Network Working Group
Internet-Draft
Intended status: Informational
Expires: 26 August 2021

S. Barguil
L.M. Contreras
V. Lopez
O. Gonzalez de Dios
Telefonica

22 February 2021

Instantiation of IETF Network Slices in service providers networks
draft-barguil-teas-network-slices-instantation-00

Abstract

The IETF has produced several YANG data models to support the Software-Defined Networking and Network Slice Architecture. This document describes the relationship between the abstract (generic, or base) Service Models utilized for the Network Slices requests and the Network Models (e.g. L3NM, L2NM). This document describes the communication between the Network Slice Controller and a network controller for IETF network slice creation.

The YANG service models available for network slicing provide a customer-oriented view of the network. Thus, once the Network Slice controller (NSC)receives a request, it needs to expand it to accomplish the specific parameters expected by the network controller. The network models are analyzed in terms of how they can satisfy the IETF Network Slice requirements. Identified gaps on existing models are reported.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IETF has produced several YANG data models to support the Software-Defined Networking and Network Slice Architecture. This document describes the relationship between the abstract (generic, or base) Service Models utilized for the Network Slices requests and the Network Models (e.g. L3NM, L2NM, TE, etc). This document describes the communication between the Network Slice Controller and a network controller for IETF network slice creation.

The YANG service models available for network slicing provide a customer-oriented view of the network. Thus, once the Network Slice controller (NSC)receives a request, it needs to expand it to accomplish the specific parameters expected by the network controller. The network models are analyzed in terms of how they can satisfy the IETF Network Slice requirements. Identified gaps on existing models are reported.

Editor’s Note: the terminology in this draft will be aligned with the final terminology selected for describing the notion of IETF Network Slice when applied to IETF technologies, which is currently under discussion. By now same terminology as used in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework] is primarily used here. Consensus to use "IETF Network Slice" term has been reached.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Reference architecture

Several architectural definitions have arisen on the IETF to support SDN and network slicing deployments. The architectural proposal defined in [I-D.ietf-teas-ietf-network-slice-definition] includes a three-level hierarchy and expresses how each level relates with the ACTN architecture framework.

Figure 1 defines a sample architecture using those concepts. It starts from a top consumer or high-level operating system. Next, the network Slice Controller function is part of the Hierarchical network controller (e.g., as the MDSC in the ACTN context [RFC8453]) as a modular function. At the bottom, two network controllers, each one can handle multiple or single underlay technologies.

```
                  +------------------------------+
                  | High-level operation system. |
                  +--------------+---------------+
                                 |Slice Request
                                 |
             +-------------------v------------------+
             |                                      |
             |    Hierarchical Network              |
             |    Controller/Orchestrator           |
             |                                      |
             |   +------------------------------+   |
             |   |   Network Slice Controller   |   |
             |   +------------------------------+   |
             |                                      |
             +-------------------+------------------+
                                 |
                                 |
                  +--------------+---------------+
                  |                              |
                  v                              v
    +-------------+----------+     +-------------+----------+
    |   Network Controller   |     |   Network Controller   |
    +-------------+----------+     +-------------+----------+
                  |                              |
                  |                              |
                  v                              v
           Network Elements                Network Elements
```

Figure 1 Network Slice Controller as a module of the Hierarchical SDN controller.

In other implementations, the NSC can be a stand-alone element and directly interact with the network controller, as depicted in Figure 2. In this scenario, the services request follows a data-enrichment path, where each entity adds more information to the service request. This document describes how the available service models and network models interact to deliver the network slices in a service provider environment.

```
   +-------------------------------+
   | High-level operation system   |
   +--------------+----------------+
                  |Slice Request
                  |
    +-------------v-----------------+
    |   Network Slice Controller    |
    +-------------+-----------------+
                  |
                  |
                  |
    +-------------v-----------------+
    |       Network Controller      |
    +-------------+-----------------+
                  |
                  |
                  v
          Network Elements
```

Figure 2 Network Slice Controller as a stand-alone entity.

As another implementation possibility, the Network Slice Controller can be integrated with the Network controller and directly realize the network slice using device data models to configure the network devices. The sample architecture is depicted in Figure 4.

```
                  +
                  |Slice/VPN Request
                  |
    +-------------v-----------------+
    |      Network Controller       |
    |                               |
    |+-----------------------------+|
    ||   Network Slice Controller  ||
    |+-----------------------------+|
    |                               |
    +-------------+-----------------+
                  |
                  |
                  v
          Network Elements
```

Figure 3 Network Slice Controller as a module of the Network controller.

3. IETF Network Slice: requirements and data models

The main set of requirements for the IETF Slice, based on the high-level slice requirements from multiple organizations and use cases, are compiled in [I-D.contreras-teas-slice-nbi] and reproduced bellow for one of the slice use cases reported as example:

Network Slice Requeriments for 5G service
Availability
Deterministic communication
Downlink throughput per network slice
Energy efficiency
Group communication support
Isolation level
Maximum supported packet size
Mission critical support
Performance monitoring
Slice quality of service parameters
Support for non-IP traffic
Uplink throughput per network slice
User data access (i.e., tunneling mechanisms)

TODO#1: Summarize the requirements based on the different slice use cases described in [I-D.contreras-teas-slice-nbi].

To accomplish those requirements, a set of YANG data models have been proposed. Those Yang models , summarized in table xx, could be used by an IETF Network Slice Controller to manage CRUD operations on the IETF Network Slice. That is, these models aim capturing the requirements from the consumer of the slice point of view and avoid entering into the detail of how the slice is actually created.

* [draft-wd-teas-ietf-network-slice-nbi-yang-01]: A Yang Data Model for IETF Network Slice NBI.

* [draft-liu-teas-transport-network-slice-yang-00]: Transport Network Slice YANG Data Model.

4. Yang Models for Network Controllers

A network controller, understood as the entity responsible for managing a particular network domain, can expose a northbound interface based on YANG models. That is, those YANG models will define datastores that apply for a whole network domain and will manage network-level concepts. The types of network models that are of interest for the instantiation of IETF Network slices are:

* LxVPN Network models:

  - These models describe a VPN service from the network point of view.

* Traffic Engineering models:

  - These models allow to manipulate Traffic Engineering tunnels within the network segment. Technology-specific extensions allow to work with a desired technology (e.g. MPLS RSVP-TE tunnels, Segment Routing paths, OTN tunnels, etc.)

* TE Service Mapping extensions:

  - These extensions allow to specify for LxVPN the details of an underlay based on TE.

* ACLs and routing policies models:

  - Even though ACLs and routing policies are device models, it’s exposure in the NBI of a domain controller allows to provide an additional granularity that the network domain controller is not able to infer on its own.

4.1. LxVPN Network Models

The framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The data models can be used during the service and network management life cycle (e.g., service instantiation, service provisioning, service optimization, service monitoring, service diagnosing, and service assurance). The so called Network models could be reused for the realization of Network slice requests.

The following models are examples of Network models that describe services.

* [I-D.ietf-opsawg-l3sm-l3nm]: A Layer 3 VPN Network YANG Model

- [I-D.ietf-opsawg-l2nm]: A Layer 2 VPN Network YANG Model

4.2. Traffic Engineering Models

TEAS has defined a collection of models to allow the management of Traffic Engineering tunnels.

- [I-D.ietf-teas-yang-te]: A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces. The model allows to instantiate paths in a TE enabled network. Note that technology augmented models are require to particular per-technology instantiations.

4.3. Traffic Engineering Service Mapping

The IETF has defined a YANG model to set up the procedure to map VPN service/network models to the TE models. This model, known as service mapping, allows the network controller to assign/retrieve transport resources allocated to specific services. At the moment there is just one service mapping model [I-D.ietf-teas-te-service-mapping-yang]. The "Traffic Engineering (TE) and Service Mapping Yang Model" augments the VPN service and network models.

5. Compliance of Network Controller models with IETF Network slice requirements.

Section 3 presented the requirements of the IETF Network slice. In this subsection it is analyzed how available YANG models that can be used by a Network Controller can satisfy those requirements and identify gaps.

5.1. Availability

As per [draft-ietf-teas-te-service-mapping-yang-05], Availability is a probabilistic measure of the length of time that a VPN/VN instance functions without a network failure. As per RFC 8330, The parameter "availability", as described in [G.827], [F.1703], and [P.530], is often used to describe the link capacity. The availability is a time scale, representing a proportion of the operating time that the requested bandwidth is ensured".

The calculation of the availability is not trivial and would need to be clearly scoped to avoid misunderstandings.

The set of Yang models proposed today allow to request tunnels/paths with different resiliency requirements in terms of protection and restoration. However, none of them include the possibility of requesting a specific availability (e.g. 99.9999%).

5.2. Downlink throughput / Uplink throughput.

The LxVPN Models allow to specify the bandwdidth at the interface level between the slice and the customer. In addition, the TE models allow to force a give bandwidth in the connection between Provider Edges.

6. Interactions

6.1. Slice requested to Hierarchical Network Controller

When the Network Slice Controller is a Hierarchical SDN controller module, the NSC’s and the Hierarchical Network Controller should share the same internal data and the same NBI. Thus, to process the customer, view the H-SDN module must be able to:

* _Map_: The customer request received using the [draft-wd-teas-ietf-network-slice-nbi-yang-01] must be processed by the NCS. The mapping process takes the network-slice SLAs selected by the customer to available Routing Policies and Forwarding policies.

* _Realize_: Create necessary network requests. The slice’s realization can be translated into one or several LXNM Network requests, depending on the number of underlay controllers. Thus, the NCS must have a complete view of the network to map the orders and distribute them across domains. The realization should include the expansion/selection of Forwarding Policies, Routing Policies, VPN policies, and Underlay transport preference.

To maintain the data coherence between the control layers, the "network-slice-id" used of the [draft-wd-teas-ietf-network-slice-nbi-yang-01] must be directly mapped to the ‘transport-instance-id at the VPN-Node level.

```
                                 +
                                 |
                                 | Slice Request:
                     draft-wd-teas-ietf-network-slice-nbi-yang-01
                                 | * network-slice-id
                                 |
             +-------------------v------------------+
             |                                      |
             |    Hierarchical Network              |
             |    Controller/Orchestrator           |
             |                                      |
             |   +------------------------------+   |
             |   |   Network Slice Controller   |   |
             |   +------------------------------+   |
             |                                      |
             +-------------------+------------------+
         Slice Realizer: LXNM    |
           VPN-id                |
   * transport-instance-id
                                 |
                  +--------------+---------------+
                  |                              |
                  v                              v
    +-------------+----------+     +-------------+----------+
    |   Network Controller   |     |   Network Controller   |
    +-------------+----------+     +-------------+----------+
                  |                              |
                  |                              |
                  v                              v
           Network Elements                Network Elements
```

Figure 4 Workflow for the slice request in an integrated architecture.

6.2. Slice requested to Network Slice Controller

When the Network Slice Controller is a stand-alone controller module, the NSC's should perform the same two tasks described before:

- _Map_: Process the customer request. The customer request can be sent using the [draft-liu-teas-transport-network-slice-yang-01]. This draft allows the topology mapping of the Slice request.

- _Realize_: Create necessary network requests. The slice's realization will be translated into one LXNM Network request. As the NCS has a topological view of the network, the realization can include the customer's traffic engineering transport preferences and policies.

```
               +
               |Slice Request
draft-liu-teas-transport-network-slice-yang-01
network-id
               |
 +-------------v----------------+
 |   Network Slice Controller   |
 +-------------+----------------+
               |
Slice Realizer: LXNM
  VPN-id       |
  * Underlay-transport
  * transport-instance-id
               |
 +-------------v----------------+
 |       Network Controller     |
 +-------------+----------------+
               |
               |
               v
       Network Elements
```

Figure 5 Workflow for the slice request in an stand-alone architecture.

TODO#2: Include description for the scenario in Figure 3.

7. Security Considerations

There are two main aspects to consider. On the one hand, the IETF Network Slice has a set of security related requirements, such as hard isolation of the slice, or encryption of the communications through the slice. All those requirements need to be analyzed in detailed and cleary mapped to the Network Controller and device interfaces. On the other hand, the communication between the IETF network slicer and the network controller (or controllers or hierarchy of controllers) need to follow the same security considerations as with the network models. The network YANG modules defines schemas 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 [RFC8466]. The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The following summarizes the foreseen risks of using the Network Models to instantiate IETF network Slices: - Malicious clients attempting to delete or modify VPN services that implements an IETF network slice. The malicious client could manipulate security related aspects of the network configuration that impact the requirements of the slice, failing to satisfy the customer requirement. - Unauthorized clients attempting to create/modify/delete a VPN hat implements an IETF network slice service. - Unauthorized clients attempting to read VPN services related information hat implements an IETF network slice - Malicious clients attempting to leak traffic of the slice.

8. IANA Considerations

This document is informational and does not require IANA allocations.

9. Conclusions

A wide variety of yang models are currently under definition in IETF that can be used by Network Controllers to instantiate IETF network slices. Some of the IETF slice requirements can be satisfied by multiple means, as there are multiple choices avaialable. However, other requirements are still not covered by the existing models. A more detailed definition of those uncovered requirements would be needed. Finally a consensus on the set of models to be exposed by Network Controllers would facilitate the deployment of IETF network slices.

10. Normative References

[I-D.contreras-teas-slice-nbi]
Contreras, L., Homma, S., and J. Ordonez-Lucena, "IETF Network Slice use cases and attributes for Northbound Interface of controller", Work in Progress, Internet-Draft, draft-contreras-teas-slice-nbi-03, 30 October 2020, <https://tools.ietf.org/html/draft-contreras-teas-slice-nbi-03>.

[I-D.ietf-opsawg-l2nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-01, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-opsawg-l2nm-01>.

[I-D.ietf-opsawg-l3sm-l3nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", Work in

Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-05, 16 October 2020, <https://tools.ietf.org/html/draft-ietf-opsawg-l3sm-l3nm-05>.

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-definition-00, 25 January 2021, <https://tools.ietf.org/html/draft-ietf-teas-ietf-network-slice-definition-00>.

[I-D.ietf-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", Work in Progress, Internet-Draft, draft-ietf-teas-te-service-mapping-yang-05, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-teas-te-service-mapping-yang-05>.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-25, 27 July 2020, <https://tools.ietf.org/html/draft-ietf-teas-yang-te-25>.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", Work in Progress, Internet-Draft, draft-nsdt-teas-ns-framework-04, 13 July 2020, <https://tools.ietf.org/html/draft-nsdt-teas-ns-framework-04>.

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

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

[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>.

[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>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/info/rfc8969>.

Authors' Addresses

Samier Barguil
Telefonica
Distrito T
Madrid

Email: samier.barguilgiraldo.ext@telefonica.com

Luis Miguel Contreras
Telefonica
Distrito T
Madrid

Email: luismiguel.contrerasmurillo@telefonica.com

Victor Lopez
Telefonica
Distrito T
Madrid

Email: victor.lopezalvarez@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Distrito T
Madrid

Email: oscar.gonzalezdedios@telefonica.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

T. Saad
V. Beeram
Juniper Networks
B. Wen
Comcast
D. Ceccarelli
J. Halpern
Ericsson
S. Peng
R. Chen
ZTE Corporation
X. Liu
Volta Networks
L. Contreras
Telefonica

February 22, 2021

Realizing Network Slices in IP/MPLS Networks
draft-bestbar-teas-ns-packet-02

Abstract

Network slicing provides the ability to partition a physical network into multiple logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers. Network slices need to operate in parallel while providing slice elasticity in terms of network resource allocation. The Differentiated Service (Diffserv) model allows for carrying multiple services on top of a single physical network by relying on compliant nodes to apply specific forwarding treatment (scheduling and drop policy) on to packets that carry the respective Diffserv code point. This document proposes a solution based on the Diffserv model to realize network slicing in IP/MPLS networks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Network slicing allows a Service Provider to create independent and logical networks on top of a common or shared physical network infrastructure. Such network slices can be offered to customers or used internally by the Service Provider to facilitate or enhance their service offerings. A Service Provider can also use network slicing to structure and organize the elements of its infrastructure. This document provides a path control technology agnostic solution that a Service Provider can deploy to realize network slicing in IP/MPLS networks.

The definition of network slice for use within the IETF and the characteristics of IETF network slice are specified in [I-D.ietf-teas-ietf-network-slice-definition]. A framework for reusing IETF VPN and traffic-engineering technologies to realize IETF network slices is discussed in [I-D.nsdt-teas-ns-framework]. These documents also discuss the function of an IETF Network Slice Controller and the requirements on its northbound and southbound interfaces.

This document introduces the notion of a slice aggregate which comprises of one of more IETF network slice traffic streams. It describes how a slice policy can be used to realize a slice aggregate by instantiating specific control and data plane behaviors on select topological elements in IP/MPLS networks. The onus is on the IETF Network Slice Controller to maintain the mapping between one or more IETF network slices and a slice aggregate. The mechanisms used by the controller to determine the mapping are outside the scope of this document. The focus of this document is on the mechanisms required at the device level to address the requirements of network slicing in packet networks.

In a Differentiated Service (Diffserv) domain [RFC2475], packets requiring the same forwarding treatment (scheduling and drop policy) are classified and marked with a Class Selector (CS) at domain ingress nodes. At transit nodes, the CS field inside the packet is inspected to determine the specific forwarding treatment to be applied before the packet is forwarded further. Similar principles are adopted by this document to realize network slicing.

When logical networks representing slice aggregates are realized on top of a shared physical network infrastructure, it is important to steer traffic on the specific network resources allocated for the slice aggregate. In packet networks, the packets that traverse a specific slice aggregate MAY be identified by one or more specific fields carried within the packet. A slice policy ingress boundary node populates the respective field(s) in packets that enter a slice aggregate to allow interior slice policy nodes to identity those packets and apply the specific Per Hop Behavior (PHB) that is associated with the slice aggregate. The PHB defines the scheduling treatment and, in some cases, the packet drop probability.

The slice aggregate traffic may further carry a Diffserv CS to allow differentiation of forwarding treatments for packets within a slice aggregate. For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same slice aggregate by carrying a global MPLS label in the label stack that identifies the slice aggregate in each packet. Additional Diffserv classification may be indicated in the Traffic Class (TC) bits of the global MPLS label to allow further differentiation of forwarding treatments for traffic traversing the same slice aggregate network resources.

This document covers different modes of slice policy and discusses how each slice policy mode can ensure proper placement of slice aggregate paths and respective treatment of slice aggregate traffic.

1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework].

The following terminology is used in the document:

IETF network slice:
a well-defined composite of a set of endpoints, the connectivity requirements between subsets of these endpoints, and associated requirements; the term 'network slice' in this document refers to 'IETF network slice' [I-D.ietf-teas-ietf-network-slice-definition].

IETF Network Slice Controller (NSC):
controller that is used to realize an IETF network slice [I-D.ietf-teas-ietf-network-slice-definition].

Slice policy:
a policy construct that enables instantiation of mechanisms in support of IETF network slice specific control and data plane

behaviors on select topological elements; the enforcement of a slice policy results in the creation of a slice aggregate.

Slice aggregate:
a collection of packets that match a slice policy selection criteria and are given the same forwarding treatment; a slice aggregate comprises of one or more IETF network slice traffic streams; the mapping of one or more IETF network slices to a slice aggregate is maintained by the IETF Network Slice Controller.

Slice policy capable node:
a node that supports one of the slice policy modes described in this document.

Slice policy incapable node:
a node that does not support any of the slice policy modes described in this document.

Slice aggregate traffic:
traffic that is forwarded over network resources associated with a specific slice aggregate.

Slice aggregate path:
a path that is setup over network resources associated with a specific slice aggregate.

Slice aggregate packet:
a packet that traverses network resources associated with a specific slice aggregate.

Slice policy topology:
a set of topological elements associated with a slice policy.

Slice aggregate aware TE:
a mechanism for TE path selection that takes into account the available network resources associated with a specific slice aggregate.

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

1.2. Acronyms and Abbreviations

BA: Behavior Aggregate

CS: Class Selector

SS: Slice Selector

S-PHB: Slice policy Per Hop Behavior as described in Section 5.1.3

SSL: Slice Selector Label as described in section Section 5.1.1

SSLI: Slice Selector Label Indicator

SLA: Service Level Agreement

SLO: Service Level Objective

Diffserv: Differentiated Services

MPLS: Multiprotocol Label Switching

LSP: Label Switched Path

RSVP: Resource Reservation Protocol

TE: Traffic Engineering

SR: Segment Routing

VRF: VPN Routing and Forwarding

2. Network Resource Slicing Membership

A slice aggregate can span multiple parts of an IP/MPLS network (e.g., all or specific network resources in the access, aggregation, or core network), and can stretch across multiple domains administered by a provider. A slice policy topology may include all or a sub-set of the physical nodes and links of an IP/MPLS network; it may be comprised of dedicated and/or shared network resources (e.g., in terms of processing power, storage, and bandwidth).

2.1. Dedicated Network Resources

Physical network resources may be fully dedicated to a specific slice aggregate. For example, traffic belonging to a slice aggregate can traverse dedicated network resources without being subjected to contention from traffic of other slice aggregates. Dedicated network resource slicing allows for simple partitioning of the physical network resources amongst slice aggregates without the need to distinguish packets traversing the dedicated network resources since only one slice aggregate traffic stream can traverse the dedicated resource at any time.

2.2. Shared Network Resources

To optimize network utilization, sharing of the physical network resources may be desirable. In such case, the same physical network resource capacity is divided among multiple slice aggregates. Shared network resources can be partitioned in the data plane (for example by applying hardware policers and shapers) and/or partitioned in the control plane by providing a logical representation of the physical link that has a subset of the network resources available to it.

3. Path Selection

Path selection in a network can be network state dependent, or network state independent as described in Section 5.1 of [I-D.ietf-teas-rfc3272bis]. The latter is the choice commonly used by IGPs when selecting a best path to a destination prefix, while the former is used by ingress TE routers, or Path Computation Engines (PCEs) when optimizing the placement of a flow based on the current network resource utilization.

For example, when steering traffic on a delay optimized path, the IGP can use its link state database’s view of the network topology to compute a path optimizing for the delay metric of each link in the network resulting in a cumulative lowest delay path.

When path selection is network state dependent, the path computation can leverage Traffic Engineering mechanisms (e.g., as defined in [RFC2702]) to compute feasible paths taking into account the incoming traffic demand rate and current state of network. This allows avoiding overly utilized links, and reduces the chance of congestion on traversed links.

To enable TE path placement, the link state is advertised with current reservations, thereby reflecting the available bandwidth on each link. Such link reservations may be maintained centrally on a network wide network resource manager, or distributed on devices (as usually done with RSVP). TE extensions exist today to allow IGPs (e.g., [RFC3630] and [RFC5305]), and BGP-LS [RFC7752] to advertise such link state reservations.

When network resource reservations are also slice aggregate aware, the link state can carry per slice aggregate state (e.g., reservable bandwidth). This allows path computation to take into account the specific network resources available for a slice aggregate when determining the path for a specific flow. In this case, we refer to the process of path placement and path provisioning as slice aggregate aware TE.

4. Slice Policy Modes

A slice policy can be used to dictate if the partitioning of the shared network resources amongst multiple slice aggregates can be achieved by realizing slice aggregates in:

a) data plane only, or

b) control plane only, or

c) both control and data planes.

4.1. Data plane Slice Policy Mode

The physical network resources can be partitioned on network devices by applying a Per Hop forwarding Behavior (PHB) onto packets that traverse the network devices. In the Diffserv model, a Class Selector (CS) is carried in the packet and is used by transit nodes to apply the PHB that determines the scheduling treatment and drop probability for packets.

When data plane slice policy mode is applied, packets need to be forwarded on the specific slice aggregate network resources and need to be applied a specific forwarding treatment that is dictated in the slice policy (refer to Section 5.1 below). A Slice Selector (SS) MUST be carried in each packet to identify the slice aggregate that it belongs to.

The ingress node of a slice policy domain, in addition to marking packets with a Diffserv CS, MAY also add an SS to each slice aggregate packet. The transit nodes within a slice policy domain MAY use the SS to associate packets with a slice aggregate and to determine the Slice policy Per Hop Behavior (S-PHB) that is applied to the packet (refer to Section 5.1.3 for further details). The CS MAY be used to apply a Diffserv PHB on to the packet to allow differentiation of traffic treatment within the same slice aggregate.

When data plane only slice policy mode is used, routers may rely on a network state independent view of the topology to determine the best paths to reach destinations. In this case, the best path selection dictates the forwarding path of packets to the destination. The SS field carried in each packet determines the specific S-PHB treatment along the selected path.

For example, the Segment-Routing Flexible Algorithm [I-D.ietf-lsr-flex-algo] may be deployed in a network to steer packets on the IGP computed lowest cumulative delay path. A slice policy may be used to allow links along the least latency path to share its data plane resources amongst multiple slice aggregates. In this case, the packets that are steered on a specific slice policy carry the SS field that enables routers (along with the Diffserv CS) to determine the S-PHB and enforce slice aggregate traffic streams.

4.2. Control Plane Slice Policy Mode

The physical network resources in the network can be logically partitioned by having a representation of network resources appear in a virtual topology. The virtual topology can contain all or a subset of the physical network resources. The logical network resources that appear in the virtual topology can reflect a part, whole, or in-excess of the physical network resource capacity (when oversubscription is desirable). For example, a physical link bandwidth can be divided into fractions, each dedicated to a slice aggregate. Each fraction of the physical link bandwidth MAY be represented as a logical link in a virtual topology that is used when determining paths associated with a specific slice aggregate. The virtual topology associated with the slice policy can be used by routing protocols, or by the ingress/PCE when computing slice aggregate aware TE paths.

To perform network state dependent path computation in this mode (slice aggregate aware TE), the resource reservation on each link needs to be slice aggregate aware. Multiple slice policies may be applied on the same physical link. The slice aggregate network resource availability on links is updated (and may eventually be advertised in the network) when new paths are placed in the network. The slice aggregate resource reservation, in this case, can be maintained on each device or be centralized on a resource reservation manager that holds reservation states on links in the network.

