even@huawei.com

Bin Wen  
Comcast

Email: bin\_wen@comcast.com

Change Liu  
China Unicom

Email: liuc131@chinaunicom.cn

Honglei Xu  
China Telecom

Email: xuhl.bri@chinatelecom.cn

CCAMP Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

H. Zheng  
I. Busi  
Huawei Technologies  
Y. Zheng  
China Unicom  
November 4, 2019

A YANG Data Model for Client Signal Performance Monitoring  
draft-zheng-ccamp-client-pm-yang-00

## Abstract

A transport network is a server-layer network to provide connectivity services to its client. Given the client signal is configured, the followup function for performance monitoring, such as latency and bit error rate, would be needed for network operation.

This document describes the data model to support the performance monitoring functionalities. The module carefully maps to relevant performance monitoring standards.

## 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 May 7, 2020.

## 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. Terminology and Notations . . . . .	3
3. Model Relationship . . . . .	3
4. Consideration on Monitoring Parameters . . . . .	4
5. YANG Model for Client Signal Performance Monitoring . . . . .	4
5.1. YANG Tree for Ethernet Performance Monitoring . . . . .	4
5.2. YANG Tree for Transparent Client Signal Performance Monitoring . . . . .	4
6. YANG Code for Performance Monitoring . . . . .	5
6.1. The ETH Service Performance Monitoring YANG Code . . . . .	5
6.2. The Transparent Client Signals Performance Monitoring YANG Code . . . . .	8
7. IANA Considerations . . . . .	11
8. Manageability Considerations . . . . .	11
9. Security Considerations . . . . .	12
10. Contributors . . . . .	12
11. References . . . . .	12
11.1. Normative References . . . . .	12
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

Client-layer network and server-layer network have been respectively modeled to allow the tunnels carrying the client traffic. Server-layers are modeled as tunnels with various switching technologies, such as [I-D.ietf-ccamp-otn-tunnel-model] and [I-D.ietf-ccamp-wson-tunnel-model]. Client-layers are modeled as client signals according to the client-signal identities specified in [I-D.ietf-ccamp-layer1-types].

In the network operation, the operator is interested in monitoring for their instantiated client signal over tunnels. The objective for such monitoring is to complete timely adjustment once there is abnormal statistic which may result in failure of the client signal. The parameters specified in the performance monitoring model can be collected for the operation need.

## 2. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this document is defined in [RFC8340]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "\*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

## 3. Model Relationship

[I-D.ietf-ccamp-client-signal-yang] has specified the two models for the client signal configuration, module ietf-trans-client-service for transparent client service and module ietf-eth-tran-service for Ethernet service. A common types module, ietf-eth-tran-types, has also been defined for the common use for service configuration. Basically the client signal types in this document is consistent with ietf-eth-tran-types, and focus on different functionality. On the perspective of operator, the modules in [I-D.ietf-ccamp-client-signal-yang] can be used to configure the service given any underlay tunnels, while the operation about monitoring the performance on given service can be achieved by using the model in this document.

Consideration on Key Performance Information (KPI) monitoring for Virtual Network (VN) and tunnels has been specified in [I-D.ietf-teas-actn-pm-telemetry-autonomics]. Usually the monitoring on the tunnels are the VNs should be separately deployed for the network operation, but it is possible to have common parameters that are both needed for the VN/TE and the configured services. Common types are imported in both modules.

VPN-level parameters and their monitoring have been defined in [I-D.www-bess-yang-vpn-service-pm]. This module focus on the performance on the topology at different layer or the overlay topology between VPN sites. On the other hand, this document is

focusing on the performance of the service configured between Customer Ends (CE), as described in [I-D.ietf-ccamp-client-signal-yang].

#### 4. Consideration on Monitoring Parameters

There can be multiple groups of parameters for monitoring, such as latency, bit error rate (BER). Some of these parameters are layer-dependent, for example, packet loss is only applicable in packet networks and won't be needed for layer 1 OTN and layer 0 WSON.

This document starts with the specification of the latency measurement for both Ethernet service and client signal service. In the future version additional parameters would be added into the data model in the same approach as the latency in the current version. A candidate list of parameters to be monitored include: Latency, Packet Loss, Bit Error Rate (BER), Jitter, Bandwidth, Byte/Packet number and so on.