Multiple slice aggregates can form a group and share the available network resources allocated to each slice aggregate. In this case, a node can update the reservable bandwidth for each slice aggregate to take into consideration the available bandwidth from other slice aggregates in the same group.

For illustration purposes, the diagram below represents bandwidth isolation or sharing amongst a group of slice aggregates. In Figure 1a, the slice aggregates: S_AGG1, S_AGG2, S_AGG3 and S_AGG4 are not sharing any bandwidths between each other. In Figure 1b, the slice aggregates: S_AGG1 and S_AGG2 can share the available bandwidth portion allocated to each amongst them. Similarly, S_AGG3 and S_AGG4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to S_AGG1 or S_AGG2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.

```
   I-----------------------------I       I-----------------------------I
   <--S_AGG1->                   I       I-----------------I           I
   I---------I                   I       I <-S_AGG1->      I           I
   I         I                   I       I I-------I       I           I
   I---------I                   I       I I       I       I           I
   I                             I       I I-------I       I           I
   <-----S_AGG2------>           I       I                 I           I
   I-----------------I           I       I <-S_AGG2->      I           I
   I                 I           I       I I---------I     I           I
   I-----------------I           I       I I         I     I           I
   I                             I       I I---------I     I           I
   <---S_AGG3---->               I       I                 I           I
   I-------------I               I       I S_AGG1 + S_AGG2 I           I
   I             I               I       I-----------------I           I
   I-------------I               I       I                             I
   I                             I       I                             I
   <---S_AGG4---->               I       I-----------------I           I
   I-------------I               I       I <-S_AGG3->      I           I
   I             I               I       I I-------I       I           I
   I-------------I               I       I I       I       I           I
   I                             I       I I-------I       I           I
   I S_AGG1+S_AGG2+S_AGG3+S_AGG4 I       I                 I           I
   I                             I       I <-S_AGG4->      I           I
   I-----------------------------I       I I---------I     I           I
   <--Max Reservable Bandwidth-->        I I         I     I           I
                                         I I---------I     I           I
                                         I                 I           I
                                         I S_AGG3 + S_AGG4 I           I
                                         I-----------------I           I
                                         I S_AGG1+S_AGG2+S_AGG3+S_AGG4 I
                                         I                             I
                                         I-----------------------------I
                                         <--Max Reservable Bandwidth-->

   (a) No bandwidth sharing              (b) Sharing bandwidth between
       between slice aggregates.             slice aggregates of the
                                             same group
```

Figure 1: Bandwidth Isolation/Sharing.

4.3. Data and Control Plane Slice Policy Mode

In order to support strict guarantees for slice aggregates, the network resources can be partitioned in both the control plane and data plane.

The control plane partitioning allows the creation of customized topologies per slice aggregate that routers or a Path Computation Engine (PCE) can use to determine optimal path placement for specific demand flows (Slice aggregate aware TE).

The data plane partitioning protects slice aggregate traffic from network resource contention that could occur due to bursts in traffic from other slice aggregates traversing the same shared network resource.

5. Slice Policy Instantiation

A network slice can span multiple technologies and multiple administrative domains. Depending on the network slice consumer’s requirements, a network slice can be differentiated from other network slices in terms of data, control or management planes.

The consumer of a network slice expresses their intent by specifying requirements rather than mechanisms to realize the slice. The requirements for a network slice can vary and can be expressed in terms of connectivity needs between end-points (point-to-point, point-to-multipoint or multipoint-to-multipoint) with customizable network capabilities that may include data speed, quality, latency, reliability, security, and services (refer to [I-D.ietf-teas-ietf-network-slice-definition] for more details). These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice consumer and the provider.

The onus is on the network slice controller to consume the service layer slice intent and realize it with an appropriate slice policy. Multiple IETF network slices can be mapped to the same slice policy resulting in a slice aggregate. The network wide consistent slice policy definition is distributed to the devices in the network as shown in Figure 2. The specification of the network slice intent on the northbound interface of the controller and the mechanism used to map the network slice to a slice policy are outside the scope of this document.

```
                                  |
                                  | IETF Network Slice
                                  | (service)
                       +--------------------+
                       |    IETF Network    |
                       |  Slice Controller  |
                       +--------------------+
                                  |
                                  | Slice Policy
                                 /|\
                                / | \
                         slice policy capable
                           nodes/controllers
                           / /    |    \ \
                          v v     v     v v
                         xxxxxxxxxxxxxxxxxxxx
                       xxxx                xxxx
                     xxxx       Slice        xxxx
                     xxxx     Aggregate      xxxx
                       xxxx                xxxx
                         xxxxxxxxxxxxxxxxxxxx

                     <------ Path Control ------>
                     RSVP-TE/SR-Policy/SR-FlexAlgo
```

Figure 2: Slice Policy Instantiation.

5.1. Slice Policy Definition

The slice policy is network-wide construct that is consumed by network devices, and may include rules that control the following:

- Data plane specific policies: This includes the SS, any firewall rules or flow-spec filters, and QoS profiles associated with the slice policy and any classes within it.

- Control plane specific policies: This includes guaranteed bandwidth, any network resource sharing amongst slice policies, and reservation preference to prioritize any reservations of a specific slice policy over others.

- Topology membership policies: This defines policies that dictate node/link/function network resource topology association for a specific slice policy.

There is a desire for flexibility in realizing network slices to support the services across networks consisting of products from multiple vendors. These networks may also be grouped into disparate domains and deploy various path control technologies and tunnel techniques to carry traffic across the network. It is expected that a standardized data model for slice policy will facilitate the instantiation and management of slice aggregates on slice policy capable nodes.

It is also possible to distribute the slice policy to network devices using several mechanisms, including protocols such as NETCONF or RESTCONF, or exchanging it using a suitable routing protocol that network devices participate in (such as IGP(s) or BGP).

5.1.1. Slice Policy Data Plane Selector

A router MUST be able to identify a packet belonging to a slice aggregate before it can apply the proper forwarding treatment or S-PHB associated with the slice policy. One or more fields within the packet MAY be used as an SS to do this.

Forwarding Address Slice Selector:

One approach to distinguish packets targeted to a destination but belonging to different slice aggregates is to assign multiple forwarding addresses (or multiple MPLS label bindings in the case of MPLS network) for the same node - one for each slice aggregate that traffic can be steered on towards the destination. For example, when realizing a network slice over an IP dataplane, the same destination can be assigned multiple IP addresses (or multiple SRv6 locators in the case of SRv6 network) to enable steering of traffic to the same destination over multiple slice policies.

Similarly, for MPLS dataplane, [RFC3031] states in Section 2.1 that: 'Some routers analyze a packet's network layer header not merely to choose the packet's next hop, but also to determine a packet's "precedence" or "class of service"'. In such case, the same destination can be assigned multiple MPLS label bindings corresponding to an LSP that traverses network resources of a specific slice aggregate towards the destination.

The slice aggregate specific forwarding address (or MPLS forwarding label) can be carried in the packet to allow (IP or MPLS) routers along the path to identify the packets and apply the respective S-PHB and forwarding treatment. This approach requires maintaining per slice aggregate state for each destination in the network in both the control and data plane and on each router in the network.

For example, consider a network slicing provider with a network composed of 'N' nodes, each with 'K' adjacencies to its neighbors. Assuming a node is reachable in as many as 'M' slice policies, the node will have to assign and advertise reachability for 'N' unique forwarding addresses, or MPLS forwarding labels. Similarly, each node will have to assign a unique forwarding address (or MPLS forwarding label) for each of its 'K' adjacencies to enable strict steering over each. Consequently, the control plane at any node in the network will need to store as many as (N+K)*M states. In addition, a node will have to store and program (N+K)*M forwarding addresses or labels entries in its Forwarding Information Base (FIB) to realize this. Therefore, as 'N', 'K', and 'M' parameters increase, this approach will have scalability challenges both in the control and data planes.

Global Identifier Slice Selector:

A slice policy can include a global Slice Selector (SS) field can be carried in each packet to identify the packet belonging to a specific slice aggregate, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the slice policy domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding path, and use the SS field in the packet to determine the specific S-PHB that gets applied on the packet. This approach allows better scale since it relies on a single forwarding address or MPLS label binding to be used independent of the number of slice policies required along the path. In this case, the additional SS field will need to be carried, and maintained in each packet while it traverses the slice policy domain.

The SS can be carried in one of multiple fields within the packet, depending on the dataplane type used. For example, in MPLS networks, the SS can be represented as a global MPLS label that is carried in the packet's MPLS label stack. All packets that belong to the same slice aggregate MAY carry the same SS label in the MPLS label stack. It is possible, as well, to have multiple SS labels that map to the same slice policy S-PHB.

The MPLS SS Label (SSL) may appear in several positions in the MPLS label stack. For example, the MPLS SSL can be maintained at the top of the label stack while the packet is forwarded along the MPLS path. In this case, the forwarding at each hop is determined by the forwarding label that resides below the SSL. Figure 3 shows an example where the SSL appears at the top of MPLS label stack in a packet. PE1 is a slice policy edge node that receives the packet that needs to be steered over a slice specific MPLS Path. PE1 computes the SR Path composed of the Label Segment-

List={9012, 9023}. It imposes an SSL 1001 corresponding to Slice-ID 1001 followed by the SR Path Segment-List. At P1, the top label sets the context of the packet to Slice-ID=1001. The forwarding of the packet is determined by inspecting the forwarding label (below the SSL) within the context of SSL.

```
     SR Adj-SID:           SSL: 1001
        9012: P1-P2
        9023: P2-PE2

            /-----\        /-----\         /-----\       /-----\
            | PE1 | -----  | P1  | ------ | P2  |------ | PE2 |
            \-----/        \-----/         \-----/       \-----/

   In
   packet:
   +------+       +------+         +------+        +------+
   | IP   |       | 1001 |         | 1001 |        | 1001 |
   +------+       +------+         +------+        +------+
   | Pay- |       | 9012 |         | 9023 |        | IP   |
   | Load |       +------+         +------+        +------+
   +----- +       | 9023 |         | IP   |        | Pay- |
                  +------+         +------+        | Load |
                  | IP   |         | Pay- |        +------+
                  +------+         | Load |
                  | Pay- |         +------+
                  | Load |
                  +------+
```

Figure 3: SSL at top of label stack.

The SSL can also reside at the bottom of the label stack. For example, the VPN service label may also be used as an SSL which allows steering of traffic towards one or more egress PEs over the same slice aggregate. In such cases, one or more service labels MAY be mapped to the same slice aggregate. The same VPN label may also be allocated on all Egress PEs so it can serve as a single SSL for a specific slice policy. Alternatively, a range of VPN labels may be mapped to a single slice aggregate to allow carrying multiple VPNs over the same slice aggregate as shown in Figure 4.

```
     SR Adj-SID:          SSL (VPN) on PE2: 1001
        9012: P1-P2
        9023: P2-PE2

            /-----\        /-----\        /-----\       /-----\
            | PE1 | -----  | P1  | ------ | P2  |------ | PE2 |
            \-----/        \-----/        \-----/       \-----/

   In
   packet:
   +------+       +------+         +------+        +------+
   | IP   |       | 9012 |         | 9023 |        | 1001 |
   +------+       +------+         +------+        +------+
   | Pay- |       | 9023 |         | 1001 |        | IP   |
   | Load |       +------+         +------+        +------+
   +----- +       | 1001 |         | IP   |        | Pay- |
                  +------+         +------+        | Load |
                  | IP   |         | Pay- |        +------+
                  +------+         | Load |
                  | Pay- |         +------+
                  | Load |
                  +------+
```

Figure 4: SSL or VPN label at bottom of label stack.

In some cases, the position of the SSL may not be at a fixed place in the MPLS label header. In this case, transit routers cannot expect the SSL at a fixed place in the MPLS label stack. This can be addressed by introducing a new Special Purpose Label from the label reserved space called a Slice Selector Label Indicator (SSLI). The slice policy ingress boundary node, in this case, will need to impose at least two additional MPLS labels (SSLI + SSL) to identify the slice aggregate that the packets belong to as shown in Figure 5.

```
        SR Adj-SID:          SSLI/SSL: SSLI/1001
           9012: P1-P2
           9023: P2-PE2

               /-----\        /-----\        /-----\       /-----\
               | PE1 | -----  | P1  | ------ | P2  |------ | PE2 |
               \-----/        \-----/        \-----/       \-----/

      In
      packet:
      +------+       +------+          +------+        +------+
      | IP   |       | 9012 |          | 9023 |        | SSLI |
      +------+       +------+          +------+        +------+
      | Pay- |       | 9023 |          | SSLI |        | 1001 |
      | Load |       +------+          +------+        +------+
      +------+       | SSLI |          | 1001 |        | IP   |
                     +------+          +------+        +------+
                     | 1001 |          | IP   |        | Pay- |
                     +------+          +------+        | Load |
                     | IP   |          | Pay- |        +------+
                     +------+          | Load |
                     | Pay- |          +------+
                     | Load |
                     +------+
```

Figure 5: SSLI and bottom SSL at bottom of label stack.

When the slice is realized over an IP dataplane, the SSL can be encoded in the IP header. For example, the SSL can be encoded in portion of the IPv6 Flow Label field as described in [I-D.filsfils-spring-srv6-stateless-slice-id].

5.1.2. Slice Policy Resource Reservation

Bandwidth and network resource allocation strategies for slice policies are essential to achieve optimal placement of paths within the network while still meeting the target SLOs.

Resource reservation allows for the managing of available bandwidth and for prioritization of existing allocations to enable preference-based preemption when contention on a specific network resource arises. Sharing of a network resource’s available bandwidth amongst a group of slice policies may also be desirable. For example, a slice aggregate may not always be using all of its reservable bandwidth; this allows other slice policies in the same group to use the available bandwidth resources.

Congestion on shared network resources may result from sub-optimal placement of paths in different slice policies. When this occurs, preemption of some slice aggregate specific paths may be desirable to alleviate congestion. A preference based allocation scheme enables prioritization of slice aggregate paths that can be preempted.

Since network characteristics and its state can change over time, the slice policy topology and its state also needs to be propagated in the network to enable ingress TE routers or Path Computation Engine (PCEs) to perform accurate path placement based on the current state of the slice policy network resources.

5.1.3. Slice Policy Per Hop Behavior

In Diffserv terminology, the forwarding behavior that is assigned to a specific class is called a Per Hop Behavior (PHB). The PHB defines the forwarding precedence that a marked packet with a specific CS receives in relation to other traffic on the Diffserv-aware network.

A Slice policy Per Hop Behavior (S-PHB) is the externally observable forwarding behavior applied to a specific packet belonging to a slice aggregate. The goal of an S-PHB is to provide a specified amount of network resources for traffic belonging to a specific slice aggregate. A single slice policy may also support multiple forwarding treatments or services that can be carried over the same logical network.

The slice aggregate traffic may be identified at slice policy ingress boundary nodes by carrying a SS to allow routers to apply a specific forwarding treatment that guarantee the SLA(s).

With Differentiated Services (Diffserv) it is possible to carry multiple services over a single converged network. Packets requiring the same forwarding treatment are marked with a Class Selector (CS) at domain ingress nodes. Up to eight classes or Behavior Aggregates (BAs) may be supported for a given Forwarding Equivalence Class (FEC) [RFC2475]. To support multiple forwarding treatments over the same slice aggregate, a slice aggregate packet MAY also carry a Diffserv CS to identify the specific Diffserv forwarding treatment to be applied on the traffic belonging to the same slice policy.

At transit nodes, the CS field carried inside the packets are used to determine the specific PHB that determines the forwarding and scheduling treatment before packets are forwarded, and in some cases, drop probability for each packet.

5.1.4. Slice Policy Topology

A key element of the slice policy is a customized topology that may include the full or subset of the physical network topology. The slice policy topology could also span multiple administrative domains and/or multiple dataplane technologies.

A slice policy topology can overlap or share a subset of links with another slice policy topology. A number of topology filtering policies can be defined as part of the slice policy to limit the specific topology elements that belong to a slice policy. For example, a topology filtering policy can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain links for a specific slice aggregate. The slice policy may also include a reference to a predefined topology (e.g. derived from a Flexible Algorithm Definition (FAD) as defined in [I-D.ietf-lsr-flex-algo], or Multi-Topology ID as defined [RFC4915].

5.2. Slice Policy Boundary

A network slice originates at the edge nodes of a network slice provider. Traffic that is steered over the corresponding slice policy may traverse slice policy capable interior nodes, as well as, slice policy incapable interior nodes.

The network slice may encompass one or more domains administered by a provider. For example, an organization's intranet or an ISP. The network provider is responsible for ensuring that adequate network resources are provisioned and/or reserved to support the SLAs offered by the network end-to-end.

5.2.1. Slice Policy Edge Nodes

Slice policy edge nodes sit at the boundary of a network slice provider network and receive traffic that requires steering over network resources specific to a slice aggregate. These edge nodes are responsible for identifying slice aggregate specific traffic flows by possibly inspecting multiple fields from inbound packets (e.g. implementations may inspect IP traffic's network 5-tuple in the IP and transport protocol headers) to decide on which slice policy it can be steered.

Network slice ingress nodes may condition the inbound traffic at network boundaries in accordance with the requirements or rules of each service's SLAs. The requirements and rules for network slice services are set using mechanisms which are outside the scope of this document.

When data plane slice policy is applied, the slice policy ingress boundary nodes are responsible for adding a suitable SS onto packets that belong to specific slice aggregate. In addition, edge nodes MAY mark the corresponding Diffserv CS to differentiate between different types of traffic carried over the same slice aggregate.

5.2.2. Slice Policy Interior Nodes

A slice policy interior node receives slice traffic and MAY be able to identify the packets belonging to a specific slice aggregate by inspecting the SS field carried inside each packet, or by inspecting other fields within the packet that may identify the traffic streams that belong to a specific slice aggregate. For example when data plane slice policy is applied, interior nodes can use the SS carried within the packet to apply the corresponding S-PHB forwarding behavior. Nodes within the network slice provider network may also inspect the Diffserv CS within each packet to apply a per Diffserv class PHB within the slice policy, and allow differentiation of forwarding treatments for packets forwarded over the same slice aggregate network resources.

5.2.3. Slice Policy Incapable Nodes

Packets that belong to a slice aggregate may need to traverse nodes that are slice policy incapable. In this case, several options are possible to allow the slice traffic to continue to be forwarded over such devices and be able to resume the slice policy forwarding treatment once the traffic reaches devices that are slice policy capable.

When data plane slice policy is applied, packets carry a SS to allow slice interior nodes to identify them. To enable end-to-end network slicing, the SS MUST be maintained in the packets as they traverse devices within the network - including slice policy incapable devices.

For example, when the SS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are slice policy incapable without any further considerations. On the other hand, when the SSL is at the top of the MPLS label stack, packets can be bypassed (or tunneled) over the slice policy incapable devices towards the next device that supports slice policy as shown in Figure 6.

```
     SR Node-SID:           SSL: 1001    @@@: slice policy enforced
        1601: P1                         ...: slice policy not enforced
        1602: P2
        1603: P3
        1604: P4
        1605: P5

               @@@@@@@@@@@@@@ ........................
                                                     .
              /-----\        /-----\        /-----\  .
              | P1  | -----  | P2  | ----- | P3  |   .
              \-----/        \-----/        \-----/  .
                                                |    @@@@@@@@@@
                                                |
                                            /-----\        /-----\
                                            | P4  | ------ | P5  |
                                            \-----/        \-----/


               +------+       +------+        +------+
               | 1001 |       | 1604 |        | 1001 |
               +------+       +------+        +------+
               | 1605 |       | 1001 |        | IP   |
               +------+       +------+        +------+
               | IP   |       | 1605 |        | Pay- |
               +------+       +------+        | Load |
               | Pay- |       | IP   |        +------+
               | Load |       +------+
               +----- +       | Pay- |
                              | Load |
                              +------+
```

Figure 6: Extending network slice over slice policy incapable device(s).

5.2.4. Combining Slice Policy Modes

It is possible to employ a combination of the slice policy modes that were discussed in Section 4 to realize a network slice. For example, data and control plane slice policy mode can be employed in parts of a network, while control plane slice policy mode can be employed in the other parts of the network. The path selection, in such case, can take into account the slice aggregate specific available network resources. The SS carried within packets allow transit nodes to enforce the corresponding S-PHB on the parts of the network that apply the data plane slice policy mode. The SS can be maintained while traffic traverses nodes that do not enforce data plane slice policy mode, and so slice PHB enforcement can resume once traffic traverses capable nodes.

5.3. Mapping Traffic on Slice Aggregates

The usual techniques to steer traffic onto paths can be applicable when steering traffic over paths established for a specific slice aggregate.

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established for a slice aggregate. In this case, the per Virtual Routing and Forwarding (VRF) instance traffic that arrives on the Provider Edge (PE) router over external interfaces can be directly mapped to a specific slice aggregate path. External interfaces can be further partitioned (e.g. using VLANs) to allow mapping one or more VLANs to specific slice aggregate paths.

Another option is steer traffic to specific destinations directly over multiple slice policies. This allows traffic arriving on any external interface and targeted to such destinations to be directly steered over the slice paths.

A third option that can also be used is to utilize a data plane firewall filter or classifier to enable matching of several fields in the incoming packets to decide whether the packet is steered on a specific slice aggregate. This option allows for applying a rich set of rules to identify specific packets to be mapped to a slice aggregate. However, it requires data plane network resources to be able to perform the additional checks in hardware.

6. Control Plane Extensions

Routing protocols may need to be extended to carry additional per slice aggregate link state. For example, [RFC5305], [RFC3630], and [RFC7752] are ISIS, OSPF, and BGP protocol extensions to exchange network link state information to allow ingress TE routers and PCE(s) to do proper path placement in the network. The extensions required to support network slicing may be defined in other documents, and are outside the scope of this document.

The instantiation of a slice policy may need to be automated. Multiple options are possible to facilitate automation of distribution of a slice policy to capable devices.

For example, a YANG data model for the slice policy may be supported on network devices and controllers. A suitable transport (e.g. NETCONF [RFC6241], RESTCONF [RFC8040], or gRPC) may be used to enable configuration and retrieval of state information for slice policies on network devices. The slice policy YANG data model is outside the scope of this document, and is defined [I-D.bestbar-teas-yang-slice-policy].

7. Applicability to Path Control Technologies

The slice policy modes described in this document are agnostic to the technology used to setup paths that carry slice aggregate traffic. One or more paths connecting the endpoints of the mapped IETF network slices may be selected to steer the corresponding traffic streams over the resources allocated for the slice aggregate.

For example, once the feasible paths within a slice policy topology are selected, it is possible to use RSVP-TE protocol [RFC3209] to setup or signal the LSPs that would be used to carry slice aggregate traffic. Specific extensions to RSVP-TE protocol to enable signaling of slice aggregate aware RSVP LSPs are outside the scope of this document.

Alternatively, Segment Routing (SR) [RFC8402] may be used and the feasible paths can be realized by steering over specific segments or segment-lists using an SR policy. Further details on how the slice policy modes presented in this document can be realized over an SR network is discussed in [I-D.bestbar-spring-scalable-ns], and [I-D.bestbar-lsr-spring-sa].

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

The main goal of network slicing is to allow for varying treatment of traffic from multiple different network slices that are utilizing a common network infrastructure and to allow for different levels of services to be provided for traffic traversing a given network resource.

A variety of techniques may be used to achieve this, but the end result will be that some packets may be mapped to specific resources and may receive different (e.g., better) service treatment than others. The mapping of network traffic to a specific slice policy is indicated primarily by the SS, and hence an adversary may be able to utilize resources allocated to a specific slice policy by injecting packets carrying the same SS field in their packets.

Such theft-of-service may become a denial-of-service attack when the modified or injected traffic depletes the resources available to forward legitimate traffic belonging to a specific slice policy.

The defense against this type of theft and denial-of-service attacks consists of a combination of traffic conditioning at slice policy domain boundaries with security and integrity of the network infrastructure within a slice policy domain.

10. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, Prabhu Raj Villadathu Karunakaran and Jie Dong for their review of this document, and for providing valuable feedback on it.

11. Contributors

The following individuals contributed to this document:

Colby Barth
Juniper Networks
Email: cbarth@juniper.net

Srihari R. Sangli
Juniper Networks
Email: ssangli@juniper.net

Chandra Ramachandran
Juniper Networks
Email: csekar@juniper.net

12. References

12.1. Normative References

[I-D.bestbar-lsr-spring-sa]
Saad, T., Beeram, V., Chen, R., Peng, S., Wen, B., and D. Ceccarelli, "IGP Extensions for SR Slice Aggregate SIDs", February 2021.

[I-D.bestbar-spring-scalable-ns]
Saad, T. and V. Beeram, "Scalable Network Slicing over SR Networks", draft-bestbar-spring-scalable-ns-00 (work in progress), December 2020.

[I-D.bestbar-teas-yang-slice-policy]
Saad, T. and V. Beeram, "YANG Data Model for Slice Policy", draft-bestbar-teas-yang-ns-phd-00 (work in progress), November 2020.

[I-D.filsfils-spring-srv6-stateless-slice-id]
Filsfils, C., Clad, F., Camarillo, P., and K. Raza, "Stateless and Scalable Network Slice Identification for SRv6", draft-filsfils-spring-srv6-stateless-slice-id-02 (work in progress), January 2021.

[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-13 (work in progress), October 2020.

[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>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[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>.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <https://www.rfc-editor.org/info/rfc3630>.

[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>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[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>.

[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>.

[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>.

12.2. Informative References

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[I-D.ietf-teas-rfc3272bis]
Farrel, A., "Overview and Principles of Internet Traffic Engineering", draft-ietf-teas-rfc3272bis-10 (work in progress), December 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

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

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, DOI 10.17487/RFC2702, September 1999, <https://www.rfc-editor.org/info/rfc2702>.

[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>.

[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>.

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Daniele Ceccarelli
Ericsson

Email: daniele.ceccarelli@ericsson.com

Joel Halpern
Ericsson

Email: joel.halpern@ericsson.com

Shaofu Peng
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen
ZTE Corporation

Email: chen.ran@zte.com.cn

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Luis M. Contreras
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2021

T. Saad
V. Beeram
Juniper Networks
B. Wen
Comcast
D. Ceccarelli
Ericsson
S. Peng
R. Chen
ZTE Corporation
LM. Contreras
Telefonica
X. Liu
Volta Networks

February 22, 2021

YANG Data Model for Slice Policy
draft-bestbar-teas-yang-slice-policy-00

Abstract

A slice policy is a policy construct that enables instantiation of mechanisms in support of IETF network slice specific control and data plane behaviors on select topological elements. This document defines a YANG data model for the management of slice policies on slice policy capable nodes and controllers in IP/MPLS networks.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An IETF network slice [I-D.ietf-teas-ietf-network-slice-definition] is a well-defined structure of connectivity requirements and associated network behaviors. An IETF Network Slice Controller (NSC) can realize an IETF network slice by mapping it to a slice aggregate [I-D.bestbar-teas-ns-packet]. A slice aggregate comprises of one or more IETF network slice traffic streams. The NSC uses a policy construct called the slice policy to enable the instantiation of mechanisms in support of IETF network slice specific control and data plane behaviors on select topological elements. The enforcement of the slice policy results in the creation of a slice aggregate.

A slice policy specifies the topology associated with the slice aggregate and dictates how a slice aggregate can be realized in IP/MPLS networks using one of three modes. The slice policy dictates if the partitioning of the shared network resources can be achieved in (a) just the data plane or in (b) just the control plane or in (c) both the control and data planes.

The slice policy modes (a) and (c) require the forwarding engine on each slice policy capable node to identify the traffic belonging to a specific slice aggregate and to apply the corresponding Per-Hop Behavior (PHB) that determines the forwarding treatment of the packets belonging to the slice aggregate. The identification of the slice aggregate that the packet belongs to and the corresponding forwarding treatment that needs to be applied to the packet is dictated by the slice policy.

The slice policy modes (b) and (c) require the distributed/centralized resource reservation manager in the control plane to manage slice aggregate resource reservation. The provisions for enabling slice aggregate aware traffic engineering are dictated by the slice policy.

This document defines a YANG data model for the management of slice policies on slice policy capable nodes and controllers in IP/MPLS networks.

1.1. Terminology

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

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slice-definition], [I-D.nsdt-teas-ns-framework] and [I-D.bestbar-teas-ns-packet]. The term "Network Slice" used in this document must be interpreted as "IETF Network Slice" [I-D.ietf-teas-ietf-network-slice-definition].

1.2. Tree Structure

A simplified graphical representation of the data model is presented in Appendix A of this document. The tree format defined in [RFC8340] is used for the YANG data model tree representation.

2. Slice Policy Data Model

2.1. Model Usage

The onus is on the IETF network slice controller to consume the service layer network slice intent and realize it with an appropriate slice policy. Multiple IETF network slices can be mapped to the same slice aggregate resulting in the application of the same slice policy. The network wide consistent slice policy definition (provided by the data model defined in this document) is distributed to the slice policy capable nodes and controllers as shown in Figure 1. The specification of the network slice intent on the northbound interface of the controller and the mechanism used to associate the network slice to a slice policy are outside the scope of this document.

```
                                  |
                                  | IETF Network Slice
                                  | (service)
                       +--------------------+
                       |    IETF Network    |
                       |  Slice Controller  |
                       +--------------------+
                                  |
                                  | Slice Policy
                                 /|\
                                / | \
                         slice policy capable
                           nodes/controllers
                           / /    |    \ \
                          v v     v     v v
                         xxxxxxxxxxxxxxxxxxxx
                       xxxx                xxxx
                     xxxx       Slice        xxxx
                     xxxx     Aggregate      xxxx
                       xxxx                xxxx
                         xxxxxxxxxxxxxxxxxxxx

                     <------ Path Control ------>
                     RSVP-TE/SR-Policy/SR-FlexAlgo
```

Figure 1: Slice Policy Instantiation

2.2. Model Structure

The high-level model structure defined by this document is as shown below:

```
   module: ietf-slice-policy
     +--rw network-slicing!
        +--rw phbs
        |  +--rw phb* [id]
        |  ............
        +--rw topology-filters
        |  +--rw topology-filter* [name]
        |     ............
        +--rw slice-policies
           +--rw slice-policy* [name]
              +  ............
              +--rw resource-reservation
              |  ............
              +--rw slice-selectors
              |  +--rw slice-selector* [index]
              |     ............
              +--rw phb?                    slice-policy-phb-ref
              +--rw member-topologies
                 +--rw member-topology* [topology-filter]
                    ............
```

In addition to the set of slice policies, the top-level container also includes placeholders for the set of PHBs and the set of topology filters that are referenced by the slice policies.

2.3. Per-Hop-Behaviors

The 'phbs' container carries a list of PHB entries. Each of these entries can be referenced by one or more slice policies. A PHB entry can either carry a reference to a generic PHB profile available on the node or carry a custom PHB profile. The custom PHB profile includes attributes to construct a slice aggregate specific QoS profile and any classes within it.

```
        +--rw phbs
        |  +--rw phb* [id]
        |     +--rw id                                     uint16
        |     +--rw (profile-type)?
        |        +--:(profile)
        |        |  +--rw profile?                         string
        |        +--:(custom-profile)
        |           ............
```

2.4. Topology Filters

The 'topology-filters' container carries a list of topology filters. Each topology filter entry could either reference a predefined topology or specify the rules to construct a customized topology using a set of include-any, include-all and exclude filters.

```
        +--rw topology-filters
        |  +--rw topology-filter* [name]
        |     +--rw name                                  string
        |     +--rw (topology-filter-type)?
        |        +--:(standard-topology)
        |        |  +--rw (standard-topo-type)?
        |        |     +--:(flex-algo)
        |        |     |  +--rw algo-id?                  uint8
        |        |     |  +--rw mt-id?                    uint16
        |        |     +--:(te-topo)
        |        |        +--rw te-topology-identifier
        |        |           +--rw provider-id?   te-global-id
        |        |           +--rw client-id?     te-global-id
        |        |           +--rw topology-id?   te-topology-id
        |        +--:(custom-topology)
        |           +--rw include-any
        |           |  +--rw link-affinity*   string
        |           |  +--rw link-name*       string
        |           |  +--rw node-prefix*     inet:ip-prefix
        |           |  +--rw as*              inet:as-number
        |           +--rw include-all
        |           |  +--rw link-affinity*   string
        |           |  +--rw link-name*       string
        |           |  +--rw node-prefix*     inet:ip-prefix
        |           |  +--rw as*              inet:as-number
        |           +--rw exclude
        |              +--rw link-affinity*   string
        |              +--rw link-name*       string
        |              +--rw node-prefix*     inet:ip-prefix
        |              +--rw as*              inet:as-number
```

2.5. Slice Policies

The 'slice-policies' container carries a list of slice policies. Each slice-policy entry is identified by a name and holds the set of attributes needed to instantiate a slice aggregate. The four key elements of each slice-policy entry are discussed in the following sub-sections.

2.5.1. Resource Reservation

The 'resource-reservation' container carries data nodes that are used to support slice aggregate aware bandwidth engineering. The data nodes in this container facilitate preference-based preemption of slice aggregate aware TE paths, sharing of resources amongst a group of slice aggregates and backup slice aggregate path bandwidth protection.

```
           +--rw resource-reservation
           |  +--rw preference?                        uint16
           |  +--rw (max-bw-type)?
           |  |  +--:(bw-value)
           |  |  |  +--rw maximum-bandwidth?           uint64
           |  |  +--:(bw-percentage)
           |  |     +--rw maximum-bandwidth-percent?
           |  |             rt-types:percentage
           |  +--rw shared-resource-groups*            uint32
           |  +--rw protection
           |     +--rw backup-sa-id?                     uint32
           |     +--rw (backup-bw-type)?
           |        +--:(backup-bw-value)
           |        |  +--rw backup-bandwidth?           uint64
           |        +--:(backup-bw-percentage)
           |           +--rw backup-bandwidth-percent?
           |                   rt-types:percentage
```

2.5.2. Slice Selectors

The ’slice-selectors’ container carries a set of data plane field selectors which are used to identify the packets belonging to the given slice aggregate. Each slice-selector entry in the list has an index associated with it. The slice selector with the lowest index is the default slice selector used by all the topological elements that are members of the given slice policy. The other entries are used only when there is a need to override the default slice selector on some select topological elements.

```
              +--rw slice-selectors
              |  +--rw slice-selector* [index]
              |     +--rw index      uint16
              |     +--rw mpls
              |     |  +--rw (ss-mpls-type)?
              |     |     +--:(label-value)
              |     |     |  +--rw label?
              |     |     |  |       rt-types:mpls-label
              |     |     |  +--rw label-position?          identityref
              |     |     |  +--rw label-position-offset?   uint8
              |     |     +--:(label-ranges)
              |     |        +--rw label-range* [index]
              |     |           +--rw index                    string
              |     |           +--rw start-label?
              |     |           |       rt-types:mpls-label
              |     |           +--rw end-label?
              |     |           |       rt-types:mpls-label
              |     |           +--rw label-position?
              |     |           |       identityref
              |     |           +--rw label-position-offset?   uint8
              |     +--rw ipv4
              |     |  +--rw destination-prefix*   inet:ipv4-prefix
              |     +--rw ipv6
              |     |  +--rw (ss-ipv6-type)?
              |     |     +--:(ipv6-destination)
              |     |     |  +--rw destination-prefix*
              |     |     |          inet:ipv6-prefix
              |     |     +--:(ipv6-flow-label)
              |     |        +--rw slid-flow-labels
              |     |           +--rw slid-flow-label* [slid]
              |     |              +--rw slid       inet:ipv6-flow-label
              |     |              +--rw bitmask?   uint32
              |     +--rw acl-ref*   slice-policy-acl-ref
```

2.5.3. Per-Hop-Behavior

The ’phb’ leaf carries a reference to the appropriate PHB that needs to be applied for the given slice aggregate. Unless specified otherwise, this is the default phb to be used by all the topological elements that are members of the given slice policy.

```
              +--rw phb?                    slice-policy-phb-ref
```

2.5.4. Member Topologies

The ’member-topologies’ container consists of a set of member topologies. Each member topology references a topology filter. The topological elements that satisfy the membership criteria can optionally override the default PHB and/or the default slice selector.

```
              +--rw member-topologies
                 +--rw member-topology* [topology-filter]
                    +--rw topology-filter
                    |       slice-policy-topo-filter-ref
                    +--rw slice-selector-override?   slice-policy-ss-ref
                    +--rw phb-override?
                            slice-policy-phb-ref
```

2.6. YANG Module

```
   <CODE BEGINS> file "ietf-slice-policy@2021-02-22.yang"
   module ietf-slice-policy {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-slice-policy";
     prefix "sl-pol";

     import ietf-inet-types {
       prefix "inet";
       reference
         "RFC 6991: Common YANG Data Types";
     }

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

     import ietf-access-control-list {
       prefix "acl";
       reference
         "RFC 8519: YANG Data Model for Network Access Control Lists
          (ACLs)";
     }

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

     organization
       "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group.";
     contact
       "WG Web:   <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:   Bin Wen
                  <mailto:Bin_Wen@cable.comcast.com>

        Editor:   Daniele Ceccarelli
                  <mailto:daniele.ceccarelli@ericsson.com>

        Editor:   Shaofu Peng
                  <mailto:peng.shaofu@zte.com.cn>

        Editor:   Ran Chen
                  <mailto:chen.ran@zte.com.cn>

        Editor:   Luis M. Contreras
                  <mailto:luismiguel.contrerasmurillo@telefonica.com>

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

     description
       "This YANG module defines a data model for managing slice
        policies on slice policy capable nodes and controllers.

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

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

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

     revision "2021-02-22" {
       description "Initial revision.";
       reference
```

```
         "RFC XXXX: YANG Data Model for Slice Policies.";
     }


     /*
      * I D E N T I T I E S
      */


     /*
      * Identity - MPLS Slice Selector Label Position Type
      */

     identity ss-mpls-label-position-type {
       description
         "Base identity for the position of the MPLS label that is used
          for slice selection.";
     }

     identity ss-mpls-label-position-top {
       base ss-mpls-label-position-type;
       description
         "MPLS label that is used for slice selection is at the top of
          the label stack.";
     }

     identity ss-mpls-label-position-bottom {
       base ss-mpls-label-position-type;
       description
         "MPLS label that is used for slice selection is either at the
          bottom or at a specific offset from the bottom of the label
          stack.";
     }

     identity ss-mpls-label-position-indicator {
       base ss-mpls-label-position-type;
       description
         "MPLS label that is used for slice selection is preceded by
          a special purpose indicator label in the label stack.";
     }

     /*
      * Identity - S-PHB Class Direction
      */

     identity s-phb-class-direction {
       description
         "Base identity for the direction of traffic to which the Slice
```

```
          PHB class profile is applied.";
     }

     identity s-phb-class-direction-in {
       base s-phb-class-direction;
       description
         "Slice PHB class profile is applied to incoming traffic.";
     }

     identity s-phb-class-direction-out {
       base s-phb-class-direction;
       description
         "Slice PHB class profile is applied to outgoing traffic.";
     }

     identity s-phb-class-direction-in-out {
       base s-phb-class-direction;
       description
         "Slice PHB class profile is applied to both incoming and
          outgoing directions of traffic.";
     }

     /*
      * Identity - S-PHB Class Priority
      */

     identity s-phb-class-priority {
       description
         "Base identity for the priority of the child class scheduler.";
     }

     identity s-phb-class-priority-low {
       base s-phb-class-priority;
       description
         "Priority of the child class scheduler is low.";
     }

     identity s-phb-class-priority-strict-high {
       base s-phb-class-priority;
       description
         "Priority of the child class scheduler is strict-high.";
     }

     /*
      * Identity - S-PHB Class Drop Probability
      */

     identity s-phb-class-drop-probability {
       description
         "Base identity for the drop probability applied to packets
          exceeding the CIR of the class queue.";
     }

     identity s-phb-class-drop-probability-low {
       base s-phb-class-drop-probability;
       description
         "Low drop probability applied to packets exceeding the CIR of
          the class queue.";
     }

     identity s-phb-class-drop-probability-medium {
       base s-phb-class-drop-probability;
       description
         "Medium drop probability applied to packets exceeding the CIR
          of the class queue.";
     }

     identity s-phb-class-drop-probability-high {
       base s-phb-class-drop-probability;
       description
         "High drop probability applied to packets exceeding the CIR of
          the class queue.";
     }

     /*
      * T Y P E D E F S
      */

     typedef slice-policy-acl-ref {
       type leafref {
         path "/acl:acls/acl:acl/acl:name";
       }
       description
         "This type is used to reference an ACL.";
     }

     typedef slice-policy-ss-ref {
       type leafref {
         path "/network-slicing/slice-policies/slice-policy/"
           + "slice-selectors/slice-selector/index";
       }
       description
         "This type is used to reference a Slice Selector (SS).";
     }

     typedef slice-policy-phb-ref {
       type leafref {
         path "/network-slicing/phbs/phb/"
           + "id";
       }
       description
         "This type is used to reference a Slice Policy Per-Hop
          Behavior (S-PHB).";
     }

     typedef slice-policy-topo-filter-ref {
       type leafref {
         path "/network-slicing/topology-filters/topology-filter/"
           + "name";
       }
       description
         "This type is used to reference a Slice Policy Topology.";
     }

     /*
      * G R O U P I N G S
      */

     /*
      * Grouping - Slice Selector MPLS: Label location specific fields
      */
     grouping sl-pol-ss-mpls-label-location {
       description
         "Grouping for MPLS (SS) label location specific fields.";
       leaf label-position {
         type identityref {
           base ss-mpls-label-position-type;
         }
         description
           "MPLS label position - top, bottom with offset, Slice label
            indicator.";
       }
       leaf label-position-offset {
         when "derived-from-or-self(../label-position,"
           + "'sl-pol:ss-mpls-label-position-bottom')" {
           description
             "MPLS label position offset is relevant only when the
              label-position is set to 'bottom'.";
         }
         type uint8;
         description
           "MPLS label position offset.";
       }
     }
     /*
      * Grouping - Slice Selector (SS)
      */
     grouping sl-pol-slice-selector {
       description
         "Grouping for Slice Selectors.";
       container slice-selectors {
         description
           "Container for Slice Selectors.";
         list slice-selector {
           key "index";
           description
             "List of Slice Selectors - this includes the default
              selector and others that are used for overriding the
              default.";
           leaf index {
             type uint16;
             description
               "An index to identify an entry in the slice-selector
                list. The entry with the lowest index is the
                default slice-selector.";
           }
           container mpls {
             description
               "Container for MPLS Slice Selector.";
             choice ss-mpls-type {
               description
                 "Choices for MPLS Slice Selector.";
               case label-value {
                 leaf label {
                   type rt-types:mpls-label;
                   description
                     "MPLS Slice Selector Label is explicitly
                      specified.";
                 }
                 uses sl-pol-ss-mpls-label-location;
               }
               case label-ranges {
                 list label-range {
                   key "index";
                   unique "start-label end-label";
                   description
                     "MPLS Slice Selector Label is picked from a
                      specified set of label ranges.";
                   leaf index {
                     type string;
                     description
                       "A string that uniquely identifies a label
```

```
                        range.";
                   }
                   leaf start-label {
                     type rt-types:mpls-label;
                     must '. <= ../end-label' {
                       error-message
                         "The start-label must be less than or equal "
                       + "to end-label";
                     }
                     description
                       "Label-range start.";
                   }
                   leaf end-label {
                     type rt-types:mpls-label;
                     must '. >= ../start-label' {
                       error-message
                         "The end-label must be greater than or equal "
                       + "to start-label";
                     }
                     description
                       "Label-range end.";
                   }
                   uses sl-pol-ss-mpls-label-location;
                 }
               }
             }
           }
           container ipv4 {
             description
               "Container for IPv4 Slice Selector.";
             leaf-list destination-prefix {
               type inet:ipv4-prefix;
               description
                 "IPv4 Slice Selector is picked from a specified set of
                  IPv4 destination prefixes.";
             }
           }
           container ipv6 {
             description
               "Container for IPv6 Slice Selector.";
             choice ss-ipv6-type {
               description
                 "Choices for IPv6 Slice Selector.";
               case ipv6-destination {
                 leaf-list destination-prefix {
                   type inet:ipv6-prefix;
                   description
                     "IPv6 Slice Selector is picked from a specified
```

```
                      set of IPv6 destination prefixes.";
                 }
               }
               case ipv6-flow-label {
                 container slid-flow-labels {
                   description
                     "Container for a set of Slice IDs that are
                      encoded within the flow label.";
                   list slid-flow-label {
                     key "slid";
                     description
                       "IPv6 Slice Selector is picked from a set of
                        Slice IDs that are encoded within the flow
                        label.";
                     leaf slid {
                       type inet:ipv6-flow-label;
                       description
                         "Slice ID encoded inside the IPv6 flow label.";
                     }
                     leaf bitmask {
                       type uint32;
                       description
                         "Bitmask to extract the encoded Slice ID from
                          the IPv6 flow label.";
                     }
                   }
                 }
               }
             }
           }
           leaf-list acl-ref {
             type slice-policy-acl-ref;
             description
               "Slice Selection is done based on the specified list of
                ACLs.";
           }
         }
       }
     }

     /*
      * Grouping - Slice Policy Resource Reservation
      */
     grouping sl-pol-resource-reservation {
       description
         "Grouping for slice policy resource reservation.";
       container resource-reservation {
         description
```

```
           "Container for slice policy resource reservation.";
         leaf preference {
           type uint16;
           description
             "Control plane preference for the corresponding
              slice aggregate. A higher preference
              indicates a more favorable resource
              reservation than a lower preference.";
         }
         choice max-bw-type {
           description
             "Choice of maximum bandwidth specification.";
           case bw-value {
             leaf maximum-bandwidth {
               type uint64;
               description
                 "The maximum bandwidth allocated to a slice aggregate
                  on the network resources - specified as absolute
                  value.";
             }
           }
           case bw-percentage {
             leaf maximum-bandwidth-percent {
               type rt-types:percentage;
               description
                 "The maximum bandwidth allocated to a slice aggregate
                  on the network resources - specified as percentage
                  of link capacity.";
             }
           }
         }
         leaf-list shared-resource-groups {
           type uint32;
           description
             "List of shared resource groups that a slice aggregate
              shares its allocated resources with.";
         }
         container protection {
           description
             "Container for slice aggregate protection reservation.";
           leaf backup-sa-id {
             type uint32;
             description
               "The ID that identifies the slice aggregate used
                for backup paths that protect primary paths in a
                specific slice aggregate.";
           }
           choice backup-bw-type {
             description
               "Choice of backup bandwidth specification.";
             case backup-bw-value {
               leaf backup-bandwidth {
                 type uint64;
                 description
                   "The maximum bandwidth on a network resource that
                    is allocated for backup traffic - specified as
                    absolute value.";
               }
             }
             case backup-bw-percentage {
               leaf backup-bandwidth-percent {
                 type rt-types:percentage;
                 description
                   "The maximum bandwidth on a network resource that
                    is allocated for backup traffic - specified as
                    percentage of the link capacity.";
               }
             }
           }
         }
       }
     }

     /*
      * Grouping - Slice policy PHB (S-PHB)
      */
     grouping sl-pol-phb {
       description
         "Grouping for S-PHB.";
       leaf phb {
         type slice-policy-phb-ref;
         description
           "Reference to a specific PHB from the list of global
            PHBs.";
       }
     }

     /*
      * Grouping - Slice policy default profile override
      */
     grouping sl-pol-override-options {
       description
         "Grouping of fields that are used to override the default
          profile of the slice policy.";
       leaf slice-selector-override {
         type slice-policy-ss-ref;
         description
           "Reference to a specific Slice Selector (different from
            default).";
       }
       leaf phb-override {
         type slice-policy-phb-ref;
         description
           "Reference to a specific PHB (different from default).";
       }
     }

     /*
      * Grouping - Standard Topology Filter
      */
     grouping sl-pol-topo-filter-standard {
       description
         "Grouping for standard topology filter.";
       choice standard-topo-type {
         description
           "Choice of standard topology filter.";
         case flex-algo {
           leaf algo-id {
             type uint8;
             description
               "Algorithm ID.";
           }
           leaf mt-id {
             type uint16;
             description
               "Multi Topology ID.";
           }
         }
         case te-topo {
           uses te-types:te-topology-identifier;
         }
       }
     }

     /*
      * Grouping - Custom Topology Filters
      */
     grouping sl-pol-topo-filter-custom {
       description
         "Grouping for custom topology filters.";
       leaf-list link-affinity {
         type string;
         description
           "Match-filter is a list of link affinities.";
       }
       leaf-list link-name {
         type string;
         description
           "Match-filter is a list of link names.";
       }
       leaf-list node-prefix {
         type inet:ip-prefix;
         description
           "Match-filter is a list of node IDs.";
       }
       leaf-list as {
         type inet:as-number;
         description
           "Match-filter is a list of AS numbers.";
       }
     }

     /*
      * Grouping - Member Topologies
      */
     grouping sl-pol-member-topologies {
       description
         "Grouping for member topologies.";
       container member-topologies {
         description
           "Container for member topologies.";
         list member-topology {
           key "topology-filter";
           description
             "List of member topologies.";
           leaf topology-filter {
             type slice-policy-topo-filter-ref;
             description
               "Reference to a specific topology filter from the list
                of global topology filters.";
           }
           uses sl-pol-override-options;
         }
       }
     }

     /*
      * Grouping - Per-Hop Behaviors (PHBs)
      */
     grouping sl-pol-phbs {
       description
         "Grouping for PHBs.";
       container phbs {
         description
           "Container for PHBs.";
         list phb {
           key "id";
           description
             "List of PHBs.";
           leaf id {
             type uint16;
             description
               "A 16-bit ID that uniquely identifies the PHB.";
           }
           choice profile-type {
             description
               "Choice of PHB profile type.";
             case profile {
               description
                 "Generic PHB profile available on the network
                  element.";
               leaf profile {
                 type string;
                 description
                   "Generic PHB profile identifier.";
               }
             }
             case custom-profile {
               description
                 "Custom PHB profile.";
               choice guaranteed-rate-type {
                 description
                   "Guaranteed rate is the committed information rate
                    (CIR) of the slice aggregate. The guaranteed rate
                    also determines the amount of excess (extra)
                    bandwidth that a group of slice aggregates can
                    share. Extra bandwidth is allocated among the
                    group in proportion to the guaranteed rate of
                    each slice aggregate.";
                 case rate {
                   leaf guaranteed-rate {
                     type uint64;
                     description
                       "Guaranteed rate specified as absolute value.";
                   }
                 }
                 case percentage {
                   leaf guaranteed-rate-percent {
                     type rt-types:percentage;
                     description
```

```
                       "Guaranteed rate specified in percentage.";
                   }
                 }
               }
               choice shaping-rate-type {
                 description
                   "Shaping rate is the maximum bandwidth of the slice
                    aggregate; the peak information rate (PIR) of a
                    slice aggregate.";
                 case rate {
                   leaf shaping-rate {
                     type uint64;
                     description
                       "Shaping rate specified as absolute value.";
                   }
                 }
                 case percentage {
                   leaf shaping-rate-percent {
                     type rt-types:percentage;
                     description
                       "Shaping rate specified in percentage.";
                   }
                 }
               }
               container classes {
                 description
                   "Container for classes.";
                 list class {
                   key class-id;
                   description
                     "List of classes.";
                   leaf class-id {
                     type string;
                     description
                       "A string to uniquely identify a class.";
                   }
                   leaf direction {
                     type identityref {
                       base s-phb-class-direction;
                     }
                     description
                       "Class direction.";
                   }
                   leaf priority {
                     type identityref {
                       base s-phb-class-priority;
                     }
                     description
```

```
                       "Priority of the class scheduler. Only one slice
                        aggregate class queue can be set as a
                        strict-high priority queue. Strict-high
                        priority allocates the scheduled bandwidth to
                        the queue before any other queue receives
                        bandwidth. Other queues receive the bandwidth
                        that remains after the strict-high queue has
                        been serviced.";
                   }
                   choice guaranteed-rate-type {
                     description
                       "Guaranteed Rate is the Committed information
                        rate (CIR) of slice aggregate class - specified
                        as absolute value or percentage.";
                     case rate {
                       leaf guaranteed-rate {
                         type uint64;
                         description
                           "Guaranteed rate specified as absolute
                            value.";
                       }
                     }
                     case percentage {
                       leaf guaranteed-rate-percent {
                         type rt-types:percentage;
                         description
                           "Guaranteed rate specified in percentage.";
                       }
                     }
                   }
                   leaf drop-probability {
                     type identityref {
                       base s-phb-class-drop-probability;
                     }
                     description
                       "Drop probability applied to packets exceeding
                        the CIR of the class queue.";
                   }
                   choice maximum-bandwidth-type {
                     description
                       "Maximum bandwidth is the Peak information
                        rate (PIR) of slice aggregate class - specified
                        as absolute value or percentage.";
                     case rate {
                       leaf maximum-bandwidth {
                         type uint64;
                         description
                           "Maximum bandwidth specified as absolute
```

```
                            value.";
                       }
                     }
                     case percentage {
                       leaf maximum-bandwidth-percent {
                         type rt-types:percentage;
                         description
                           "Maximum bandwidth specified as percentage.";
                       }
                     }
                   }
                   choice delay-buffer-size-type {
                     description
                       "Size of the queue buffer as a percentage of the
                        dedicated buffer space - specified as value or
                        percentage.";
                     case value {
                       leaf delay-buffer-size {
                         type uint64;
                         description
                           "Delay buffer size.";
                       }
                     }
                     case percentage {
                       leaf delay-buffer-size-percent {
                         type rt-types:percentage;
                         description
                           "Delay buffer size specified as percentage.";
                       }
                     }
                   }
                 }
               }
             }
           }
         }
       }
     }

     /*
      * Grouping - Topology Filters
      */
     grouping sl-pol-topology-filters {
       description
         "Grouping for topology filters.";
       container topology-filters {
         description
           "Container for topology filters.";
         list topology-filter {
           key "name";
           description
             "List of topology filters.";
           leaf name {
             type string;
             description
               "A string that uniquely identifies the topology filter.";
           }
           choice topology-filter-type {
             description
               "Choice of topology filter type.";
             case standard-topology {
               uses sl-pol-topo-filter-standard;
             }
             case custom-topology {
               container include-any {
                 description
                   "Include-any filters.";
                 uses sl-pol-topo-filter-custom;
               }
               container include-all {
                 description
                   "Include-all filters.";
                 uses sl-pol-topo-filter-custom;
               }
               container exclude {
                 description
                   "Exclude filters.";
                uses sl-pol-topo-filter-custom;
               }
             }
           }
         }
       }
     }

     /*
      * Grouping - Slice Policies
      */
     grouping sl-policies {
       description
         "Grouping for slice policies.";
       container slice-policies {
         description
           "Container for slice policies.";
         list slice-policy {
           key "name";
           unique "sa-id";
           description
             "List of slice policies.";
           leaf name {
             type string;
             description
               "A string that uniquely identifies the slice policy.";
           }
           leaf sa-id {
             type uint32;
             description
               "A 32-bit ID that uniquely identifies the slice
                aggregate created by the enforcement of this slice
                policy.";
           }
           uses sl-pol-resource-reservation;
           uses sl-pol-slice-selector;
           uses sl-pol-phb;
           uses sl-pol-member-topologies;
         }
       }
     }

     /*
      * Top-level container - Network Slicing
      */
     container network-slicing {
       presence "Enable network slicing.";
       description
         "Top-level container for network slicing specific constructs
          on a slice policy capable network entity.";
       uses sl-pol-phbs;
       uses sl-pol-topology-filters;
       uses sl-policies;
     }
   }
   <CODE ENDS>
```

3. Acknowledgements

The authors would like to thank Krzysztof Szarkowicz for his input from discussions.