#### 5. YANG Model for Client Signal Performance Monitoring

##### 5.1. YANG Tree for Ethernet Performance Monitoring

```
module: ietf-eth-service-pm
  +--rw performance-monitoring
    +--rw service-pm* [service-name]
      +--rw service-name          leafref
      +--rw pm-enable?            boolean
      +--rw latency-monitoring
        | +--rw latency-measure-enable?  boolean
      +--ro service-pm-state
        +--ro start-time?          yang:date-and-time
        +--ro last-update-time?    yang:date-and-time
        +--ro latency?             uint32
        +--ro error-message?       string
        +--ro service-oper-status? identityref
```

##### 5.2. YANG Tree for Transparent Client Signal Performance Monitoring



```
module: ietf-trans-client-svc-pm
  +--rw performance-monitoring
    +--rw service-pm* [service-name]
      +--rw service-name          leafref
      +--rw pm-enable?            boolean
      +--rw latency-monitoring
        | +--rw latency-measure-enable?  boolean
      +--ro service-pm-state
        +--ro start-time?          yang:date-and-time
        +--ro last-update-time?    yang:date-and-time
        +--ro latency?             uint32
        +--ro error-message?       string
        +--ro service-oper-status? identityref
```

## 6. YANG Code for Performance Monitoring

### 6.1. The ETH Service Performance Monitoring YANG Code

```
<CODE BEGINS> file "ietf-eth-service-pm@2019-11-04.yang"
module ietf-eth-service-pm {
  /* TODO: FIXME */
  yang-version 1.1;

  namespace "urn:ietf:params:xml:ns:yang:ietf-eth-service-pm";
  prefix "ethsvc-pm";

  import ietf-eth-tran-service {
    prefix "ethtsvc";
  }

  import ietf-eth-tran-types {
    prefix "eth-t-types";
  }

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

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

  organization
    "Internet Engineering Task Force (IETF) CCAMP WG";
```

contact

```
"
  WG List: <mailto:ccamp@ietf.org>

  ID-draft editor:
    Haomian Zheng (zhenghaomian@huawei.com);
    Italo Busi (italo.busi@huawei.com);
    Yanlei Zheng (zhengyanlei@chinaunicom.cn);
";
```

description

"This module defines the performance monitoring for Ethernet services. 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."

```
revision 2019-11-04 {
  description
    "Initial version";
  reference
    "ADD REFERENCE HERE";
}
```

```
container performance-monitoring {
  description
    "This part is for performance monitoring. ";
  list service-pm {
    key "service-name";
    description
      "The list of service to be monitored.";
    leaf service-name {
      type leafref {
        path "/ethtsvc:etht-svc/ethtsvc:etht-svc-instances/ethtsvc:etht-svc-na
me";
      }
      description "The name of service.";
    }
  }
}
```

```
leaf pm-enable {
  type boolean;
  description
    "Indicate whether the performance monitoring
    is enable or not.";
}

container latency-monitoring {
  description
    "To monitor the latency of service.";
  leaf latency-measure-enable {
    type boolean;
    description
      "Indicate whether the latency measurement
      is enable or not.";
  }
}

container service-pm-state {
  config false;
  description
    "The state of service performance monitoring.";
  leaf start-time {
    type yang:date-and-time;
    description
      "The time stamp when the service is started.";
  }

  leaf last-update-time {
    type yang:date-and-time;
    description
      "The time stamp when the service is last updated.";
  }

  leaf latency {
    type uint32;
    units microsecond;
    description
      "The latency of service.";
  }

  leaf error-message {
    type string;
    description
      "The message of error.";
  }

  leaf service-oper-status {
```

```
        type identityref {
            base te-types:tunnel-state-type;
        }
        description
            "The operational status of the services.";
    }
}
}
```

<CODE ENDS>

## 6.2. The Transparent Client Signals Performance Monitoring YANG Code

```
<CODE BEGINS> file "ietf-trans-client-svc-pm@2019-11-04.yang"
module ietf-trans-client-svc-pm {
    /* TODO: FIXME */
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-trans-client-svc-pm";
    prefix "clntsvc-pm";

    import ietf-trans-client-service {
        prefix "clntsvc";
    }

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

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

    organization
        "Internet Engineering Task Force (IETF) CCAMP WG";
    contact
        "
            WG List: <mailto:ccamp@ietf.org>

            ID-draft editor:
                Haomian Zheng (zhenghaomian@huawei.com);
                Italo Busi (italo.busi@huawei.com);
                Yanlei Zheng (zhengyanlei@chinaunicom.cn);
        ";
```

## description

"This module defines the performance monitoring for transparent client signals. 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."

revision 2019-11-04 {

description

"Initial version";

reference

"ADD REFERENCE HERE";

}

container performance-monitoring {

description

"This part is for performance monitoring. ";

list service-pm {

key "service-name";

description

"The list of service to be monitored.";

leaf service-name {

type leafref {

path "/clntsvc:client-svc/clntsvc:client-svc-instances/clntsvc:client-svc-name";