4. Contributors

The following individuals contributed to this document:

Colby Barth

Juniper Networks
Email: cbarth@juniper.net

Srihari R. Sangli
Juniper Networks
Email: ssangli@juniper.net

Chandra Ramachandran
Juniper Networks
Email: csekar@juniper.net

5. IANA Considerations

This document registers the following 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-slice-phd
Registrant Contact: The TEAS WG of the IETF.
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-slice-phd
namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice-phd
prefix: ns-phd
reference: RFCXXXX

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.

The data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default) 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:

* "/network-slicing/phbs": This subtree specifies the configurations for slice policy per-hop behaviors. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to be provided for the slice aggregate traffic on the network element.

* "/network-slicing/topology-filters": This subtree specifies the configurations for slice policy topology filters. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to be provided for the slice aggregate traffic on the network element.

* "/network-slicing/slice-policies": This subtree specifies the configurations for slice policies on a given network element. By manipulating these data nodes, a malicious attacker may cause unauthorized and improper behavior to be provided for the slice aggregate traffic on the network element.

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:

* "/network-slicing/phbs": Unauthorized access to this subtree can disclose the slice policy PHBs defined on the network element.

* "/network-slicing/topology-filters": Unauthorized access to this subtree can disclose the slice policy topology filters on the network element.

* "/network-slicing/slice-policies": Unauthorized access to this subtree can disclose the slice policy definitions on the network element.

7. References

7.1. Normative References

[I-D.bestbar-teas-ns-packet]
Saad, T., Beeram, V., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., and X. Liu, "Realizing Network Slices in IP/MPLS Networks", draft-bestbar-teas-ns-packet-01 (work in progress), December 2020.

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[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>.

[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

[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. Complete Model Tree Structure

```
   module: ietf-slice-policy
     +--rw network-slicing!
        +--rw phbs
        |  +--rw phb* [id]
        |     +--rw id                                     uint16
        |     +--rw (profile-type)?
        |        +--:(profile)
        |        |  +--rw profile?                         string
        |        +--:(custom-profile)
        |           +--rw (guaranteed-rate-type)?
        |           |  +--:(rate)
        |           |  |  +--rw guaranteed-rate?           uint64
        |           |  +--:(percentage)
        |           |     +--rw guaranteed-rate-percent?
        |           |             rt-types:percentage
        |           +--rw (shaping-rate-type)?
        |           |  +--:(rate)
        |           |  |  +--rw shaping-rate?              uint64
        |           |  +--:(percentage)
        |           |     +--rw shaping-rate-percent?
        |           |             rt-types:percentage
        |           +--rw classes
        |              +--rw class* [class-id]
        |                 +--rw class-id
        |                 |       string
        |                 +--rw direction?
        |                 |       identityref
        |
        |                 +--rw priority?
        |                 |       identityref
        |                 +--rw (guaranteed-rate-type)?
        |                 |  +--:(rate)
        |                 |  |  +--rw guaranteed-rate?
        |                 |  |          uint64
        |                 |  +--:(percentage)
        |                 |     +--rw guaranteed-rate-percent?
        |                 |             rt-types:percentage
        |                 +--rw drop-probability?
        |                 |       identityref
        |                 +--rw (maximum-bandwidth-type)?
        |                 |  +--:(rate)
        |                 |  |  +--rw maximum-bandwidth?
        |                 |  |          uint64
        |                 |  +--:(percentage)
        |                 |     +--rw maximum-bandwidth-percent?
        |                 |             rt-types:percentage
        |                 +--rw (delay-buffer-size-type)?
        |                    +--:(value)
        |                    |  +--rw delay-buffer-size?
        |                    |          uint64
        |                    +--:(percentage)
        |                       +--rw delay-buffer-size-percent?
        |                               rt-types:percentage
        +--rw topology-filters
        |  +--rw topology-filter* [name]
        |     +--rw name                                  string
        |     +--rw (topology-filter-type)?
        |        +--:(standard-topology)
        |        |  +--rw (standard-topo-type)?
        |        |     +--:(flex-algo)
        |        |     |  +--rw algo-id?                  uint8
        |        |     |  +--rw mt-id?                    uint16
        |        |     +--:(te-topo)
        |        |        +--rw te-topology-identifier
        |        |           +--rw provider-id?   te-global-id
        |        |           +--rw client-id?     te-global-id
        |        |           +--rw topology-id?   te-topology-id
        |        +--:(custom-topology)
        |           +--rw include-any
        |           |  +--rw link-affinity*   string
        |           |  +--rw link-name*       string
        |           |  +--rw node-prefix*     inet:ip-prefix
        |           |  +--rw as*              inet:as-number
        |           +--rw include-all
        |           |  +--rw link-affinity*   string
        |           |  +--rw link-name*       string
        |           |  +--rw node-prefix*     inet:ip-prefix
        |           |  +--rw as*              inet:as-number
        |           +--rw exclude
        |              +--rw link-affinity*   string
        |              +--rw link-name*       string
        |              +--rw node-prefix*     inet:ip-prefix
        |              +--rw as*              inet:as-number
        +--rw slice-policies
           +--rw slice-policy* [name]
              +--rw name                    string
              +--rw sa-id?                  uint32
              +--rw resource-reservation
              |  +--rw preference?                        uint16
              |  +--rw (max-bw-type)?
              |  |  +--:(bw-value)
              |  |  |  +--rw maximum-bandwidth?           uint64
              |  |  +--:(bw-percentage)
              |  |     +--rw maximum-bandwidth-percent?
              |  |             rt-types:percentage
              |  +--rw shared-resource-groups*            uint32
              |  +--rw protection
              |     +--rw backup-sa-id?                     uint32
              |     +--rw (backup-bw-type)?
              |        +--:(backup-bw-value)
              |        |  +--rw backup-bandwidth?           uint64
              |        +--:(backup-bw-percentage)
              |           +--rw backup-bandwidth-percent?
              |                   rt-types:percentage
              +--rw slice-selectors
              |  +--rw slice-selector* [index]
              |     +--rw index      uint16
              |     +--rw mpls
              |     |  +--rw (ss-mpls-type)?
              |     |     +--:(label-value)
              |     |     |  +--rw label?
              |     |     |  |       rt-types:mpls-label
              |     |     |  +--rw label-position?          identityref
              |     |     |  +--rw label-position-offset?   uint8
              |     |     +--:(label-ranges)
              |     |        +--rw label-range* [index]
              |     |           +--rw index                    string
              |     |           +--rw start-label?
              |     |           |       rt-types:mpls-label
              |     |           +--rw end-label?
              |     |           |       rt-types:mpls-label
              |     |           +--rw label-position?
              |     |           |       identityref
              |     |           +--rw label-position-offset?   uint8
              |     +--rw ipv4
              |     |  +--rw destination-prefix*   inet:ipv4-prefix
              |     +--rw ipv6
              |     |  +--rw (ss-ipv6-type)?
              |     |     +--:(ipv6-destination)
              |     |     |  +--rw destination-prefix*
              |     |     |          inet:ipv6-prefix
              |     |     +--:(ipv6-flow-label)
              |     |        +--rw slid-flow-labels
              |     |           +--rw slid-flow-label* [slid]
              |     |              +--rw slid       inet:ipv6-flow-label
              |     |              +--rw bitmask?   uint32
              |     +--rw acl-ref*   slice-policy-acl-ref
              +--rw phb?                    slice-policy-phb-ref
              +--rw member-topologies
                 +--rw member-topology* [topology-filter]
                    +--rw topology-filter
                    |       slice-policy-topo-filter-ref
                    +--rw slice-selector-override?   slice-policy-ss-ref
                    +--rw phb-override?
                            slice-policy-phb-ref
```

Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Daniele Ceccarelli
Ericsson

Email: daniele.ceccarelli@ericsson.com

Shaofu Peng
ZTE Corporation

Email: peng.shaofu@zte.com.cn

Ran Chen
ZTE Corporation

Email: chen.ran@zte.com.cn

Luis M. Contreras
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

TEAS Working Group
Internet Draft
Intended status: Informational

Italo Busi
Huawei
Xufeng Liu
Volta Networks
Igor Bryskin
Individual
Vishnu Pavan Beeram
Tarek Saad
Juniper Networks
Oscar Gonzalez de Dios
Telefonica

Expires: August 2021
February 19, 2021

Profiles for Traffic Engineering (TE) Topology Data Model
draft-busi-teas-te-topology-profiles-01

Abstract

This document describes how profiles of the Traffic Engineering (TE) Topology Model, defined in RFC8795, can be used to address applications beyond "Traffic Engineering".

Status of this Memo

This Internet-Draft will expire on August 19, 2021.

Copyright Notice

Table of Contents

1. Introduction

There are many network scenarios being discussed in various IETF Working Groups (WGs) that are not classified as "Traffic Engineering" but can be addressed by a sub-set (profile) of the Traffic Engineering (TE) Topology YANG data model, defined in [RFC8795].

Traffic Engineering (TE) is defined in [RFC3272bis] as aspects of Internet network engineering that deal with the issues of performance evaluation and performance optimization of operational IP networks. TE encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of Internet traffic.

The TE Topology Model is augmenting the Network Topology Model defined in [RFC8345] with generic and technology-agnostic features that some are strictly applicable to TE networks, while others applicable to both TE and non-TE networks.

Examples of such features that are applicable to both TE and non-TE networks are: inter-domain link discovery (plug-id), geo-localization, and admin/operational status.

It is also worth noting that the TE Topology Model is quite an extensive and comprehensive model in which most features are optional. Therefore, even though the full model appears to be complex, at the first glance, a sub-set of the model (profile) can be used to address specific scenarios, e.g. suitable also to non-TE use cases.

The implementation of such TE Topology profiles can simplify and expedite adoption of the full TE topology YANG data model, and allow for its reuse even for non-TE use case. The key question being whether all or some of the attributes defined in the TE Topology Model are needed to address a given network scenario.

Section 2 provides examples where profiles of the TE Topology Model can be used to address some generic use cases applicable to both TE and non-TE technologies.

2. Examples of non-TE scenarios

2.1. UNI Topology Discovery

UNI Topology Discovery is independent from whether the network is TE or non-TE.

The TE Topology Model supports inter-domain link discovery (including but not being limited to UNI link discovery) using the plug-id attribute. This solution is quite generic and does not require the network to be a TE network.

The following profile of the TE Topology model can be used for the UNI Topology Discovery:

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw te-tp-id?   te-types:te-tp-id
    +--rw te!
       +--rw admin-status?
       |       te-types:te-admin-status
       +--rw inter-domain-plug-id?          binary
       +--ro oper-status?                   te-types:te-oper-status
```

Figure 1 - UNI Topology

The profile data model shown in Figure 1 can be used to discover TE and non TE UNIs as well as to discover UNIs for TE or non TE networks.

Such a UNI TE Topology profile model can also be used with technology-specific UNI augmentations, as described in section 3.

For example, in [CLIENT-TOPO], the eth-svc container is defined to represent the capabilities of the Termination Point (TP) to be configured as an Ethernet client UNI, together with the Ethernet classification and VLAN operations supported by that TP.

The [OTN-TOPO] provides another example, where:

o the client-svc container is defined to represent the capabilities of the TP to be configured as an transparent client UNI (e.g., STM-N, Fiber Channel or transparent Ethernet);

o the OTN technology-specific Link Termination Point (LTP) augmentations are defined to represent the capabilities of the TP to be configured as an OTN UNI, together with the information about OTN label and bandwidth availability at the OTN UNI.

For example, the UNI TE Topology profile can be used to model features defined in [UNI-TOPO]:

o The inter-domain-plug-id attribute would provide the same information as the attachment-id attribute defined in [UNI-TOPO];

o The admin-status and oper-status that exists in this TE topology profile can provide the same information as the admin-status and oper-status attributes defined in [UNI-TOPO].

Following the same approach in [CLIENT-TOPO] and [OTN-TOPO], the type and encapsulation-type attributes can be defined by technology-specific UNI augmentations to represent the capability of a TP to be configured as a L2VPN/L3VPN UNI Service Attachment Point (SAP).

The advantages of using a TE Topology profile would be having common solutions for:

- discovering UNIs as well as inter-domain NNI links, which is applicable to any technology (TE or non TE) used at the UNI or within the network;

- modelling non TE UNIs such as Ethernet, and TE UNIs such as OTN, as well as UNIs which can configured as TE or non-TE (e.g., being configured as either Ethernet or OTN UNI).

2.2. Administrative and Operational status management

The TE Topology Model supports the management of administrative and operational state, including also the possibility to associate some administrative names, for nodes, termination points and links. This solution is generic and also does not require the network to be a TE network.

The following profile of the TE Topology Model can be used for administrative and operational state management:

```
   module: ietf-te-topology
     augment /nw:networks/nw:network/nw:network-types:
       +--rw te-topology!
     augment /nw:networks/nw:network:
       +--rw te-topology-identifier
       |  +--rw provider-id?   te-global-id
       |  +--rw client-id?     te-global-id
       |  +--rw topology-id?   te-topology-id
       +--rw te!
          +--rw name?                     string
     augment /nw:networks/nw:network/nw:node:
       +--rw te-node-id?   te-types:te-node-id
       +--rw te!
          +--rw te-node-attributes
          |  +--rw admin-status?               te-types:te-admin-status
          |  +--rw name?                       string
          +--ro oper-status?                   te-types:te-oper-status
     augment /nw:networks/nw:network/nt:link:
       +--rw te!
          +--rw te-link-attributes
          |  +--rw name?                       string
          |  +--rw admin-status?               te-types:te-admin-status
          +--ro oper-status?                   te-types:te-oper-status
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--rw te-tp-id?   te-types:te-tp-id
       +--rw te!
          +--rw admin-status?                  te-types:te-admin-status
          +--rw name?                          string
          +--ro oper-status?                   te-types:te-oper-status
```

Figure 2 - Generic Topology with admin and operational state

The TE topology data model profile shown in Figure 2 is applicable to any technology (TE or non-TE) that requires management of the administrative and operational state and administrative names for nodes, termination points and links.

2.3. Geolocation

The TE Topology model supports the management of geolocation coordinates for nodes and termination points. This solution is generic and does not necessarily require the network to be a TE network.

The TE topology data model profile shown in Figure 3can be used to model geolocation data for networks.

```
   module: ietf-te-topology
     augment /nw:networks/nw:network/nw:network-types:
       +--rw te-topology!
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--rw te-tp-id?   te-types:te-tp-id
       +--rw te!
          +--ro geolocation
             +--ro altitude?    int64
             +--ro latitude?    geographic-coordinate-degree
             +--ro longitude?   geographic-coordinate-degree
     augment /nw:networks/nw:network/nw:node:
       +--rw te-node-id?   te-types:te-node-id
       +--rw te!
          +--ro geolocation
             +--ro altitude?    int64
             +--ro latitude?    geographic-coordinate-degree
             +--ro longitude?   geographic-coordinate-degree
     augment /nw:networks/nw:network/nw:node/nt:termination-point:
       +--rw te-tp-id?   te-types:te-tp-id
       +--rw te!
          +--ro geolocation
             +--ro altitude?    int64
             +--ro latitude?    geographic-coordinate-degree
             +--ro longitude?   geographic-coordinate-degree
```

Figure 3 - Generic Topology with geolocation information

This profile is applicable to any network technology (TE or non-TE) that requires management of the geolocation information for its nodes and termination points.

2.4. Overlay and Underlay non-TE Topologies

The TE Topology model supports the management of overlay/underlay relationship for nodes and links, as described in section 5.8 of [RFC8795]. This solution is generic and does not require the network to be a TE network.

The following TE topology data model profile can be used to manage overlay/underlay network data:

```
   module: ietf-te-topology
     augment /nw:netorks/nw:network/nw:network-types:
       +--rw te-topology!
     augment /nw:networks/nw:network/nw:node:
       +--rw te-node-id?   te-types:te-node-id
       +--rw te!
          +--rw te-node-attributes
             +--rw underlay-topology {te-topology-hierarchy}?
                +--rw network-ref?   -> /nw:networks/network/network-id
     augment /nw:networks/nw:network/nt:link:
       +--rw te!
          +--rw te-link-attributes
             +--rw underlay {te-topology-hierarchy}?
                +--rw enabled?                     boolean
                +--rw primary-path
                   +--rw network-ref?
                   |       -> /nw:networks/network/network-id
                   +--rw path-element* [path-element-id]
                      +--rw path-element-id              uint32
                      +--rw (type)?
                         +--:(numbered-link-hop)
                         |  +--rw numbered-link-hop
                         |     +--rw link-tp-id    te-tp-id
                         |     +--rw hop-type?     te-hop-type
                         |     +--rw direction?    te-link-direction
                         +--:(unnumbered-link-hop)
                            +--rw unnumbered-link-hop
                               +--rw link-tp-id    te-tp-id
                               +--rw node-id       te-node-id
                               +--rw hop-type?     te-hop-type
                               +--rw direction?    te-link-direction
```

Figure 4 - Generic Topology with overlay/underlay information

This profile is applicable to any technology (TE or non-TE) when it is needed to manage the overlay/underlay information. It is also allows a TE underlay network to support a non-TE overlay network and, vice versa, a non-TE underlay network to support a TE overlay network.

2.5. Nodes with switching limitations

A node can have some switching limitations where connectivity is not possible between all its TP pairs, for example when:

o the node represents a physical device with switching limitations;

- the node represents an abstraction of a network topology.

This scenario is generic and applies to both TE and non-TE technologies.

A connectivity TE Topology profile data model supports the management of the node connectivity matrix to represent feasible connections between termination points across the nodes. This solution is generic and does not necessarily require a TE enabled network.

The following profile of the TE Topology model can be used for nodes with connectivity constraints:

```
module: ietf-te-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw te-topology!
  augment /nw:networks/nw:network/nw:node:
    +--rw te-node-id?   te-types:te-node-id
    +--rw te!
       +--rw te-node-attributes
          +--rw connectivity-matrices
             +--rw number-of-entries?     uint16
             +--rw is-allowed?            boolean
             +--rw connectivity-matrix* [id]
                +--rw id                  uint32
                +--rw from
                |  +--rw tp-ref?               leafref
                +--rw to
                |  +--rw tp-ref?               leafref
                +--rw is-allowed?              boolean
```

Figure 5 - Generic Topology with connectivity constraints

The TE topology data model profile shown in Figure 5 is applicable to any technology (TE or non-TE) networks that requires managing nodes with certain connectivity constraints. When used with TE technologies, additional TE attributes, as defined in [RFC8795], can also be provided.

3. Technology-specific augmentations

There are two main options to define technology-specific Topology Models which can use the attributes defined in the TE Topology Model [RFC8795].

Both options are applicable to any possible profile of the TE Topology Model, such as those defined in section 2.

The first option is to define a technology-specific TE Topology Model which augments the TE Topology Model, as shown in Figure 6:

```
                           +--------------------+
                           | Network Topology   |
                           +--------------------+
                                     ^
                                     |
                                     | Augments
                                     |
                         +-----------+-----------+
                         |       TE Topology       |
                         |       (profile)       |
                         +-----------------------+
                                     ^
                                     |
                                     | Augments
                                     |
                          +----------+----------+
                          | Technology-Specific |
                          |     TE Topology     |
                          +---------------------+
```

Figure 6 Augmenting the TE Topology Model

This approach is more suitable for cases when the technology-specific TE topology model provides augmentations to the TE Topology constructs, such as bandwidth information (e.g., link bandwidth), tunnel termination points (TTPs) or connectivity matrices. It also allows providing augmentations to the Network Topology constructs, such as nodes, links, and termination points (TPs).

This is the approach currently used in [CLIENT-TOPO] and [OTN-TOPO].

It is worth noting that a profile of the technology-specific TE Topology model not using any TE topology attribute or constructs can be used to address any use case that do not require these attributes. In this case, only the te-topology presence container of the TE Topology Model needs to be implemented.

The second option is to define a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, which is implicit in the network-types definition of [RFC8345], to allow using also the generic attributes defined in the TE Topology model:

```
                    +-----------------------+
                    |    Network Topology   |
                    +-----------------------+
                        ^               ^
                        |               |
           Augments +---+               +--+ Augments
                    |                      |
          +---------+---+       +----------+----------+
          | TE Topology |       | Technology-specific |
          |  (profile)  |       |  Network Topology   |
          +-------------+       +---------------------+
```

Figure 7 Augmenting the Network Topology Model with multi-inheritance

This approach is more suitable in cases where the technology-specific Network Topology Model provides augmentation only to the constructs defined in the Network Topology Model, such as nodes, links, and termination points (TPs). Therefore, with this approach, only the generic attributes defined in the TE Topology Model could be used.

It is also worth noting that in this case, technology-specific augmentations for the bandwidth information could not be defined.

In principle, it would be also possible to define both a technology specific TE Topology Model which augments the TE Topology Model, and a technology-specific Network Topology Model which augments the Network Topology Model and to rely on the multiple inheritance capability, as shown in Figure 8:

```
                    +-----------------------+
                    |    Network Topology   |
                    +-----------------------+
                        ^               ^
                        |               |
           Augments +---+               +--+ Augments
                    |                      |
          +---------+---+       +----------+----------+
          | TE Topology |       | Technology-specific |
          |  (profile)  |       |  Network Topology   |
          +-------------+       +---------------------+
                 ^                         ^
                 |                         |
                 | Augments                | References
                 |                         |
      +----------+----------+              |
      | Technology-Specific +--------------+
      |     TE Topology     |
      +---------------------+
```

Figure 8 Augmenting both the Network and TE Topology Models

This option does not provide any technical advantage with respect to the first option, shown in Figure 6, but could be useful to add augmentations to the TE Topology constructs and to re-use an already existing technology-specific Network Topology Model.

It is worth noting that the technology-specific TE Topology model can reference constructs defined by the technology-specific Network Topology model but it could not augment constructs defined by the technology-specific Network Topology model.

3.1. Example (Link augmentation)

This section provides an example on how technology-specific attributes can be added to the Link construct:

```
      +--rw link* [link-id]
         +--rw link-id           link-id
         +--rw source
         |  +--rw source-node?   -> ../../../nw:node/node-id
         |  +--rw source-tp?     leafref
         +--rw destination
         |  +--rw dest-node?   -> ../../../nw:node/node-id
         |  +--rw dest-tp?     leafref
         +--rw supporting-link* [network-ref link-ref]
         |  +--rw network-ref
         |  |       -> ../../../nw:supporting-network/network-ref
         |  +--rw link-ref       leafref
         +--rw example-link-attributes
         |   <...>
         +--rw te!
            +--rw te-link-attributes
               +--rw name?                            string
               +--rw example-te-link-attributes
               |   <...>
               +--rw max-link-bandwidth
                  +--rw te-bandwidth
                     +--rw (technology)?
                        +--:(generic)
                        |  +--rw generic?   te-bandwidth
                        +--:(example)
                           +--rw example?   example-bandwidth
```

Figure 9 Augmenting the Link with technology-specific attributes

The technology-specific attributes within the example-link-attributes container can be defined either in the technology-specific TE Topology Model (Option 1) or in the technology-specific Network Topology Model (Option 2 or Option 3). These attributes can only be non-TE and do not require the implementation of the te container.

The technology-specific attributes within the example-te-link-attributes container as well as the example max-link-bandwidth can only be defined in the technology-specific TE Topology Model (Option 1 or Option 3). These attributes can be TE or non-TE and require the implementation of the te container.

4. Security Considerations

This document provides only information about how the TE Topology Model, as defined in [RFC8795], can be profiled to address some scenarios which are not considered as TE.

As such, this document does not introduce any additional security considerations besides those already defined in [RFC8795].

5. IANA Considerations

This document requires no IANA actions.

6. References

6.1. Normative References

[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>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

6.2. Informative References

[RFC3272bis] Farrel A., "Overview and Principles of Internet Traffic Engineering", draft-dt-teas-rfc3272bis-08, work in progress.

[UNI-TOPO] Gonzalez de Dios, O. et al., "A YANG Model for User-Network Interface (UNI) Topologies", draft-ogondio-opsawg-uni-topology-01, work in progress.

[CLIENT-TOPO] Zheng, H. et al., "A YANG Data Model for Client-layer Topology", draft-zheng-ccamp-client-topo-yang-10, work in progress.

[OTN-TOPO] Zheng, H. et al., "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-11, work in progress.

Acknowledgments

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

Contributors

Aihua Guo
Futurewei Inc.

Email: aihuaguo.ietf@gmail.com

Haomian Zheng
Huawei

Email: zhenghaomian@huawei.com

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com

Authors’ Addresses

Italo Busi
Huawei

Email: italo.busi@huawei.com

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

Oscar Gonzalez de Dios
Telefonica

Email: oscar.gonzalezdedios@telefonica.com

IDR Working Group
Internet-Draft
Intended status: Informational
Expires: August 24, 2021

W. Cheng
W. Jiang
China Mobile
R. Chen
ZTE Corporation
L. Gong
China Mobile
C. F
H3C Corporation
Sh. Peng
ZTE Corporation

February 20, 2021

IETF Network Slice use cases
draft-cheng-teas-network-slice-usecase-00

Abstract

This draft supplements the usecase described in [I-D.ietf-teas-ietf-network-slice-definition] from the perspective of the operator.In specific,it mainly includes two types of the network slice customers from the perspective of operators:

o End-to-end slicing cloud-network collaboration

o The branch departments that use slices within the operator.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-teas-ietf-network-slice-definition] defines the concept of IETF network slices that provide connectivity coupled with a set of specific commitments of network resources between a number of endpoints over a shared network infrastructure and describes a number of use-cases benefiting from network slicing including:

o 5G network slicing

o Network wholesale services

o Network sharing among operators

o NFV connectivity and Data Center Interconnect

In the document also clearly stated services that might benefit from the network slices include but not limited to the above use-cases.

This document supplements two use-cases from the perspective of operators. In specific, it mainly includes two types of the network slice customers from the perspective of operators:

o End-to-end slicing cloud-network collaboration

o The branch departments that use slices within the operator.

2. Requirements notation

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].

3. Network Slice use cases

3.1. cloud-network service for enterprise

```
                                +-------------------------------------------+
                                | Consumer higher level operation system    |
               +----------------|   (e.g E2E network slice orchestrator)    |-----
--------+
               |                +-------------------------------------------+
        |
               |                          A                      A
        |
               |                          |                      |
        |
               |                          |                      |
        |
               |                          |                      |
        |
               V                          V                      V
        V
         |----------------|   |-------------------|     |-----------------|    |-
----------------|
         |  MAN Slice     |   |    Edge Cloud     |     |  Backbone Slice |    |
  DC Slice      |
         |  Controller    |   |  Slice Controller |     |   Controller    |    |
  Controller    |
         |----------------|   |-------------------|     |-----------------|    |-
----------------|
                  |                   |                          |
     |
                  |                   |                          |
     |
                  V                   |                          V
     |
              ...............         |                 ..................
     |
              : MAN         :         |                 : IP Backbone    :
     V
   CPE       PE          PE :         |                PE              PE
..............
   |----|   |-----|     |-----|       |              |-----|          |-----|
: DC Network :
   | NS1o---|o---o|.--- |o---o|-------|-------------|o---o|----------|o---o|-----
o          :
   | NS2o---|o---o|.\-- |o---o|-------|-------------|o---o|----------|o---o|-----
o          :
   |----|   |-----| \\  |-----|       |              |-----|          |-----|
:............:
              :      \\     :         |                 :                :
              :.......\\....:         |                 :................:
                       \\             V
                        \\ .................
                         \ o Edge Cloud    :
                          \o               :
                           :               :
                           :               :
                           :...............:
```

Figure 1

A cloud-network service for enterprise will involve several domains, each with its own controller. MAN, Edge Cloud, IP Backbone and DC domains need to be coordinated in order to deliver a cloud-network service for enterprise.

In Figure 1, the network operator has created two E2E network slices, there are two types of traffic from the client, and each traffic is mapped to different slice, which is NS1 and NS2.Each NS with its own MAN, Edge Cloud, IP Backbone and DC network slices. The mechanism used to establish network slices in different domains and map the traffic to a network slice is outside the scope of this document.

3.2. The branch departments that use slices within the operator.

```
        |---------------|      |-----------------|       |-------------|
        | A network     |      |  Backbone Slice |       | N network   |
        | Controller    |      |   Controller    |       | Controller  |
        |---------------|      |-----------------|       |-------------|
               |                       |                       |
  |------------------------------------|--------------------------------|
  |            |    .---------------------------------------.  |        |
  |            |   /    IP Backbone  Network                 \ |        |
  |            |   \                                         / |        |
  |            |    `---------------------------------------'  |        |
  |            -------|                               |--------|        |
  |          .-------------.             .-------------.                |
  |         / sub-company A \           / sub-company N \               |
  |         \ network       / ......    \ network       /               |
  |          `-------------'             `-------------'                |
  |                                         Operator IP network         |
  |---------------------------------------------------------------------|
```

Figure 2

There are multiple sub-company network and IP Backbone network in an operator IP network, each with its own slice controller. Sub-company network can be the branches of the operator using slices.

IP Backbone network slice is orchestrated by the IP Backbone network orchestrator, and the path is calculated through the IP Backbone network slice controller.

For network slicing inside the local branch (sub-company network in the figure) is orchestrated through the orchestrator of the sub-company network. The sub-company network slice controller performs unified control and path calculation for the sub-company network. The path calculation and control of slices related to the IP Backbone are sent to the IP Backbone network slice controller through the eastbound and westbound interfaces, and the IP Backbone network slice controller controls and calculates the path.

3.2.1. Network Slice resource management

Resource Type	Orchestrator resource management
Slice ID	Unified resource orchestration and planning, plan Slice ID by sub-company. The orchestrator ensures that the IDs do not conflict with each other.
Node SID	Unified resource orchestration and planning. A unified coding mode is recommended.
SR Policy Color	Unified resource orchestration and planning, and resource pool allocation.
VPN name	Unified resource orchestration and planning. Perform unified resource conflict detection. VPN name within the same network element shall not be repeated.
VLAN sub-intf	Unified resource orchestration and planning: Resources are divided for VLAN sub-interfaces under the same physical interface.

3.2.2. Domain governance of network slice

```
    |----------------------------------------------------------------------------
------------------|
    |
                  V
    |                                |-------------------|
                .---------.
 .--------.      .---------.         |                   V
               / Operation\
/ Operation\    / Role       \-------|       . --------------------------.
               \   Set    /
\   Set    /    \ management/              . Security        System       .
     .---------. `--------' .---------.
 `--------'      `---------'              .  administrator   administrator .
    / Role Set \   ...... / Role Set \
     A             A      |              /                                  \----
--> \ A       /          \ N        /
     |             |      |--------|    \   Maintainer  Operator   monitor /
     ` -------'            ` -------'
     |             |               V       .                              .
```

```
         /\                /  |
     |         .---------.    .----------.   ` --------------------------'
        /  \             /    |
     |--------/ Operation\   / User      \      .------------------------.
       V    V          /      V
              \   Set    /   \ management/---> .  All user    User Group  .
 .-----.    .-----.  V      .-----.
               `________'     `________'      /                            \-->
/ User  \  / User  \...... / User  \
                                              \   Current                  /
\ A     /  \ B     /       \ N     /
                                                . Login User  Locked User .
 ` ____'    ` ____'         ` ____'
                                                  `_______________________'
     |         /    \           |

     V        V      \          V
                                                                              .--
-----------.          V  .-------------.
                                                                             / su
b-company A \  ......   / sub-company N \
                                                                             \ ne
twork       /           \ network       /
                                                                              `__
___________'             `_____________'
```

Role-based user rights management uses the role template to quickly allocate user rights, and provides network resources and sub-network slice resources for different users.

4. Security Considerations

TBD

5. IANA Considerations

This document does not have any requests for IANA allocation. This section may be removed before the publication of the draft.

6. Normative References

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[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>.

Authors' Addresses

Weiqiang Cheng
China Mobile
Beijing
CN

Email: chengweiqiang@chinamobile.com

Wenying Jiang
China Mobile
Beijing
CN

Email: jiangwenying@chinamobile.com

Ran Chen
ZTE Corporation

Email: chen.ran@zte.com.cn

Liyan Gong
China Mobile
Beijing
CN

Email: gongliyan@chinamobile.com

Chi Fan
H3C Corporation

Email: fanchi@h3c.com

Shaofu Peng
ZTE Corporation

Email: peng.shaofu@zte.com.cn

TEAS
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

LM. Contreras
Telefonica
R. Rokui
Nokia
J. Tantsura
Apstra
B. Wu
Huawei
X. Liu
Volta
D. Dhody
Huawei
S. Belloti
Nokia

February 22, 2021

IETF Network Slice Controller and its associated data models
draft-contreras-teas-slice-controller-models-01

Abstract

This document describes the major functional components of an IETF Network Slice Controller (NSC) as well as references the data models required for supporting the requests of IETF network slices and their realization.

Status of This Memo

This Internet-Draft will expire on August 26, 2021.

Copyright Notice

Table of Contents

1. Introduction

Editor's Note: the terminology in this draft will be aligned with the final terminology selected for describing the notion of IETF Network Slice when applied to IETF technologies, which is currently under discussion. By now same terminology as used in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework] is primarily used here. Consensus to use "IETF Network Slice" term has been reached.

The generic idea of network slicing intends to provide tailored end-to-end network capabilities to customers in the way that they could be perceived as a dedicated network, despite the fact that it makes use of shared physical infrastructure facilities.

Among the capabilities mentioned, connectivity of different parts of a network slice with particular characteristics play a central role. Thus, the concept of IETF Network Slice, realized by any of the IETF technologies, emerges as complementary but essential part of an end-to-end network slice.

In order to facilitate the request, realization and lifecycle control and management of a transport slice, a new element named IETF Network Slice Controller (NSC) is being proposed in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework].

The NSC from its North Bound Interface (NBI) exposes set of APIs that allow a higher level system to request an end-to-end transport slice. It receives the request of enablement of an IETF Network Slice by a customer (i.e. creation, modification or deletion). Upon receiving a request from its NBI, NSC finds the resources needed for realization of the IETF Network Slice and in turn interfaces from its South Bound Interface (SBI) with one or more Network Controllers for the realization of the requested IETF Network Slice request and the management of its lifecycle. Figure 1 presents a high-level view of the TSC.

```
                +------------------------------------------+
                |          A higher level system           |
                |   (e.g E2E network slice orchestrator)   |
                +------------------------------------------+
                                     A
                                     | NSC NBI
                                     V
                +------------------------------------------+
                |   IETF Network Slice Controller (NSC)    |
                +------------------------------------------+
                                     A
                                     | NSC SBI
                                     V
                +------------------------------------------+
                |           Network Controller(s)          |
                +------------------------------------------+
```

Figure 1: Interface of Transport Slice Controller

This memo describes the characteristics of the NSC as well as a detailed structure of the NSC and its major components. In addition, it describes the characteristics of the data models to identify the IETF Network Slice and its realization. Then the data models referred are mapped to the interfaces among components.

2. IETF Network Slice data models

At the time of provisioning and operating IETF Network Slices different views can be identified as necessary:

- Customer's view, mostly focused on the individual IETF Network Slice request process, reflecting the needs of each particular customer, including SLOs and other characteristics of the slice relevant for it. This view is technology agnostics and describes the characteristics of the IETF Network Slice from a customer's point of view. It can include the slice topology, performance parameters, endpoints of the slice, traffic characteristics of the slice, and the KPIs to monitor the slice.

- Provider's view, mostly focused on the provisioning and operation of the IETF Network Slices in the transport network, considering how a particular IETF Network Slice interplays with other IETF Network Slices maintained by the provider on a shared infrastructure. In other words, operator's view shows how an IETF Network Slice is realized in operator's network along with all the resources used during the its realization.

Both views are complementary, each of them specialized for a given purpose. In consequence, it should be consistency between both in order to ensure alignment.

Currently there are two different models proposed, on for each of the categories above. The model in [I-D.wd-teas-ietf-network-slice-nbi-yang] fits into the customer view, while the model defined in [I-D.liu-teas-transport-network-slice-yang] fits in to the provider view.

It should be noted that for the realization of a transport slice, the NSC interacts with one or more Network Controllers. In that case, the data models to be used are particular for each Network Controller (e.g., technology dependent), as well as the mapping function from its NBI to SBI and the details of this mapping function are both out of the scope of this document.

3. Structure of the IETF Network Slice Controller (NSC)

The NSC should work with both data models. The NSC takes first the customer's view by analyzing the needs of the customer, processing such requests taking into account the overall view of the network and the IETF Network Slices already instantiated, normalizing its instantiation across different technologies, and finally generates the provider view.

Once the new request is processed and declared as feasible, the NSC triggers its realization by interacting with the Network Controllers and communicates back to the higher level controller to start the billing cycle.

In order to accommodate these procedures, the internal structure of the NSC can be divided into:

- IETF Network Slice Mapper: this high-level component processes the customer request, putting it into the context of the overall IETF Network Slices in the network.

- IETF Network Slice Realizer: this high-level component processes the complete view of transport slices including the one requested by the customer, decides the proper technologies for realizing the IETF Network Slice and triggers its realization.

Figure 2 illustrates the components described and the associated models, as follows

- (a) -> customer's view, e.g. [I-D.wd-teas-ietf-network-slice-nbi-yang].

- (b) -> provider's view, e.g. [I-D.liu-teas-transport-network-slice-yang].

- (c) -> models per network controller, out of scope of this document

```
    Higher Level System
             |
             |
---------------------------
| NSC        | (a)        |
|            v            |
|   -------------------   |
|   |                 |   |
|   |    NS Mapper    |   |
|   |                 |   |
|   -------------------   |
|            | (b)        |
|            v            |
|   -------------------   |
|   |                 |   |
|   |    NS Realizer  |   |
|   |                 |   |
|   -------------------   |
|            | (c)        |
---------------------------
             |
             v
    Network Controllers
```

Figure 2: IETF Network Slice Controller structure and asspociated data models

IETF Network Slices with different level of detail could be requested:

- The IETF network slice can be abstracted as a set of edge-to-edge links (Type 1).

- The IETF network slice can be abstracted as a topology of virtual nodes and virtual links (Type 2) which represent the partitioning of underlay network resources for use by network slice connectivity.

The use cases of these two types of networks are further described by [RFC8453]. [I-D.wd-teas-ietf-network-slice-nbi-yang] models the Type 1 service, while [I-D.liu-teas-transport-network-slice-yang] models the Type 2 service. When a customer intends to request a Type 2 service, [I-D.liu-teas-transport-network-slice-yang] can also be used at the point (a) in Figure 2. As an example, when ACTN is used to realize an IETF network slice, model mappings are described in more details in [I-D.ietf-teas-actn-yang].

3.1. NS Mapper

The Mapper will receive the IETF Network Slice request from the customer. It will process it obtaining an overall view of how this new request complements or fits with the rest of IETF Network Slices, if any, as provisioned in the network. As part of that processing, a single customer IETF Network Slice request could result in the need of actually provisioning different IETF Network Slices in the network. The Mapper will maintain the relationship among customer IETF Network Slice request and provisioned IETF Network Slices.

3.2. NS Realizer

The Realizer will receive from the Mapper one or more requests for provision of IETF Network Slices, potentially including some technology-specific information. With that information, the Realizer will determine the realization of each particular IETF Network Slice interacting with technology-specific Network Controllers.

4. Model types in IETF Network Slice Controller interfaces

Both [RFC8309] and [RFC8969] offer a complete view of customer, service and network model types. In this sense a potential mapping of models to IETF Network Slcie Controller interfaces is as follows:

o NBI of the IETF NSC (interface (a) in Figure 2) -> Customer service model. According to [RFC8309] "a customer's service request is (or should be) technology agnostic. That is, a customer is unaware of the technology that the network operator has available to deliver the service, so the customer does not make requests specific to the underlying technology but is limited to making requests specific to the service that is to be delivered". This definition matches the expected behavior of the IETF NSC NBI as considered in in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework].

o Interface between NS Mapper and NS Realizer (interface (b) in Figure 2) -> Service Delivery model. According to [RFC8309] "a service delivery module is expressed as a core set of parameters that are common across a network type and technology [...] Service delivery modules include technology-specific modules.". Furthermore, [RFC8969] (in its Figures 3 and 5) considers L3SM or VN Service models to be later on fed into a controller.

o SBI of the IETF NSC (interface (c) in Figure 2) -> Network Configuration model. According to [RFC8309] "the orchestrator must map the service request to its view, and this mapping may include a choice of which networks and technologies to use depending on which service features have been requested". This is coincideent with the expected behavior of the IETF NSC SBI as considered in in [I-D.nsdt-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework].

5. Security Considerations

To be done.

6. IANA Considerations

This draft does not include any IANA considerations

7. References

[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-06 (work in progress), August 2020.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-02 (work in progress), November 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-02 (work in progress), December 2020.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

[I-D.wd-teas-ietf-network-slice-nbi-yang]
Bo, W., Dhody, D., Han, L., and R. Rokui, "A Yang Data Model for IETF Network Slice NBI", draft-wd-teas-ietf-network-slice-nbi-yang-01 (work in progress), November 2020.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/info/rfc8309>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/info/rfc8969>.

Authors’ Addresses

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com/

Reza Rokui
Nokia
Canada

Email: reza.rokui@nokia.com

Jeff Tantsura
Apstra
USA

Email: jefftant.ietf@gmail.com

Bo Wu
Huawei Technologies
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: lana.wubo@huawei.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Sergio Belloti
Nokia

Email: sergio.belotti@nokia.com

TEAS Working Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

J. Dong
Z. Li
Huawei Technologies
F. Qin
China Mobile
G. Yang
China Telecom
J. Guichard
Futurewei Technologies

February 22, 2021

Scalability Considerations for Enhanced VPN (VPN+)
draft-dong-teas-enhanced-vpn-vtn-scalability-02

Abstract

Enhanced VPN (VPN+) aims to provide enhancements to existing VPN services to support the needs of new applications, particularly including the applications that are associated with 5G services. VPN+ could be used to provide network slicing, and may also be of use in more generic scenarios, such as enterprise services which have demanding requirement. With the requirement for VPN+ services increase, scalability would become an important factor for the deployment of VPN+. This document describes the scalability considerations in the control plane and data plane to enable VPN+ services, some optimization mechanisms are also described.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Virtual Private Networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated connectivity over a common network infrastructure. The VPN service is provided with two network layers: the overlay and the underlay. The underlay is responsible for establishing network connectivity and managing network resources to meet the service requirement. The overlay is used to distribute the membership and reachability information of the tenants, and provide logical separation of service delivery between different tenants.

Enhanced VPN service (VPN+) [I-D.ietf-teas-enhanced-vpn] is targeted at new applications which require better isolation between tenants and/or services, and have more stringent performance requirements than can be provided with existing VPNs. To meet the requirement of VPN+ services, Virtual Transport Networks (VTN) need to be created, each has a subset of the underlay network topology and a set of network resources allocated to meet the requirements of one or a group of VPN+ services. The VPN together with the corresponding VTN in the underlay provide the VPN+ service.

[I-D.ietf-teas-enhanced-vpn] provides some general analysis of the scalability of VPN+. This document gives detailed analysis of the scalability considerations when enabling VPN+ services. The focus of this document is mainly on the scalability of the underlay of VPN+, i.e. the VTN.

2. VPN+ Scalability Requirements

As described in [I-D.ietf-teas-enhanced-vpn], VPN+ services may require additional state to be introduced into the network to take advantage of the enhanced functionality. This introduces some scalability considerations to the network. This section gives some analysis of the number of VPN+ services that might be needed in a network.

There are several use cases where VPN+ may be needed, and these determine how many VPN+ will be required in a network. One typical use case of VPN+ is to deliver IETF network slice [I-D.ietf-teas-ietf-network-slice-definition] for applications or services in 5G and other scenarios, thus the number of IETF network slices needed could reflect the number of VPN+ services. With the development and evolution of 5G, it is expected that more and more network slices will be deployed. The number of network slices required is relevant to how network slicing will be used, and the progress of 5G for the vertical industrial services. The potential number of network slices is analyzed by classifying the network slicing deployment into three typical scenarios:

1. Network slicing can be used by a network operator internally to isolate different types of services. For example, in a converged multi-service network, different network slices can be created to carry mobile transport service, fixed broadband service and enterprise services respectively, each type of service could be managed by a separate department or management team. Some service types, such as multicast service may also be deployed in a dedicated network slice. It is also possible that an infrastructure network operator provides network slices to other network operators as a wholesale service. In this scenario, the

number of network slices in a network would be relatively small, such as on the order of 10 or so. This could be the typical case in the beginning of the network slicing deployment.

2. Network slicing can be used to provide isolated and customized virtual networks for tenants in different vertical industries. At the early stage of the vertical industrial service deployment, a few top tenants in some typical industries will begin to use network slicing to support their business, such as smart grid, manufacturing, public safety, on-line gaming, 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 the order of 100.

3. With the evolution of 5G, network slicing could be widely used by both vertical industrial tenants and enterprise tenants which require guaranteed or predictable service performance. The total amount of network slices may increase to the order of 1000 or more. However, it is expected that the number of network slices would still be less than the number of traditional VPN services in the network.

In 3GPP [TS23501], a 5G network slice is identified using Single Network Slice Selection Assistance Information (S-NSSAI), which is a 32-bit identifier comprised of 8-bit Slice/Service Type (SST) and 24-bit Slice Differentiator (SD). This allows the mobile networks (RAN and CN) to provide a large number of network slices. Although it is possible that multiple network slices in RAN and CN can be mapped to the same IETF network slice, the number of IETF network slices may still be comparable with the number of 5G network slices. Thus the scalability of IETF network slices needs to be taken into consideration.