}

description "The name of service.";

}

leaf pm-enable {

type boolean;

description

"Indicate whether the performance monitoring is enable or not.";

}

container latency-monitoring {

description

"To monitor the latency of service.";

```
    leaf latency-measure-enable {
      type boolean;
      description
        "Indicate whether the latency measurement
         is enable or not.";
    }
  }

  container service-pm-state {
    config false;
    description
      "The state of service performance monitoring.";

    leaf start-time {
      type yang:date-and-time;
      description
        "The time stamp when the service is started.";
    }

    leaf last-update-time {
      type yang:date-and-time;
      description
        "The time stamp when the service is last updated.";
    }

    leaf latency {
      type uint32;
      units microsecond;
      description
        "The latency of service.";
    }

    leaf error-message {
      type string;
      description
        "The message of error.";
    }

    leaf service-oper-status {
      type identityref {
        base te-types:tunnel-state-type;
      }
      description
        "The operational status of the services.";
    }
  }
```

```
    }  
  }  
}
```

<CODE ENDS>

## 7. IANA Considerations

It is proposed that IANA should assign new URIs from the "IETF XML Registry" [RFC3688] as follows:

URI: urn:ietf:params:xml:ns:yang:ietf-eth-service-pm  
Registrant Contact: The IESG  
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-trans-client-svc-pm  
Registrant Contact: The IESG  
XML: N/A; the requested URI is an XML namespace.

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

name:	ietf-eth-service-pm
namespace:	urn:ietf:params:xml:ns:yang:ietf-eth-service-pm
prefix:	ethsvc-pm
reference:	RFC XXXX (This document)

name:	ietf-trans-client-svc-pm
namespace:	urn:ietf:params:xml:ns:yang:ietf-trans-client-svc-pm
prefix:	clntsvc-pm
reference:	RFC XXXX (This document)

## 8. Manageability Considerations

TBD.

## 9. Security Considerations

The data following the model defined in this document is exchanged via, for example, the interface between an orchestrator and a transport network controller. The security concerns mentioned in [I-D.ietf-ccamp-client-signal-yang] also applies to this document.

The YANG module defined in this document can be accessed via the RESTCONF protocol defined in [RFC8040], or maybe via the NETCONF protocol [RFC6241].

## 10. Contributors

Chaode YU  
Huawei Technologies,  
Email: yuchaode@huawei.com

## 11. References

### 11.1. Normative References

- [I-D.ietf-ccamp-client-signal-yang]  
Zheng, H., Guo, A., Busi, I., Snitser, A., Lazzeri, F., Xu, Y., Zhao, Y., Liu, X., and G. Fioccola, "A YANG Data Model for Transport Network Client Signals", draft-ietf-ccamp-client-signal-yang-01 (work in progress), November 2019.
- [I-D.ietf-ccamp-layer1-types]  
Zheng, H. and I. Busi, "A YANG Data Model for Layer 1 Types", draft-ietf-ccamp-layer1-types-03 (work in progress), November 2019.
- [I-D.ietf-teas-actn-pm-telemetry-autonomics]  
Lee, Y., Dhody, D., Karunanithi, S., Vilata, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", draft-ietf-teas-actn-pm-telemetry-autonomics-01 (work in progress), October 2019.
- [I-D.www-bess-yang-vpn-service-pm]  
Wang, Z., WU, Q., Even, R., Wen, B., Liu, C., and H. Xu, "A YANG Model for Network and VPN Service Performance Monitoring", draft-www-bess-yang-vpn-service-pm-04 (work in progress), November 2019.



- [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>>.
- [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>>.

## 11.2. Informative References

- [I-D.ietf-ccamp-otn-tunnel-model]  
Zheng, H., Busi, I., Belotti, S., Lopezalvarez, V., and Y. Xu, "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model-09 (work in progress), November 2019.
- [I-D.ietf-ccamp-wson-tunnel-model]  
Lee, Y., Zheng, H., Guo, A., Lopezalvarez, V., King, D., Yoon, B., and R. Vilata, "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model-04 (work in progress), September 2019.
- [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>>.

## Authors' Addresses

Haomian Zheng  
Huawei Technologies  
H1-1-A043S Huawei Industrial Base, Songshanhu  
Dongguan, Guangdong 523808  
China

Email: [zhenghaomian@huawei.com](mailto:zhenghaomian@huawei.com)

Italo Busi  
Huawei Technologies

Email: Italo.Busi@huawei.com

Yanlei Zheng  
China Unicom

Email: zhengyanlei@chinaunicom.cn