```
                   8-bit              24-bit
              +------------+-------------------------+
              |    SST     |   Slice Differentiator  |
              +------------+-------------------------+
```

Figure 1. Format of S-NSSAI in 3GPP

VPN+ needs to meet the scalability requirement of network slicing in different scenarios. The increased number of VPN+ will introduce additional complexity and overhead to both the control plane and data plane, especially in the aspects related to the underlying VTNs. Although multiple VPN+ services can be mapped to the same VTN as the underlay, there still can be scalability challenges with the increased number of VTNs.

3. VPN+ Scalability Considerations

In this section, the scalability in the control plane and data plane is analyzed to understand the possible gaps in meeting the scalability requirement of VPN+.

3.1. Control Plane Scalability

As described in [I-D.ietf-teas-enhanced-vpn], the control plane of VPN+ could be based on the hybrid of a centralized controller and the distributed control plane.

3.1.1. Distributed Control Plane

At part of the construction of VPN+ services, it is necessary to create different VTNs that provide customized topology and resource attributes. The attributes and state information of each VTN needs to be exchanged in the control plane. The scalability of the distributed control plane for the establishment and maintenance of VTNs needs to be considered in the following aspects:

o The number of control protocol instances maintained on each node

o The number of protocol sessions maintained on each link

o The number of routes advertised by each node

o The amount of attributes associated with each route

o The number of route computation (i.e. SPF computation) executed on each node

As the number of VTNs increases, it is expected that for some of the above aspects, the overhead in the control plane may increase dramatically. For example, the overhead of maintaining separated control protocol instances (e.g. IGP instances) for different VTNs is considered higher than maintaining the information of separated VTNs in the same control protocol instance, and the overhead of maintaining separate protocol sessions for different VTNs is considered higher than using a shared protocol session for the information exchange of multiple VTNs. To meet the requirement of the increasing number of VTNs, It is suggested to choose the control plane mechanisms which could improve the scalability while still provide the required functionality.

3.1.2. Centralized Control Plane

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

To provide global optimization for the Traffic Engineered (TE) paths in different VTNs, the controller needs to keep the topology and resource information of all the VTNs 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, or multiple VTNs requires global optimization concurrently, 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.

3.2. Data Plane Scalability

To provide different VPN+ services with the required isolation and performance characteristics, it is necessary to allocate different sets of network resources to different VTNs. As the number of VPN+ increases, the number of VTNs will increase accordingly. This requires the underlying network to provide finer-granular network resource partitioning, which means the amount of state about the reserved network resources to be maintained on network nodes will also increase.

In data plane, traffic of different VPN+ services need to be processed separately according to the topology and resource constraints of the associated VTN , thus the identifier of VTN needs to be carried either directly or implicitly in the data packet. Different representations of the VTN information in data packet can have different scalability implications.

One approach is to reuse some existing fields in the data packet to additionally identify the VTN the packet belongs to. This avoids the cost of defining new fields in the data packet, while since it introduces additional semantics to an existing field, it may change the processing of the existing field in packet forwarding. To distinguish different VTNs, the number of identifiers which were used to identify a node or link may be increased in proportion to the number of the VTNs, which may cause scalability problem in some networks.

An alternative approach is to introduce a dedicated field in the packet for VTN identification. This could avoid the impact to the existing fields in the packet. And if this new field carries a global-significant VTN identifier, it could be used together with the existing fields to determine the VTN-specific packet forwarding. The potential issue with this approach is the difficulty in introducing a new field in some types of the data plane.

In addition, the introduction of per VTN packet forwarding has impact on the scalability of the forwarding entries on network nodes, as a network node needs to maintain separate forwarding entries for a target node in each VTN it participates.

3.3. Gap Analysis of Existing Mechanisms

One candidate approach to build VTN is to use Segment Routing (either SR-MPLS or SRv6) as the data plane, and define and distribute the customized topology and resource attribute of each VTN based on Multi-topology [RFC4915] [RFC5120], Flex-Algo [I-D.ietf-lsr-flex-algo] or the combination of these mechanisms in the control plane. As the number of VTNs increases, there may be several scalability concerns with this approach:

1. The number of SR SIDs needed will increase dependent upon the number of VTNs in the network, which will bring challenges both to the SID information distribution in the control plane and to the installation of forwarding entries for the SIDs in data plane.

2. The number of SPF computation will increase in proportion to the number of VTNs in the network, which can introduce significant overhead of the computing resources on network nodes.

3. The maximum number of network topology supported by OSPF Multi-topology is 128, and the maximum number of Flex-Algo is 128, which may not meet the required number of VTNs in some networks.

4. Possible Scalability Optimizations

4.1. Control Plane Optimizations

For the distributed control plane, several optimizations can be considered to reduce the overhead and improve the scalability.

The first optimization mechanism is to reduce the amount of control plane sessions used for the establishment and maintenance of the VTNs. For multiple VTNs which have the same peering relationship between two adjacent network nodes, it is proposed that one single control session is used for the establishment of multiple VTNs. Information of different VTNs can be exchanged over the same control session, with necessary identification information to distinguish them in the control messages. This could reduce the overhead of maintaining a large number of control protocol sessions, and could also reduce the amount of control plane message flooding in the network.

The second optimization mechanism is to decompose the attributes of a VTN into different groups, so that different types of attribute can be advertised and processed separately in control plane. For a VTN, there are two basic types of attributes: the topology attribute and the associated network resource attribute. In a network, it is possible that multiple VTNs share the same topology, and multiple VTNs may share the same set of network resource on particular network segments. It is more efficient if only one copy of the topology attribute is advertised, then multiple VTNs sharing the same topology could refer to the topology information. More importantly, the result of topology-based route computation could be shared by these VTNs, so that the overhead of per-VTN route computation could be reduced. Similarly, information of a subset of network resources reserved on network segments could be advertised once and then be used by multiple VTNs. This methodology could also apply to other attributes of VTN which may be introduced later and can be processed independently.

```
                  O#####O#####O          O*****O*****O
                  #     #     #          *     *     *
                  #     #     #          *     *     *
                  O#####O#####O          O*****O*****O

                      VTN-1                  VTN-2

                             O-----O-----O
                             |     |     |
                             |     |     |
                             O-----O-----O

                         Shared Network Topology

       Legend

       O     Virtual node
       ###   Virtual links with a set of reserved resources
       ***   Virtual links with another set of reserved resources
```

Figure 2. Topology Sharing between VTNs

FIG-2

Figure 2 gives an example of multiple VTNs which share the same topology attribute. As shown in the figure, VTN-1 and VTN-2 have the same topology, while the link resource attributes of each VTN are different. In this case, only one copy of the network topology information needs to be advertised, and the topology-based route computation result can be used by both VTNs to generate the routing tables.

```
                    O#####O#####O         O-  -O#####O
                    #     #     #           \/ #     #
                    #     #     #           /\ #     #
                    O#####O#####O         O-  -O#####O

                        VTN-1                VTN-2

        Legend

        O     Virtual node
        ###   Virtual links with a set of reserved resource
        ---   Virtual links with another set of reserved resource
```

Figure 3. Resource Sharing between VTNs

Figure 3 gives another example of multiple VTNs which shares the same set of network resources on some links. In this case, information about the reserved resource on each link only needs to be advertised once, then both VTN-1 and VTN-2 could refer to the link resource for constraint based path computation.

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

4.2. Data Plane Optimizations

To support more VPN+ services while keeping the amount of data plane state at a reasonable scale, one possible approach is to classify a set of VPN+ services which have similar service characteristics and performance requirements into a group, and such group of VPN+ is mapped to one VTN, which is allocated with an aggregated set of network topology and resources to meet the service requirement of the whole group of VPN+. Different groups of VPN+ need to be mapped to different VTNs with different set of network resources allocated. With appropriate grouping of VPN+ services, a reasonable number of

VTNs with network resources reservation and aggregation could still meet the service requirements.

Another optimization in the data plane is to decouple the identifier used for topology-based forwarding and the identifier used for the resource-specific processing introduced by VTN. One possible mechanism is to introduce a dedicated field in the packet header to uniquely identify the set of local network resources allocated to a VTN on each network node for the processing and forwarding of the received packet. Then the existing identifier in the packet header used for topology based forwarding is kept unchanged. The benefit is the number of existing topology-specific identifiers will only increase in proportion to the number of topologies rather than the number of VTNs, so that its scalability will not be impacted by the increase of VTN. Since this new VTN field will be used together with the existing fields to determine the VTN-specific packet forwarding, this probably requires network nodes to support a hierarchical forwarding table in the data plane. Figure 4 shows the concept of using different data plane identifiers for topology-based and VTN resource-based packet processing respectively.

```
                    +--------------------------+
                    |       Packet Header      |
                    |                          |
                    | +----------------------+ |
                    | | Topology-specific ID | |
                    | +----------------------+ |
                    |                          |
                    | +----------------------+ |
                    | |    VTN Resource ID   | |
                    | +----------------------+ |
                    +--------------------------+
```

Figure 4. Decoupled Data Plane Identifiers

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or one of the extension headers to carry the VTN identifier for the resource-specific forwarding, while keeping the destination IP address field used for routing towards the destination prefix in the corresponding topology. Note that the VTN ID needs to be parsed by every node along the path which is capable of VTN-specific forwarding. In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated MPLS label to identify the VTN instance, while the existing MPLS labels could be used for topology-based packet forwarding towards the associated destination prefix. This requires that both labels be parsed by each node along the forwarding path of the packet. Another option with MPLS data plane is to introduce a new VTN header which follows the MPLS label stack. The detailed extensions in IPv6 and MPLS encapsulation are out of the scope of this document.

5. Solution Evolution for Improved Scalability

Based on the analysis in this document, the control plane and data plane for VPN+ needs to evolve to support the increasing number of VPN+ services in the network.

As the first step, by introducing resource-awareness to segment routing SIDs [I-D.ietf-spring-resource-aware-segments], and using Multi-Topology or Flex-Algo as the control plane, it could provide a solution for building a limited number of VTNs in the network to meet the requirement of a small number of VPN+ services in the network. This mechanism is considered as the basic SR VTN.

As the number of required VPN+ services increases, more VTNs may need to be created, then the control plane scalability could be improved by decoupling the topology attribute from other attributes (e.g. resource attribute) of VTN, so that multiple VTNs could share the same topology or resource attribute. This mechanism is considered as the optimized SR VTN. Both the basic and the optimized SR VTN mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

If the data plane scalability becomes a concern, dedicated data plane VTN identifiers can be introduced to decouple the topology-specific identifiers from the VTN-specific resource identifier in the data plane, this could help to reduce the number of SR SIDs needed to support . This mechanism is considered as resource-independent VTNs.

6. Security Considerations

TBD

7. IANA Considerations

This document makes no request of IANA.

8. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

9. Acknowledgments

The authors would like to thank Adrian Farrel for the review and discussion of this document.

10. Informative References

[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-13 (work in progress), October 2020.

[I-D.ietf-spring-resource-aware-segments]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Introducing Resource Awareness to SR Segments", draft-ietf-spring-resource-aware-segments-01 (work in progress), January 2021.

[I-D.ietf-spring-sr-for-enhanced-vpn]
Dong, J., Bryant, S., Miyasaka, T., Zhu, Y., Qin, F., Li, Z., and F. Clad, "Segment Routing based Virtual Transport Network (VTN) for Enhanced VPN", February 2021, <https://tools.ietf.org/html/draft-ietf-spring-sr-for-enhanced-vpn>.

[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-06 (work in progress), July 2020.

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[TS23501] "3GPP TS23.501", 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

Authors' Addresses

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: jie.dong@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: lizhenbin@huawei.com

Fengwei Qin
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing
China

Email: qinfengwei@chinamobile.com

Guangming Yang
China Telecom
No.109 West Zhongshan Ave., Tianhe District
Guangzhou
China

Email: yangguangm@chinatelecom.cn

James N Guichard
Futurewei Technologies
2330 Central Express Way
Santa Clara
USA

Email: james.n.guichard@futurewei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

X. Geng
J. Dong
Huawei Technologies
R. Pang
China Unicom
L. Han
China Mobile
T. Niwa
Individual
J. Jin
LG U+
C. Liu
China Unicom
N. Nageshar
Individual

February 22, 2021

5G End-to-end Network Slice Mapping from the view of Transport Network
draft-geng-teas-network-slice-mapping-03

Abstract

Network Slicing is one of the core featrures in 5G. End-to-end network slice consists of 3 major types of network segments: Access Network (AN), Mobile Core Network (CN) and Transport Network (TN). This draft describes the procedure of mapping 5G end-to-end network slice to transport network slice defined in IETF. This draft also intends to expose some gaps in the existing network management plane and data plane technologies to support inter-domain network slice mapping. Further work may require cooperation between IETF and 3GPP (or other standard organizations). Data model specification, signaling protocol extension and new encapsulation definition are out of the scope of this draft.

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 will expire on August 26, 2021.

Copyright Notice

Table of Contents

1. Introduction

Driven by the new applications of 5G, the concept of network slicing is defined to provide a logical network with specific capabilities and characteristics. Network slice contains a set of network functions and allocated resources(e.g. computation, storage and network resources). According to [TS28530], a 5G end-to-end network slice is composed of three major types network segments: Radio Access Network (RAN), Transport Network (TN) and Mobile Core Network (CN). Transport network is supposed to provide the required connectivity between AN and CN, with specific performance commitment. For each end-to-end network slice, the topology and performance requirement for transport network can be very different, which requests transport network to have the capability of supporting multiple different transport network slices.

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). A transport network slice could span multiple technology (IP, Optical) and multiple administrative domains. Depending on the consumer’s requirement, a transport network slice could be isolated from other concurrent transport network slices, in terms of data plane, control plane and management plane. Transport network slice is being defined and discussed in IETF.

Editor’s Note: The definition of transport network slice will align with [I-D.ietf-teas-ietf-network-slice-definition].

The procedure of end-to-end network slice instance creation, network slice subnet instance creation and network slice instance termination in management plane is defined in [TS28531]. The end-to-end network slice allocation is defined in ETSI [ZSM003]. But there is no specifications about how to map end-to-end network slice in 5G system to transport network slice. This draft describes the procedure of mapping 5G end-to-end network slice into transport network slice in management plane, control plane and user plane.

5G end-to-end network slice mapping is treated as an independent mechanism from 5G end-to-end QoS mapping. The latter is not covered by this version.

2. Terminologies

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].

The following terms are used in this document:

NS: Network Slice

NSI: Network Slice Instance

NSSI: Network Slice Subnet Instance

NSSAI: Network Slice Selection Assistance Information

S-NSSAI: Single Network Slice Selection Assistance Information

AN: Access Network

RAN: Radio Access Network

TN: Transport Network

CN: Mobile Core Network

DSCP: Differentiated Services Code Point

CSMF: Communication Service Management Function

NSMF: Network Slice Management Function

NSSMF: Network Slice Subnet Management Function

GST: General Slice Template

TNSII: Transport Network Slice Interworking Identifier

TNSI: Transport Network Slice Identifier

PDU: Protocol Data Unit

Editor’s Note: Terminologies defined in 3GPP, e.g.,Network Slice Subnet Management Function(NSSMF), Network Slice Subnet Instance(NSSI) and Network Slice Selection Assistance Information(NSSAI), are used in the end-to-end network slice mapping, which may not be used necessarily within the transport network.

3. Network Slice Mapping Structure

The following figure shows the necessary elements for mapping end-to-end network slice into transport network slice. All these network slice elements are classified into three groups: requirements/ capabilities, identifiers and relevant functions.

```
                   +-----------------+
                   |       CSMF      |
                   +--------+--------+
                            |
                   +--------V--------+
                   |       NSMF      |
                   +-----------------+
        +----------|  NSI Identifier |----------+
        |          | Service Profile |          |
        |          |  TN  Network-   |          |
        |          | -Slice Profile  |          |
        |          +-----------------+          |
        |                   |                   |
 +------V------+ +----------V----------+ +------V------+
 |  AN NSSMF   | |       TN NSSMF       | |  CN NSSMF   |
 +-------------+ +---------------------+ +-------------+
 |   AN-NSSI-  | |  TN-NSSI Identifier | |   CN-NSSI-  |
 | -Identifier | |  Function Management| | -Identifier |
 |     ...     | |         ...         | |     ...     |    Management
 +-------------+ +---------------------+ +-------------+      Plane
      |           |                   |           |      -----------------
      |<----------PDU session (S-NSSAI)---------->|           Control
      |           |                   |           |            Plane
      V           V                   V           V      -----------------
      /\       +-----+             +-----+    +-------+        Data
     /AN\ -----|  PE |-----...-----| PE  |----|  UPF  |        Plane
    /____\     +-----+             +-----+    +-------+
      |-->TNSII<--|------>TNSI<-------|-->TNSII<--|
```

3.1. Requirements Profile

In order to satisfy a tenant's request for a network slice with certain characteristics, creating a new network slice or using existing network slice instance is constrained by the requirement profile and the capability of the network slices.

- Service Profile: represents the properties of network slice related requirement that should be supported by the network slice instance in 5G network. Service profile is defined in [TS28541] 6.3.3.

- TN Network Slice Profile: represents the properties of transport network slice related requirement that should be supported by the transport network slice in a 5G network. Slice Profile is defined in [TS28541] 6.3.4. TN Network slice profile is newly defined in this draft.

3.2. Identifiers

Network slice related identifiers in management plane, control plane and data(user) plane play an important role in end-to-end network slice mapping.

- Single Network Slice Selection Assistance Information(S-NSSAI): end-to-end network slice identifier in control plane, which is defined in [TS23501];

- Network Slice Instance(NSI) Identifier:end-to-end network slice identifier in management plane, which is created in NSMF; NSI is is set of Network Function instances and the required resources (e.g. computing, storage and networking resources) which form a deployed Network Slice, which is defined in [TS23501]; ;

- Transport Network Slice Instance(TN-NSSI) Identifier: transport network slice identifier in management plane, which is created in TN NSSMF; TN-NSSI is newly defined in this draft.

- Transport Network Slice Interworking Identifier (TNSII): network slice identifier which is used for mapping end-to-end network slice into transport network slice in data plane. TNSII is a new concept introduced by this draft, which can be instantiated with existing data plane identifiers and doesn't necessarilly request new encapsulation. TNSII could be pre-allocated as a global identifier.

- Transport Network Slice Identifier(TNSI): transport network slice identifier in data plane(user plane). TNSI is newly defined in this draft.

The relationship between these identifiers are specifies in the following sections.

3.3. Relevant functions

There are a set of slice relevant functions that are necessary for transport network slice management:

- Topology management

- QoS management

- Resource management

- Measurement management

o ...

Some of these functions are implemented inside the transport network and independent from the end-to-end network slice, e.g., topology management, QoS management, resource management; Some of the functions are related to the end-to-end network slice and should cooperate with other network elements from other domain, e.g., Measurement management.

4. Network Slice Mapping Procedure

This section provides a general procedure of network slice mapping:

```
        +--------------------------------+
        |       Requirement Matching     |
        +---------------+----------------+
                        |
                        V
        +--------------------------------+
        |       NSI<->TN NSSI  Mapping   |
        +---------------+----------------+
                        |
                        V
        +--------------------------------+
        |       S-NSSAI Selection        |
        +---------------+----------------+
                        |
                        V
        +--------------------------------+
        |S-NSSAI<---------->TNSII Mapping|
        |      (NSI<->TN NSSI)           |
        +---------------+----------------+
                        |
                        V
        +--------------------------------+
        |       TNSII<->TNSI  Mapping    |
        +--------------------------------+
```

1. NSMF receives the request from CSMF for allocation of a network slice instance with certain characteristics.

2. Based on the service requirement , NSMF acquires requirements for the end-to-end network slice instance , which is defined in Service Profile([TS28541] section 6.3.3).

3. NSMF derives transport network slice related requirements from the Service profile, and maintains them in Transport Network Slice Profile, So as to CN Slice Profile and AN Slice Profile, in order to

decide on the constituent NSSIs(including AN NSSI, CN NSSI and TN NSSI) of the NSI, based on the service profile and the endpoint information(AN/CN edge nodes).

4. NSMF sends the Transport Network Slice Profile, endpoint information, along with other TS NBI attributes to TN NSSMF for TN NSSI allocation.

5. TN NSSMF allocates TN NSSI which could satisfy the requirement of Transport Network Slice Profile between the specified endpoints (AN/ CN edge nodes) and sends the TN NSSI Identifier to NSMF.

6. NSMF acquires the mapping relationship between NSI and TN NSSI.

7. NSMF matains the mapping relationship between NSI and S-NSSAI and the mapping relationship between TN NSSI and TNSII, which could be used to set up mapping relationship between S-NSSAI and TNSII.

8. When a PDU session is set up between AN and CN, an S-NSSAI is selected for the PDU session.

9. AN/CN edge nodes encapsulates the packet using TNSII, according to the selected S-NSSAI. Network Slice could also be differentiated by physical interface, if different network slices are transported through different interface;

10. The edge node of transport network parses the TNSII from the packet and maps the packet to the corresponding transport network slice. It may encapsulate packet with TNSI. The nodes in transport network transit the packet inside the corresponding transport network slice according to TNSI.

The procedure of end-to-end network slice mapping involves the mapping in three network planes: management plane, control plane and data plane.

4.1. Network Slice Mapping in Management Plane

The transport network management Plane maintains the interface between NSMF and TN NSSMF, which 1) guarantees that transport network slice could connect the AN and CN with specified characteristics that satisfy the requirements of communication; 2) builds up the mapping relationship between NSI identifier and TN NSSI identifier; 3) maintains the end-to-end slice relevant functions;

Service Profile defined in[TS28541] represents the requirement of end-to-end network slice instance in 5G network. Parameters defined in Service Profile include Latency, resource sharing level, availability and so on. How to decompose the end-to-end requirement to the transport network requirement is one of the key issues in Network slice requirement mapping. GSMA(Global System for Mobile Communications Association) defines the [GST] to indicate the network slice requirement from the view of service provider. [I-D.contreras-teas-slice-nbi] analysis the parameters of GST and categorize the parameters into three classes, including the attributes with direct impact on the transport network slice definition. It is a good start for selecting the transport network relevant parameters in order to define Network Slice Profile for Transport Network. Network slice requirement parameters are also necessary for the definition of transport network northbound interface.

Inside the TN NSSMF, it is supposed to maintain the attributes of the transport network slice. If the attributes of an existing TN NSSI could satisfy the requirement from TN Network Slice Profile, the existing TN NSSI could be selected and the mapping is finished If there is no existing TN NSSI which could satisfy the requirement, a new TN NSSI is supposed to be created by the NSSMF with new attributes.

TN NSSI resource reservation should be considered to avoid over allocation from multiple requests from NSMF (but the detailed mechanism should be out of scope in the draft)

TN NSSMF sends the selected or newly allocated TN NSSI identifier to NSMF. The mapping relationship between NSI identifier and TN NSSI identifier is maintained in both NSMF and TN NSSMF.

YANG data model for the Transport Slice NBI, which could be used by a higher level system which is the Transport slice consumer of a Transport Slice Controller (TSC) to request, configure, and manage the components of a transport slices, is defined in [I-D.wd-teas-transport-slice-yang]. The northbound Interface of IETF network slice refers to [I-D.wd-teas-ietf-network-slice-nbi-yang].

4.2. Network Slice Mapping in Control Plane

There is no explicit interaction between transport network and AN/CN in the control plane, but the S-NSSAI defined in [TS23501] is treated as the end-to-end network slice identifier in the control plane of AN and CN, which is used in UE registration and PDU session setup. In this draft, we assume that there is mapping relationship between S-NSSAI and NSI in the management plane, thus it could be mapped to a transport network slice .

Editor’s note: The mapping relationship between NSI defined in [TS23501] and S-NSSAI defined in [TS23501] is still in discussion.

4.3. Network Slice Mapping in Data Plane

If multiple network slices are carried through one physical interface between AN/CN and TN, transport network slice interworking identifier(TNSII) in the data plane needs to be introduced. If different network slices are transported through different physical interfaces, Network Slices could be distinguished by the interface directly. Thus TNSII is not the only option for network slice mapping, while it may help in introducing new network slices.

4.3.1. Data Plane Mapping Considerations

The mapping relationship between AN or CN network slice identifier (either S-NSSAI in control plane or NSI/NSSI in management plane) and TNSII needs to be maintained in AN/CN network nodes, and the mapping relationship between TNSII and TNSI is maintained in the edge node of transport network. When the packet of a uplink flow goes from AN to TN, the packet is encapsulated based on the TNSII; then the encapsulation of TNSII is read by the edge node of transport network, which maps the packet to the corresponding transport network slice.

Editor’s Note: We have considered to add "Network Instance" defined in [TS23501]in the draft. However, after the discussion with 3GPP people, we think the concept of "network instance" is a ’neither Necessary nor Sufficient Condition’ for network slice. Network Instance could be determined by S-NSSAI, it could also depends on other information; Network slice could also be allocated without network instance (in my understanding) And, TNSII is not a competitive concept with network instance.TNSII is a concept for the data plane interconnection with transport network, network instance may be used by AN and CN nodes to associate a network slice with TNSII

4.3.2. Data Plane Mapping Options

The following picture shows the end-to-end network slice in data plane:

```
+--+       +-----+                            +-----------------+
|UE|- - - -|(R)AN|----------------------------|       UPF       |
+--+       +-----+                            +-----------------+
 |<----AN NS---->|<-----------TN NS---------->|<----CN NS----->|
```

The mapping between 3GPP slice and transport slice in user plane could happens in:

(R)AN: User data goes from (radio) access network to transport network

UPF: User data goes from core network functions to transport network

Editor’s Note: As figure 4.7.1. in [TS28530] describes, TN NS will not only exist between AN and CN but may also within AN NS and CN NS. However, here we just show the TN between AN and CN as an example to avoid unncessary complexity.

The following picture shows the user plane protocol stack in end-to-end 5G system.

```
  +-----------+                    |                  |               |
  |Application+--------------------|------------------|---------------|
  +-----------+                    |                  | +-----------+ |
  | PDU Layer +--------------------|------------------|-| PDU Layer | |
  +-----------+   +-------------+  |  +-------------+ | +-----------+ |
  |           |   | ___Relay___ |--|--| ___Relay___ |-|-|           | |
  |           |   |     \/ GTP-U|--|--|GTP-U\/ GTP-U|-|-|   GTP-U   | |
  |   5G-AN   |   |5G-AN +------+  |  +------+------+ | +-----------+ |
  |  Protocol |   |Protoc|UDP/IP|--|--|UDP/IP|UDP/IP|-|-|   UDP/IP  | |
  |   Layers  |   |Layers+------+  |  +------+------+ | +-----------+ |
  |           |   |      |  L2  |--|--|  L2  |  L2  |-|-|     L2    | |
  |           |   |      +------+  |  +------+------+ | +-----------+ |
  |           |   |      |  L1  |--|--|  L1  |  L1  |-|-|     L1    | |
  +-----------+   +-------------+  |  +-------------+ | +-----------+ |
       UE              5G-AN       |        UPF       |      UPF      |
                                   N3                 N9              N6
```

The following figure shows the typical encapsulation in N3 interface which could be used to carry the transport network slice interworking identifier (TNSII) between AN/CN and TN.

```
   +-------------------------+
   | Application Protocols   |
   +-------------------------+
   |       IP (User)         |
   +-------------------------+
   |          GTP            |
   +-------------------------+
   |          UDP            |
   +-------------------------+
   |          IP             |
   +-------------------------+
   |       Ethernet          |
   +-------------------------+
```

4.3.2.1. Layer 3 and Layer 2 Encapsulations

If the encapsulation above IP layer is not visible to Transport Network, it is not able to be used for network slice interworking with transport network. In this case, IP header and Ethernet header could be considered to provide information of network slice interworking from AN or CN to TN.

```
   +-------------------------+-----------
   | Application Protocols   |      ^
   +-------------------------+      |
   |       IP (User)         |  Invisible
   +-------------------------+     for
   |          GTP            |     TN
   +-------------------------+      |
   |          UDP            |      V
   +-------------------------+------------
   |          IP             |
   +-------------------------+
   |       Ethernet          |
   +-------------------------+
```

The following field in IP header and Ethernet header could be considered :

IP Header:

- o DSCP: It is traditionally used for the mapping of QoS identifier between AN/CN and TN network. Although some values (e.g. The unassigned code points) may be borrowed for the network slice interworking, it may cause confusion between QoS mapping and network slicing mapping.;

- Destination Address: It is possible to allocate different IP addresses for entities in different network slice, then the destination IP address could be used as the network slice interworking identifier. However, it brings additional requirement to IP address planning. In addition, in some cases some AN or CN network slices may use duplicated IP addresses.

- Option fields/headers: It requires that both AN and CN nodes can support the encapsulation and decapsulation of the options.

Ethernet header

- VLAN ID: It is widely used for the interconnection between AN/CN nodes and the edge nodes of transport network for the access to different VPNs. One possible problem is that the number of VLAN ID can be supported by AN nodes is typically limited, which effects the number of transport network slices a AN node can attach to. Another problem is the total amount of VLAN ID (4K) may not provide a comparable space as the network slice identifiers of mobile networks.

Two or more options described above may also be used together as the TNSII, while it would make the mapping relationship more complex to maintain.

In some other case, when AN or CN could support more layer 3 encapsulations, more options are available as follows:

If the AN or CN could support MPLS, the protocol stack could be as follows:

```
   +------------------------+-----------
   | Application Protocols  |      ^
   +------------------------+      |
   |       IP (User)        |  Invisible
   +------------------------+     for
   |          GTP           |     TN
   +------------------------+      |
   |          UDP           |      V
   +------------------------+------------
   |         MPLS           |
   +------------------------+
   |          IP            |
   +------------------------+
   |       Ethernet         |
   +------------------------+
```

A specified MPLS label could be used to as a TNSII.

If the AN or CN could support SRv6, the protocol stack is as follows:

```
+------------------------+-----------
| Application Protocols  |      ^
+------------------------+      |
|       IP (User)        |  Invisible
+------------------------+     for
|          GTP           |     TN
+------------------------+      |
|          UDP           |      V
+------------------------+------------
|          SRH           |
+------------------------+
|         IPv6           |
+------------------------+
|       Ethernet         |
+------------------------+
```

The following field could be considered to identify a network slice:

SRH:

- o SRv6 functions: AN/CN is supposed to support the new function extension of SRv6.

- o Optional TLV: AN/CN is supposed to support the extension of optional TLV of SRH.

4.3.2.2. Above Layer 3 Encapsulations

If the encapsulation above IP layer is visible to Transport Network, it is able to be used to identify a network slice. In this case, UPD and GTP-U could be considered to provide information of network slice interworking between AN or CN and TN.

```
+------------------------+----------
| Application Protocols  |     |
+------------------------+ Invisible
|       IP (User)        |    for
+------------------------+    TN
|          GTP           |     |
+------------------------+------------
|          UDP           |
+------------------------+
|          IP            |
+------------------------+
|       Ethernet         |
+------------------------+
```

The following field in UDP header could be considered:

UDP Header:

- o UDP Source port: The UDP source port is sometimes used for load balancing. Using it for network slice mapping would require to disable the load-balancing behavior.

5. Network Slice Mapping Summary

The following picture shows the mapping relationship between the network slice identifier in management plane, control plane and user plane.

```
                 AN/CN            |              TN
   Management +---------+         |        +---------+
     Plane    |   NSI   |<--------|------->| TN NSSI |
              +---------+         |        +---------+
                   |              |             |
                   |              |             |
    Control  +-----V-----+        |  +----------+----------+
     Plane   |  S-NSSAI  |        |  |                     |
             +-----------+        |  |                     |
                   |            +----V----+           +----V----+
                   +----------->|  TNSII  |<--------->|   TNSI  |
     User                       |  /Port  |<--------->|         |
     Plane                      +---------+           +---------+
```

6. IANA Considerations

TBD

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 Shunsuke Homma for reviewing the draft and giving valuable comments.

9. Normative References

[GST] "Generic Network Slice Template", <https://www.gsma.com/newsroom/all-documents/generic-network-slice-template-v2-0/>.

[I-D.contreras-teas-slice-nbi] Contreras, L., Homma, S., and J. Ordonez-Lucena, "IETF Network Slice use cases and attributes for Northbound Interface of controller", draft-contreras-teas-slice-nbi-03 (work in progress), October 2020.

[I-D.ietf-teas-ietf-network-slice-definition] Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[I-D.wd-teas-ietf-network-slice-nbi-yang] Bo, W., Dhody, D., Han, L., and R. Rokui, "A Yang Data Model for IETF Network Slice NBI", draft-wd-teas-ietf-network-slice-nbi-yang-01 (work in progress), November 2020.

[I-D.wd-teas-transport-slice-yang] Bo, W., Dhody, D., Han, L., and R. Rokui, "A Yang Data Model for Transport Slice NBI", draft-wd-teas-transport-slice-yang-02 (work in progress), July 2020.

[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>.

[TS23501] "3GPP TS23.501", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

[TS28530] "3GPP TS28.530", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3273>.

[TS28531] "3GPP TS28.531", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3274>.

[TS28541] "3GPP TS 28.541", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3400>.

[ZSM003] "ETSI ZSM003", <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>.

Authors' Addresses

Xuesong Geng
Huawei Technologies

Email: gengxuesong@huawei.com

Jie Dong
Huawei Technologies

Email: jie.dong@huawei.com

Ran Pang
China Unicom

Email: pangran@chinaunicom.cn

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

Tomonobu Niwa
Individual

Email: tomonobu.niwa@gmail.com

Jaehwan Jin
LG U+

Email: daenamu1@lguplus.co.kr

Chang Liu
China Unicom

Email: liuc131@chinaunicom.cn

Nikesh Nageshar
Individual

Email: nikesh.nageshar@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 23, 2021

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
S. Karunanithi
Huawei Technologies
R. Vilalta
CTTC
D. King
Lancaster University
D. Ceccarelli
Ericsson

February 19, 2021

YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics

draft-ietf-teas-actn-pm-telemetry-autonomics-05

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

Copyright Notice

Table of Contents

1. Introduction

The YANG [RFC7950] 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].

[Editor’s Note: A suggestion is made to remove the word KPI from the name of the model. Further discussion is needed.]

1.1. Terminology

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

Key Performance Data: This refers to a set of data the customer is interested in monitoring for their instantiated VNs or TE-tunnels. Key performance data and key performance indicators are inter-exchangeable in this draft.

Scaling: This refers to the network ability to re-shape its own resources. Scale out refers to improve network performance by increasing the allocated resources, while scale in refers to decrease the allocated resources, typically because the existing resources are unnecessary.

Scaling Intent: To declare scaling conditions, scaling intent is used. Specifically, scaling intent refers to the intent expressed by the client that allows the client to program/configure conditions of their key performance data either for scaling out or scaling in. Various conditions can be set for scaling intent on either VN or TE-tunnel level.

Network Autonomics: This refers to the network automation capability that allows client to initiate scaling intent mechanisms and provides the client with the status of the adjusted network resources based on the client’s scaling intent in an automated fashion.

1.1.1. Requirements Language

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

1.2. Tree diagram

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

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
te	ietf-te	[I-D.ietf-teas-yang-te]
te-types	ietf-te-types	[RFC8776]
te-tel	ietf-te-kpi-telemetry	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
vn-tel	ietf-vn-kpi-telemetry	[RFCXXXX]

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.

Further, the following additional documents are refrenced in the model defined in this document -

- o [RFC7471] - OSPF Traffic Engineering (TE) Metric Extensions.
- o [RFC8570] - IS-IS Traffic Engineering (TE) Metric Extensions.
- o [RFC7823] - Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions.

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.

```
      +----------------------------------------------+
      | Client   +-----------------------------+     |
      |          | Dynamic Service Control APP |     |
      |          +-----------------------------+     |
      +----------------------------------------------+
      1.Traffic|  /|\4.Traffic            | /|\
      Monitor& |   | Monitor              |  | 8.Traffic
      Optimize |   | Result     5.Service |  | modify &
      Policy   |   |              modify& |  | optimize
              \|/  |        optimize Req.\|/ | result
      +----------------------------------------------+
      | Orchestrator                                 |
      |    +-------------------------------+         |
      |    |Dynamic Service Control Agent  |         |
      |    +-------------------------------+         |
      |    +---------------+ +-------------------+   |
      |    | Flow Optimize | | vConnection Agent |   |
      |    +---------------+ +-------------------+   |
      +----------------------------------------------+
      2. Path |   /|\3.Traffic            | /|\
      Monitor |    | Monitor              |  |7.Path
      Request |    | Result      6.Path   |  | modify &
              |    |             modify&  |  | optimize
             \|/   |        optimize Req.\|/ | result
      +----------------------------------------------+
      | Network SDN Controller                       |
      |  +----------------------+ +-----------------+|
      |  | Network Provisioning | |Abstract Topology||
      |  +----------------------+ +-----------------+|
      |  +------------------+ +--------------------+ |
      |  |Network Monitoring| |Physical Topology DB| |
      |  +------------------+ +--------------------+ |
      +----------------------------------------------+
```

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

Some of the key points from [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).

- Customer services have various Service Level Agreement (SLA) requirements, such as service availability, latency, latency jitter, packet loss rate, Bit Error Rate (BER), etc. The transport network can satisfy service availability and BER requirements by providing different protection and restoration mechanisms. However, for other performance parameters, there are no such mechanisms. In order to provide high quality services according to customer SLA, one possible solution is to measure the SLA related performance parameters, and dynamically provision and optimize services based on the performance monitoring results.

- Performance monitoring in a large scale network could generate a huge amount of performance information. Therefore, the appropriate way to deliver the information in the client and network interfaces should be carefully considered.

3. Design of the Data Models

The YANG models developed in this document describe two models:

(i) TE KPI Telemetry Model which provides the TE-Tunnel level of performance monitoring mechanism and scaling intent mechanism that allows scale in/out programming by the customer. (See Section 3.1 & 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.

```
                +------------+          +--------------+
                |  TE-Tunnel |          |    TE KPI    |
                |   Model    |<---------|  Telemetry   |
                +------------+ augments |     Model    |
                                        +--------------+
```

3.2. VN KPI Telemetry Model

This module describes performance telemetry for VN model. The telemetry data is augmented both at the VN Level as well as individual VN member level. This module also allows autonomic traffic engineering scaling intent configuration mechanism on the VN level. Scale in/out criteria might be used for network autonomics in order the controller to react to a certain set of variations in monitored parameters (See Section 4 for illustrations).

Moreover, this module also provides mechanism to define aggregated telemetry parameters as a grouping of underlying VN level telemetry parameters. Grouping operation (such as maximum, mean) could be set at the time of configuration. For example, if maximum grouping operation is used for delay at the VN level, the VN telemetry data is reported as the maximum {delay_vn_member_1, delay_vn_member_2,.. delay_vn_member_N}. Thus, this telemetry abstraction mechanism allows the grouping of a certain common set of telemetry values under a grouping operation. This can be done at the VN-member level to suggest how the E2E telemetry be inferred from the per domain tunnel created and monitored by PNCs. One proposed example is the following:

```
     +-------------------------------------------------------------+
     |                      Client                                 |
     |                                                             |
     +-------------------------------------------------------------+
     1.Client sets the      |   /|\   2. Orchestrator pushes:
     grouping op, and       |    |
     subscribes to the      |    |    VN level telemetry for
     VN level telemetry for |    |    - VN Utilized-bw-percentage
     Delay and              |    |       (Minimum across VN Members)
     Utilized-bw-pecentage  |    |    - VN Delay (Maximum across VN
                           \|/   |     Members)
      +-------------------------------------------------------------+
      | Orchestrator                                                |
      |                                                             |
      +-------------------------------------------------------------+
```

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

```
+----------+          +--------------+
|    VN    | augments |      VN      |
|   Model  |<---------|   Telemetry  |
+----------+          |     Model    |
                      +--------------+
```

4. Autonomic Scaling Intent Mechanism

Scaling intent configuration mechanism allows the client to configure automatic scale-in and scale-out mechanisms on both the TE-tunnel and the VN level. Various conditions can be set for auto- scaling based on the PM telemetry data.

There are a number of parameters involved in the mechanism:

- scale-out-intent or scale-in-intent: whether to scale-out or scale-in.

- performance-type: performance metric type (e.g., one-way-delay, one-way-delay-min, one-way-delay-max, two-way-delay, two-way-delay-min, two-way-delay-max, utilized bandwidth, etc.)

- threshold-value: the threshold value for a certain performance-type that triggers scale-in or scale-out.

- scaling-operation-type: in case where scaling condition can be set with one or more performance types, then scaling-operation-type (AND, OR, MIN, MAX, etc.) is applied to these selected performance types and its threshold values.

- Threshold-time: the duration for which the criteria MUST hold true.

- Cooldown-time: the duration after a scaling action has been triggered, for which there will be no further operation.

The following tree is a part of ietf-te-kpi-telemetry tree whose model is presented in full detail in Sections 6 & 7.

```
   module: ietf-te-kpi-telemetry
     augment /te:te/te:tunnels/te:tunnel:
       +--rw te-scaling-intent
       |  +--rw scale-in-intent
       |  |  +--rw threshold-time?      uint32
       |  |  +--rw cooldown-time?       uint32
       |  |  +--rw scaling-condition* [performance-type]
       |  |  |  +--rw performance-type           identityref
       |  |  |  +--rw threshold-value?           string
       |  |  |  +--rw scale-in-operation-type?
       |  |  |          scaling-criteria-operation
       |  |  +--rw scale-in-op?         identityref
       |  |  +--rw scale?               string
       |  +--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
       |     +--rw scale-out-op?        identityref
       |     +--rw scale?               string
```

Let say the client wants to set the scaling out operation based on two performance-types (e.g., two-way-delay and utilized-bandwidth for a te-tunnel), it can be done as follows:

- Set Threshold-time: x (sec) (duration for which the criteria must hold true)

- Set Cooldown-time: y (sec) (the duration after a scaling action has been triggered, for which there will be no further operation)

- Set AND for the scale-out-operation-type

In the scaling condition's list, the following two components can be set:

List 1: Scaling Condition for Two-way-delay

- performance type: Two-way-delay

- threshold-value: z milli-seconds

List 2: Scaling Condition for Utilized bandwidth

- performance type: Utilized bandwidth

- threshold-value: w megabytes

5. Notification

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

- Subscribe to notifications on a per client basis.

- Specify subtree filters or xpath filters so that only interested contents will be sent.

- Specify either periodic or on-demand notifications.

5.1. YANG Push Subscription Examples

[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 (Tunnel1). The telemetry parameter that the client is interested in is one-way-delay.

```
   <netconf:rpc netconf:message-id="101"
       xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
       <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
          <filter netconf:type="subtree">
             <te xmlns="urn:ietf:params:xml:ns:yang:ietf-te">
                <tunnels>
                   <tunnel>
                     <name>Tunnel1</name>
                     <identifier/>
                     <state>
                       <te-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                            ietf-te-kpi-telemetry">
                           <one-way-delay/>
                        </te-telemetry>
                     </state>
                    </tunnel>
                 </tunnels>
             </te>
          </filter>
          <period>500</period>
          <encoding>encode-xml</encoding>
       </establish-subscription>
    </netconf:rpc>
```

This example shows the way for a client to subscribe to the telemetry information for all VNs. The telemetry parameter that the client is interested in is one-way-delay and one-way-utilized- bandwidth.

```
   <netconf:rpc netconf:message-id="101"
       xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
       <establish-subscription
          xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push:1.0">
          <filter netconf:type="subtree">
             <vn-state xmlns="urn:ietf:params:xml:ns:yang:ietf-vn">
                <vn>
                   <vn-list>
                     <vn-id/>
                     <vn-name/>
                     <vn-telemetry xmlns="urn:ietf:params:xml:ns:yang:
                                           ietf-vn-kpi-telemetry">
                         <one-way-delay/>
                         <one-way-utilized-bandwidth/>
                     </vn-telemetry >
                   </vn-list>
                 </vn>
             </vn-state>
          </filter>
          <period>500</period>
       </establish-subscription>
    </netconf:rpc>
```

6. YANG Data Tree

```
   module: ietf-te-kpi-telemetry
     augment /te:te/te:tunnels/te:tunnel:
       +--rw te-scaling-intent
       |  +--rw scale-in-intent
       |  |  +--rw threshold-time?      uint32
       |  |  +--rw cooldown-time?       uint32
       |  |  +--rw scaling-condition* [performance-type]
       |  |  |  +--rw performance-type           identityref
       |  |  |  +--rw threshold-value?           string
       |  |  |  +--rw scale-in-operation-type?
       |  |  |          scaling-criteria-operation
       |  |  +--rw scale-in-op?         identityref
       |  |  +--rw scale?               string
       |  +--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
       |     +--rw scale-out-op?        identityref
       |
       │     +--rw scale?               string
       +--ro te-telemetry
          +--ro id?                            telemetry-id
          +--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:
       +--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-in-op?         identityref
       │  │  +--rw scale?               string
       │  +--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
       |     +--rw scale-out-op?        identityref
       |     +--rw scale?               string
       +--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/vn:vn-member:
       +--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@2021-02-19.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 TE */

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

   /* Import TE Common types */

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

   organization
     "IETF Traffic Engineering Architecture and Signaling (TEAS)
      Working Group";
   contact
     "WG Web:  <https://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 describes YANG data model for performance
      monitoring telemetry for te tunnels.
      Copyright (c) 2021 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.
      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject to
      the license terms contained in, the Simplified BSD License set
      forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

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

      The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
      NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
      'MAY', and 'OPTIONAL' in this document are to be interpreted as
      described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   /* 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 2021-02-19 {
     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.

        At the VN level, it is the max delay of the VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: 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.

        At the VN level, it is the max delay of the VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: 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.

        At the VN level, it is the max delay variation of the
        VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: 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.

        At the VN level, it is the max delay variation of the
        VN-members.";
     reference
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: 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
       "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions.
        RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions.
        RFC 7823: 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.)";
   }

   identity scale-op {
     description
       "Base identity for scaling operation";
   }

   identity scale-capacity-up {
     base scale-op;
     description
       "Scale up the bandwidth capacity";
   }

   identity scale-capacity-down {
     base scale-op;
     description
       "Scale down the bandwidth capacity";
   }

   /* Typedef */

   typedef telemetry-id {
     type string;
     description
       "Identifier for the telemetry data.";
   }

   typedef scaling-criteria-operation {
     type enumeration {
       enum AND {
         description
           "AND operation";
       }
       enum OR {
         description
           "OR operation";
       }
     }
     description
       "Operations to analize list of scaling criterias";
   }

   grouping scaling-duration {
     description
       "Base scaling criteria durations";
     leaf threshold-time {
       type uint32;
       units "seconds";
       description
         "The duration for which the criteria must hold true";
     }
     leaf cooldown-time {
       type uint32;
       units "seconds";
       description
         "The duration after a scaling-in/scaling-out action has been
          triggered, for which there will be no further operation";
     }
   }

   grouping scaling-criteria {
     description
       "Grouping for scaling criteria";
     leaf performance-type {
       type identityref {
         base telemetry-param-type;
       }
       description
         "Reference to the tunnel level telemetry type";
     }
     leaf threshold-value {
       type string;
       description
         "Scaling threshold for the telemetry parameter type";
     }
   }

   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";
       }
     }
     leaf scale-in-op {
       type identityref {
         base scale-op;
       }
       default "scale-capacity-down";
       description
         "The scaling operation to be performed when scaling condition
          is met";
     }
     leaf scale {
       type string;
       description
         "Additional scaling-by information to be interpritted as per
          the scale-in-op.";
     }
   }

   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.
            Defauls to OR";
       }
     }
     leaf scale-out-op {
       type identityref {
         base scale-op;
       }
       default "scale-capacity-up";
       description
         "The scaling operation to be performed when scaling condition
          is met";
     }
     leaf scale {
       type string;
       description
         "Additional scaling-by information to be interpritted as per
          the scale-out-op.";
     }
   }

   augment "/te:te/te:tunnels/te:tunnel" {
     description
       "Augmentation parameters for config scaling-criteria TE
        tunnel topologies. Scale in/out criteria might be used
        for network autonomics in order the controller to react
        to a certain set of monitored params.";
     container te-scaling-intent {
       description
         "The 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 Data";
       leaf id {
         type telemetry-id;
         description
           "ID of telemetry data used for easy reference";
       }
       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@2021-02-19.yang"
   module ietf-vn-kpi-telemetry {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-vn-kpi-telemetry";
     prefix vn-kpi;

     /* Import VN */

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

     /* Import TE */

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

     /* Import TE Common types */

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

     /* Import TE KPI */

     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 XXXX with the number
        assigned to this draft.*/

     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 describes YANG data models for performance
        monitoring telemetry for Virtual Network (VN).

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

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

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

        The key words ’MUST’, ’MUST NOT’, ’REQUIRED’, ’SHALL’, ’SHALL
        NOT’, ’SHOULD’, ’SHOULD NOT’, ’RECOMMENDED’, ’NOT RECOMMENDED’,
        ’MAY’, and ’OPTIONAL’ in this document are to be interpreted as
        described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";

     /* 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 2021-02-19 {
       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 analize list of monitored params";
     }

     grouping vn-telemetry-param {
       description
         "augment of te-kpi:telemetry-param for VN specific params";
       leaf-list te-grouped-params {
         type leafref {
           path
             "/te:te/te:tunnels/te:tunnel/te-kpi:te-telemetry/te-kpi:id";
         }
         description
           "Allows the definition of a vn-telemetry param
            as a grouping of underlying TE params";
       }
       leaf grouping-operation {
         type grouping-operation;
         description
           "describes the operation to apply to
            te-grouped-params";
       }
     }

     augment "/vn:vn/vn:vn" {
       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/vn:vn-member" {
       description
         "Augmentation parameters for state TE vn member topologies.";
       container vn-member-telemetry {
         config false;
         description
           "VN member telemetry params";
         uses te-types:performance-metrics-attributes;
         uses vn-telemetry-param;
       }
     }
   }

   <CODE ENDS>
```

8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF users to a preconfigured subset of all available NETCONF protocol operations and content. The NETCONF Protocol over Secure Shell (SSH) [RFC6242] describes a method for invoking and running NETCONF within a Secure Shell (SSH) session as an SSH subsystem. The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

A number of configuration data nodes defined in this document are writable/deletable (i.e., "config true"). These data nodes may be considered sensitive or vulnerable in some network environments.

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

- /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-in-intent

- /te:te/te:tunnels/te:tunnel/te-scaling-intent/scale-out-intent

- /vn:vn/vn:vn/vn-scaling-intent/scale-in-intent

- /vn:vn/vn:vn/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
--------------------------------------------------------------------

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

10. Acknowledgements

We thank Rakesh Gandhi, Tarek Saad, Igor Bryskin and Kenichi Ogaki 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-10 (work in progress), November 2020.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", draft-ietf-teas-yang-te-25 (work in progress), July 2020.

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

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

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

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

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/info/rfc7926>.

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

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

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

[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>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

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

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

[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>.

[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>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <https://www.rfc-editor.org/info/rfc8776>.

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.

[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>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/info/rfc8309>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[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>.

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

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

Email: dhruv.ietf@gmail.com

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

Email: 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   Y. Lee, Ed.
Internet-Draft   Samsung Electronics
Intended status: Standards Track   D. Dhody, Ed.
Expires: August 23, 2021   Huawei Technologies
                                                              D. Ceccarelli
                                                                   Ericsson
                                                                 I. Bryskin
                                                                 Individual
                                                                   B. Yoon
                                                                       ETRI
                                                          February 19, 2021

A YANG Data Model for VN Operation
draft-ietf-teas-actn-vn-yang-11

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

Sections 2 and 3 provide the discussion of how the VN YANG model is applicable to the ACTN context where Virtual Network Service (VNS) operation is implemented for the Customer Network Controller (CNC)-Multi-Domain Service Coordinator (MSDC) interface (CMI).

The YANG model on the CMI is also known as customer service model in [RFC8309]. The YANG model discussed in this document is used to operate customer-driven VNs during the VN instantiation, VN computation, and its life-cycle service management and operations.

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

1.1. Terminology

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

1.1.1. Requirements Language

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

1.2. Tree diagram

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

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
vn	ietf-vn	[RFCXXXX]
yang	ietf-yang-types	[RFC6991]
nw	ietf-network	[RFC8345]
nt	ietf-network-topology	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te-topo	ietf-te-topology	[RFC8795]

Table 1: Prefixes and corresponding YANG modules

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

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

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

```
                         +-------+
                         |  CNC  |
                         +-------+
                             |
                             |    VN YANG + TE-topology YANG
                             |
                  +------------------------+
                  |          MDSC          |
                  +------------------------+
```

Figure 1: ACTN CMI

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

2.1. Type 1 VN

As defined in [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:

```
                        +---------------+
               L1 ------|               |------ L4
               L2 ------|     AN 1      |------ L7
               L3 ------|               |------ L8
                        +---------------+
```

Figure 2: Abstract Node (One node topology)

Modeling a VN as one abstract node is the easiest way for customers to express their end-to-end connectivity; however, customers are not limited to express their VN only with one abstract node.

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.

```
              +----------------------------------------------+
              |             S1               S2              |
              |              O---------------O               |
              |     ________/ \______         \              |
              |    /                 \         \             |
              |S3 /                   \ S4      \ S5         |
        L1----|-O----------------------O---------O-----------|------L4
              |   \                     \         \          |
              |    \                     \         \         |
              |     \ S6                  \ S7      \ S8     |
              |      O     ----------------O---------O-------|------L7
              |     / \   /                 \   ____/        |
              |S9  /   \ /S10                \ /             |
       L2-----|---O-----O---------------------O--------------|------L8
              |  /                          S11              |
       L3-----|--                                            |
              |                                              |
              +----------------------------------------------+
```

Figure 3: Type 2 topology

As you see from Figure 3, the Type 1 abstract node is depicted as a Type 1 abstract topology comprising of detailed virtual nodes and virtual links.

As an example, if VN-member 1 (L1-L4) is chosen to configure its own path over Type 2 topology, it can select, say, a path that consists of the ERO {S3,S4,S5} based on the topology and its service requirement. This capability is enacted via TE-topology configuration by the customer.

3. High-Level Control Flows with Examples

3.1. Type 1 VN Illustration

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

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

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

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

```
                   +----------------+
          L1 ------|                |------ L4
          L2 ------|      AN 1      |------ L7
          L3 ------|                |------ L8
                   +----------------+
```

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

```
              +--------+                        +---------+
               |  CNC   |                        |  MDSC  |
               +--------+                        +---------+
                    |                                 |
                    |                                 |
   CNC POST TE-topo |  POST /nw:networks/nw:network/  |
   model(with Conn. |  nw:node/te-node-id/            |
   Matrix on one    |  tet:connectivity-matrices/     |
   Abstract node    |  tet:connectivity-matrix        |
                    |-------------------------------->|
                    |                   HTTP 200      |
                    |<--------------------------------|
                    |                                 |
   CNC POST the     |  POST /VN                       |
   VN identifying   |-------------------------------->| If there is
   AP, VNAP and VN- |                                 | multi-src/dest
   Members and maps |                                 | then MDSC
   to the TE-topo   |                 HTTP 200        | selects a
                    |<--------------------------------| src or dest
                    |                                 | and update
                    |                                 | VN YANG
   CNC GET the      |  GET /VN                        |
   VN YANG status   |-------------------------------->|
                    |                                 |
                    |  HTTP 200 (VN with status:      |
                    |  selected VN-members            |
                    |  in case of multi s-d)          |
                    |<--------------------------------|
                    |                                 |
```

3.2. Type 2 VN Illustration

For some VN members, the customer may want to "configure" explicit routes over the path that connects its two end-points. Let us consider the following example.

VN-Member 1 L1-L4 (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).

```
                                   AN1
              ............................................
              .             S1               S2          .
              .              O---------------O           .
              .     ________/ \______         \          .
              .    /                 \         \         .
              . S3/                   \ S4      \ S5     .
        L1----.-O----------------------O---------O-------.----------L4
              .  \                      \         \      .
              .   \                      \         \     .
              .    \ S6                   \ S7      \ S8 .
              .      O     ----------------O---------O---.----------L5
              .     / \   /                 \   ____/ \__.__________L6
              .S9  /   \ /S10                \ /         .
       L2-----.---O-----O----------------------O----------.----------L7
              .  /                          S11\_________.__________L8
       L3-----.--                                        .
              ............................................
```

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.

```
               +--------+                                +--------+
               |  CNC   |                                |  MDSC  |
               +--------+                                +--------+
                    |                                         |
                    |                                         |
   CNC POST TE-topo |  POST /nw:networks/nw:network/          |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-   |
   Matrix on one    |  matrices/tet:connectivity-matrix       |
   Abstract node and|---------------------------------------->|
   Explicit paths in|                                         |
   the conn. matrix)|                       HTTP 200          |
                    |<----------------------------------------|
                    |                                         |
   CNC POST the     |  POST /VN                               |
   VN identifying   |---------------------------------------->|
   AP, VNAP and VN- |                                         |
   Members and maps |                                         |
   to the TE-topo   |                         HTTP 200        |
                    |<----------------------------------------|
                    |                                         |
                    |                                         |
   CNC GET the      |  GET /VN                                |
   VN YANG status   |---------------------------------------->|
                    |                                         |
                    |  HTTP 200 (VN with status)              |
                    |<----------------------------------------|
                    |                                         |
```

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.

```
               +---------+                       +---------+
               |  CNC    |                       |  MDSC   |
               +---------+                       +---------+
                    |                                 |
                    |                                 |
   CNC POST VN      |                                 |
   Identifying AP,  |                                 |
   VNAP and VN-     |  POST /VN                       | MDSC populates
   Members          |--------------------------------->| a single Abst.
                    |                 HTTP 200        | node topology
                    |<--------------------------------| by itself
                    |                                 |
   CNC GET VN &     |  GET /VN  &                     |
   POST TE-Topo     |  POST /nw:networks/nw:network/  |
   Models (with     |  nw:node/te-node-id/tet:        |
   Conn. Matrix     |  connectivity-matrices/         |
   on the           |  tet:connectivity-matrix        |
   Abstract Node    |--------------------------------->|
   and explicit     |                                 |
   paths in the     |                                 |
   conn. matrix)    |                                 |
                    |                 HTTP 200        |
                    |<--------------------------------|
                    |                                 |
                    |                                 |
   CNC GET the      |  GET /VN                        |
   VN YANG status   |--------------------------------->|
                    |                                 |
                    |  HTTP 200 (VN with status)      |
                    |<--------------------------------|
                    |                                 |
```

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.

```
               +--------+                                +--------+
               |  CNC   |                                |  MDSC  |
               +--------+                                +--------+
                    |                                         |
                    |                                         |
   CNC POST TE-topo |  POST /nw:networks/nw:network/          |
   model(with Conn. |  nw:node/te-node-id/tet:connectivity-   |
   Matrix on one    |  matrices/tet:connectivity-matrix       |
   Abstract node and|---------------------------------------->|
   constraints in   |                                         |
   the conn. matrix)|                       HTTP 200          |
                    |<----------------------------------------|
                    |                                         |
                    |                                         |
   CNC calls RPC to |  RPC /vn-compute                        |
   compute the VN   |---------------------------------------->|
   as per the       |                                         |
   refered TE-Topo  |                                         |
                    |                                         |
                    |           HTTP 200 (Computed VN)        |
                    |<----------------------------------------|
                    |                                         |
```

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

```
               +--------+                                +--------+
               |  CNC   |                                |  MDSC  |
               +--------+                                +--------+
                    |                                         |
                    |                                         |
   CNC calls RPC to |  RPC /vn-compute                        |
   compute the VN   |---------------------------------------->|
   as per the       |                                         |
   constraints and  |                                         |
   VN-Members       |                                         |
                    |           HTTP 200 (Computed VN)        |
                    |<----------------------------------------|
                    |                                         |
```

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

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

- te-types:generic-path-constraints: This is used optionally in the RPC input at the VN and/or VN-member level. The VN-member level overrides the VN-level data. This also overrides any constraints in the referred abstract node in the TE topology.

- te-types:generic-path-optimization: This is used optionally in the RPC input at the VN and/or VN-member level. The VN-member level overrides the VN-level data. This also overrides any optimization in the referred abstract node in the TE topology.

- vn-member: This identifies the VN member in both RPC input and output.

- vn-policy: This is used optionally in the RPC input to apply any VN level policies.

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

```
     rpcs:
       +---x vn-compute
          +---w input
          |  +---w abstract-node?
          |  |       -> /nw:networks/network/node/tet:te-node-id
          |  +---w path-constraints
          |  |     ...
          |  +---w optimizations
          |  |     ...
          |  +---w vn-member-list* [vnm-id]
          |  |  +---w vnm-id                    vnm-id
          |  |  +---w src
          |  |  |  +---w src?            -> /ap/ap/ap-id
          |  |  |  +---w src-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
          |  |  |  +---w multi-src?      boolean {multi-src-dest}?
          |  |  +---w dest
          |  |  |  +---w dest?            -> /ap/ap/ap-id
          |  |  |  +---w dest-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
          |  |  |  +---w multi-dest?      boolean {multi-src-dest}?
          |  |  +---w connectivity-matrix-id?   leafref
          |  |  +---w path-constraints
          |  |  |  |     ...
          |  |  +---w optimizations
          |  |           ...
          |  +---w vn-level-diversity?   te-types:te-path-disjointness
          +--ro output
             +--ro abstract-node?
             |       -> /nw:networks/network/node/tet:te-node-id
             +--ro vn-member-list* [vnm-id]
                +--ro vnm-id                    vnm-id
                +--ro src
                |  +--ro src?            -> /ap/ap/ap-id
                |  +--ro src-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
                |  +--ro multi-src?      boolean {multi-src-dest}?
                +--ro dest
                |  +--ro dest?            -> /ap/ap/ap-id
                |  +--ro dest-vn-ap-id?   -> /ap/ap/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?           vn-compute-status
                +--ro error-info
                   +--ro error-description?   string
                   +--ro error-timestamp?     yang:date-and-time
                   +--ro error-reason?        identityref
```

4.3.2. Multi-sources and Multi-destinations

In creating a virtual network, the list of sources or destinations or both may not be pre-determined by the customer. For instance, for a given source, there may be a list of multiple-destinations to which the optimal destination may be chosen depending on the network resource situations. Likewise, for a given destination, there may also be multiple-sources from which the optimal source may be chosen. In some cases, there may be a pool of multiple sources and destinations from which the optimal source-destination may be chosen. The following YANG module is shown for describing source container and destination container. The following YANG tree shows how to model multi-sources and multi-destinations.

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

4.3.3. Others

The VN YANG model can be easily augmented to support the mapping of VN to the Services such as L3SM and L2SM as described in [I-D.ietf-teas-te-service-mapping-yang].

The VN YANG model can be extended to support telemetry, performance monitoring and network 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.

- Maintenance of AP and VNAP along with VN

- VN construct to group of edge-to-edge links

- VN Compute (pre-instantiate)

- Multi-Source / Multi-Destination

- Ability to support various VN and VNS Types

  * VN Type 1: Customer configures the VN as a set of VN Members. No other details need to be set by customer, making for a simplified operations for the customer.

  * VN Type 2: Along with VN Members, the customer could also provide an abstract topology, this topology is provided by the Abstract TE Topology YANG Model.

5. VN YANG Model (Tree Structure)

```
   module: ietf-vn
     +--rw ap
     |  +--rw ap* [ap-id]
     |     +--rw ap-id            ap-id
     |     +--rw pe?
     |     |       -> /nw:networks/network/node/tet:te-node-id
     |     +--rw max-bandwidth?   te-types:te-bandwidth
     |     +--rw avl-bandwidth?   te-types:te-bandwidth
     |     +--rw vn-ap* [vn-ap-id]
     |        +--rw vn-ap-id         ap-id
     |        +--rw vn?              -> /vn/vn/vn-id
     |        +--rw abstract-node?
     |        |       -> /nw:networks/network/node/tet:te-node-id
     |        +--rw ltp?             leafref
     |        +--ro max-bandwidth?   te-types:te-bandwidth
     +--rw vn
        +--rw vn* [vn-id]
           +--rw vn-id                 vn-id
           +--rw vn-topology-id?       te-types:te-topology-id
           +--rw abstract-node?
           |       -> /nw:networks/network/node/tet:te-node-id
           +--rw vn-member* [vnm-id]
           |  +--rw vnm-id                    vnm-id
           |  +--rw src
           |  |  +--rw src?            -> /ap/ap/ap-id
           |  |  +--rw src-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
           |  |  +--rw multi-src?      boolean {multi-src-dest}?
           |  +--rw dest
           |  |  +--rw dest?            -> /ap/ap/ap-id
           |  |  +--rw dest-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
           |  |  +--rw multi-dest?      boolean {multi-src-dest}?
           |  +--rw connectivity-matrix-id?   leafref
           |  +--ro oper-status?              te-types:te-oper-status
           +--ro if-selected?          boolean {multi-src-dest}?
           +--rw admin-status?         te-types:te-admin-status
           +--ro oper-status?          te-types:te-oper-status
           +--rw vn-level-diversity?   te-types:te-path-disjointness

     rpcs:
       +---x vn-compute
          +---w input
          |  +---w abstract-node?
          |  |       -> /nw:networks/network/node/tet:te-node-id
          |  +---w path-constraints
          |  |  +---w te-bandwidth
          |  |  |  +---w (technology)?
          |  |  |        ...
          |  |  +---w link-protection?          identityref
          |  |  +---w setup-priority?           uint8
          |  |  +---w hold-priority?            uint8
          |  |  +---w signaling-type?           identityref
          |  |  +---w path-metric-bounds
          |  |  |  +---w path-metric-bound* [metric-type]
          |  |  |        ...
          |  |  +---w path-affinities-values
          |  |  |  +---w path-affinities-value* [usage]
          |  |  |        ...
          |  |  +---w path-affinity-names
          |  |  |  +---w path-affinity-name* [usage]
          |  |  |        ...
          |  |  +---w path-srlgs-lists
          |  |  |  +---w path-srlgs-list* [usage]
          |  |  |        ...
          |  |  +---w path-srlgs-names
          |  |  |  +---w path-srlgs-name* [usage]
          |  |  |        ...
          |  |  +---w disjointness?             te-path-disjointness
          |  +---w optimizations
          |  |  +---w (algorithm)?
          |  |     +--:(metric) {path-optimization-metric}?
          |  |     |     ...
          |  |     +--:(objective-function)
          |  |              {path-optimization-objective-function}?
          |  |           ...
          |  +---w vn-member-list* [vnm-id]
          |  |  +---w vnm-id                    vnm-id
          |  |  +---w src
          |  |  |  +---w src?            -> /ap/ap/ap-id
          |  |  |  +---w src-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
          |  |  |  +---w multi-src?      boolean {multi-src-dest}?
          |  |  +---w dest
          |  |  |  +---w dest?            -> /ap/ap/ap-id
          |  |  |  +---w dest-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
          |  |  |  +---w multi-dest?      boolean {multi-src-dest}?
          |  |  +---w connectivity-matrix-id?   leafref
          |  |  +---w path-constraints
          |  |  |  +---w te-bandwidth
          |  |  |  |     ...
          |  |  |  +---w link-protection?          identityref
          |  |  |  +---w setup-priority?           uint8
          |  |  |  +---w hold-priority?            uint8
          |  |  |  +---w signaling-type?           identityref
          |  |  |  +---w path-metric-bounds
          |  |  |  |     ...
          |  |  |  +---w path-affinities-values
          |  |  |  |     ...
          |  |  |  +---w path-affinity-names
          |  |  |  |     ...
          |  |  |  +---w path-srlgs-lists
          |  |  |  |     ...
          |  |  |  +---w path-srlgs-names
          |  |  |  |     ...
          |  |  |  +---w disjointness?             te-path-disjointness
          |  |  +---w optimizations
          |  |     +---w (algorithm)?
          |  |           ...
          |  +---w vn-level-diversity?   te-types:te-path-disjointness
          +--ro output
             +--ro abstract-node?
             |       -> /nw:networks/network/node/tet:te-node-id
             +--ro vn-member-list* [vnm-id]
                +--ro vnm-id                    vnm-id
                +--ro src
                |  +--ro src?            -> /ap/ap/ap-id
                |  +--ro src-vn-ap-id?   -> /ap/ap/vn-ap/vn-ap-id
                |  +--ro multi-src?      boolean {multi-src-dest}?
                +--ro dest
                |  +--ro dest?            -> /ap/ap/ap-id
                |  +--ro dest-vn-ap-id?   -> /ap/ap/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?           vn-compute-status
                +--ro error-info
                   +--ro error-description?   string
                   +--ro error-timestamp?     yang:date-and-time
                   +--ro error-reason?        identityref
```

6. VN YANG Model

The YANG model is as follows:

```
   <CODE BEGINS> file "ietf-vn@2021-02-19.yang"
   module ietf-vn {
     yang-version 1.1;
     namespace "urn:ietf:params:xml:ns:yang:ietf-vn";
     prefix vn;

     /* Import network */

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

     /* Import network topology */

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

     /* Import TE Common types */

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

     /* Import TE Topology */

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

     organization
       "IETF Traffic Engineering Architecture and Signaling (TEAS)
        Working Group";
     contact
       "WG Web:  <https://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
        as per RFC 8453.

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

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

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

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
        NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
        'MAY', and 'OPTIONAL' in this document are to be interpreted as
        described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";

     revision 2021-02-19 {
       description
         "initial version.";
       reference
         "RFC XXXX: A YANG Data Model for VN Operation";
     }

     /* Features */

     feature multi-src-dest {
       description
         "Support for selection of one src or destination
          among multiple.";
       reference
         "RFC 8453: Framework for Abstraction and Control of TE
          Networks (ACTN)";
     }

     /* Typedef */

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

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

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

     typedef vn-compute-status {
       type te-types:te-common-status;
       description
         "Defines a type representing the VN compute status";
     }

     /* identities */

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

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

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

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

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

     /* Groupings */

     grouping vn-ap {
       description
         "VNAP related information";
       leaf vn-ap-id {
         type ap-id;
         description
           "A unique identifier for the referred VNAP";
       }
       leaf vn {
         type leafref {
           path "/vn/vn/vn-id";
         }
```

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

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

```
       }
       leaf max-bandwidth {
         type te-types:te-bandwidth;
         description
           "The max bandwidth of the AP";
       }
       leaf avl-bandwidth {
         type te-types:te-bandwidth;
         description
           "The available bandwidth of the AP";
       }
       /*add details and any other properties of AP,
       not associated by a VN
       CE port, PE port etc.
        */
       list vn-ap {
         key "vn-ap-id";
         uses vn-ap;
         description
           "List of VNAP in this AP";
       }
       reference
         "RFC 8453: Framework for Abstraction and Control of TE
          Networks (ACTN), Section 6";
     } //access-point

     grouping vn-member {
       description
         "The vn-member is described by this grouping";
       leaf vnm-id {
         type vnm-id;
         description
           "A vn-member identifier";
       }
       container src {
         description
           "The source of VN Member";
         leaf src {
           type leafref {
             path "/ap/ap/ap-id";
           }
           description
             "A reference to source AP";
         }
         leaf src-vn-ap-id {
           type leafref {
             path "/ap/ap/vn-ap/vn-ap-id";
           }
           description
             "A reference to source VNAP";
         }
         leaf multi-src {
           if-feature "multi-src-dest";
           type boolean;
           default "false";
           description
             "Is the source part of multi-source, where
              only one of the source is enabled";
         }
       }
       container dest {
         description
           "the destination of VN Member";
         leaf dest {
           type leafref {
             path "/ap/ap/ap-id";
           }
           description
             "A reference to destination AP";
         }
         leaf dest-vn-ap-id {
           type leafref {
             path "/ap/ap/vn-ap/vn-ap-id";
           }
           description
             "A reference to dest VNAP";
         }
         leaf multi-dest {
           if-feature "multi-src-dest";
           type boolean;
           default "false";
           description
             "Is destination part of multi-destination, where only one
              of the destination is enabled";
         }
       }
       leaf connectivity-matrix-id {
         type leafref {
           path "/nw:networks/nw:network/nw:node/tet:te/"
              + "tet:te-node-attributes/"
              + "tet:connectivity-matrices/"
              + "tet:connectivity-matrix/tet:id";
         }
         description
           "A reference to connectivity-matrix";
         reference
```

```
           "RFC 8795: YANG Data Model for Traffic Engineering (TE)
            Topologies";
       }
       reference
         "RFC 8454: Information Model for Abstraction and Control of TE
          Networks (ACTN)";
     } //vn-member

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

     /* Configuration data nodes */

     container ap {
       description
         "AP configurations";
       list ap {
         key "ap-id";
         description
           "access-point identifier";
         uses access-point {
           description
             "The access-point information";
         }
       }
       reference
         "RFC 8453: Framework for Abstraction and Control of TE
          Networks (ACTN), Section 6";
     }
     container vn {
       description
         "VN configurations";
       list vn {
         key "vn-id";
         description
           "A virtual network is identified by a vn-id";
         leaf vn-id {
           type vn-id;
           description
             "A unique VN identifier";
         }
         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 {
           key "vnm-id";
           description
             "List of vn-members in a VN";
           uses vn-member;
           leaf oper-status {
             type te-types:te-oper-status;
             config false;
             description
               "The vn-member operational state.";
           }
         }
         leaf if-selected {
           if-feature "multi-src-dest";
           type boolean;
           default "false";
           config false;
           description
             "Is the vn-member is selected among the multi-src/dest
              options";
         }
         leaf admin-status {
           type te-types:te-admin-status;
           default "up";
           description
             "VN administrative state.";
         }
         leaf oper-status {
           type te-types:te-oper-status;
           config false;
           description
             "VN operational state.";
         }
         uses vn-policy;
       } //vn
       reference
         "RFC 8453: Framework for Abstraction and Control of TE
          Networks (ACTN)";
     } //vn

     /* RPC */

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

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

          The output includes a reference to the single node
          abstract topology with each VN-member including a
          reference to the connectivity-matrix-id where the
          path properties could be found. Error information is
          also included.";
       input {
         leaf abstract-node {
           type leafref {
             path "/nw:networks/nw:network/nw:node/tet:te-node-id";
           }
           description
             "A reference to the abstract node in TE Topology";
         }
         uses te-types:generic-path-constraints;
         uses te-types:generic-path-optimization;
         list vn-member-list {
           key "vnm-id";
           description
             "List of VN-members in a VN";
           uses vn-member;
           uses te-types:generic-path-constraints;
           uses te-types:generic-path-optimization;
         }
         uses vn-policy;
       }
       output {
         leaf abstract-node {
           type leafref {
             path "/nw:networks/nw:network/nw:node/tet:te-node-id";
           }
           description
             "A reference to the abstract node in TE Topology";
         }
         list vn-member-list {
           key "vnm-id";
           description
             "List of VN-members in a VN";
           uses vn-member;
           leaf if-selected {
             if-feature "multi-src-dest";
             type boolean;
             default "false";
             description
               "Is the vn-member is selected among the multi-src/dest
                options";
             reference
               "RFC 8453: Framework for Abstraction and Control of TE
                Networks (ACTN), Section 7";
           }
           leaf compute-status {
             type vn-compute-status;
             description
               "The VN-member compute state.";
           }
           container error-info {
             description
               "Error information related to the VN member";
             leaf error-description {
               type string;
               description
                 "Textual representation of the error occurred during
                  VN compute.";
             }
             leaf error-timestamp {
               type yang:date-and-time;
               description
                 "Timestamp of the attempt.";
             }
             leaf error-reason {
               type identityref {
                 base vn-computation-error-reason;
               }
               description
                 "Reason for the VN computation error.";
             }
           }
         }
       }
     } //vn-compute

   }

   <CODE ENDS>
```

7. JSON Example

This section provides json implementation examples as to how VN YANG model and TE topology model are used together to instantiate virtual networks.

The example in this section includes following VN

- VN1 (Type 1): Which maps to the single node topology abstract1 (node D1) and consist of VN Members 104 (L1 to L4), 107 (L1 to L7), 204 (L2 to L4), 308 (L3 to L8) and 108 (L1 to L8). We also show how disjointness (node, link, srlg) is supported in the example on the global level (i.e., connectivity matrices level).
- VN2 (Type 2): Which maps to the single node topology abstract2 (node D2), this topology has an underlay topology (absolute) (see figure in section 3.2). This VN has a single VN member 105 (L1 to L5) and an underlay path (S4 and S7) has been set in the connectivity matrix of abstract2 topology;
- VN3 (Type 1): This VN has a multi-source, multi-destination feature enable for VN Member 104 (L1 to L4)/107 (L1 to L7) {multi-src} and VN Member 204 (L2 to L4)/304 (L3 to L4) {multi-dest} usecase. The selected VN-member is known via the field "if-selected" and the corresponding connectivity-matrix-id.

Note that the VN YANG model also include the AP and VNAP which shows various VN using the same AP.

7.1. VN JSON

```
     {
           "ap":{
              "ap": [
                   {
                    "ap-id": "101",
                    "vn-ap": [
                       {
                          "vn-ap-id": "10101",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "1-0-1"
                       },
                       {
                          "vn-ap-id": "10102",
                          "vn": "2",
                          "abstract-node": "D2",
                          "ltp": "1-0-1"
                       },
                       {
                          "vn-ap-id": "10103",
                          "vn": "3",
                          "abstract-node": "D3",
                          "ltp": "1-0-1"
                       },
                    ]
                 },
                 {
                    "ap-id": "202",
                    "vn-ap": [
                       {
                          "vn-ap-id": "20201",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "2-0-2"
                       }
                    ]
                 },
                 {
                    "ap-id": "303",
                    "vn-ap": [
                       {
                          "vn-ap-id": "30301",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "3-0-3"
                       },
                       {
                          "vn-ap-id": "30303",
                          "vn": "3",
                          "abstract-node": "D3",
                          "ltp": "3-0-3"
                       }
                    ]
                 },
                 {
                    "ap-id": "440",
                    "vn-ap": [
                       {
                          "vn-ap-id": "44001",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "4-4-0"
                       }
                    ]
                 },
                 {
                    "ap-id": "550",
                    "vn-ap": [
                       {
                          "vn-ap-id": "55002",
                          "vn": "2",
                          "abstract-node": "D2",
                          "ltp": "5-5-0"
                       }
                    ]
                 },
                 {
                    "ap-id": "770",
                    "vn-ap": [
                       {
                          "vn-ap-id": "77001",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "7-7-0"
                       },
                       {
                          "vn-ap-id": "77003",
                          "vn": "3",
                          "abstract-node": "D3",
                          "ltp": "7-7-0"
                       }
                    ]
                 },
                 {
                    "ap-id": "880",
                    "vn-ap": [
                       {
                          "vn-ap-id": "88001",
                          "vn": "1",
                          "abstract-node": "D1",
                          "ltp": "8-8-0"
                       },
                       {
                          "vn-ap-id": "88003",
                          "vn": "3",
                          "abstract-node": "D3",
                          "ltp": "8-8-0"
                       }
                    ]
                 }
              ]
           },
           "vn":{
              "vn": [
                 {
                    "vn-id": "1",
                    "vn-topology-id": "te-topology:abstract1",
                    "abstract-node": "D1",
                    "vn-member": [
                       {
                          "vnm-id": "104",
                          "src": {
                             "src": "101",
                             "src-vn-ap-id": "10101",
                          },
                          "dest": {
                             "dest": "440",
                             "dest-vn-ap-id": "44001",
                          },
                          "connectivity-matrix-id": 104
                       },
                       {
                          "vnm-id": "107",
                          "src": {
                             "src": "101",
                             "src-vn-ap-id": "10101",
                          },
                          "dest": {
                             "dest": "770",
                             "dest-vn-ap-id": "77001",
                          },
                          "connectivity-matrix-id": 107
                       },
                       {
                          "vnm-id": "204",
                          "src": {
                             "src": "202",
                             "dest-vn-ap-id": "20401",
                          },
                          "dest": {
                             "dest": "440",
                             "dest-vn-ap-id": "44001",
                          },
                          "connectivity-matrix-id": 204
                       },
                       {
                          "vnm-id": "308",
                          "src": {
                             "src": "303",
                             "src-vn-ap-id": "30301",
                          },
                          "dest": {
                             "dest": "880",
                             "src-vn-ap-id": "88001",
                          },
                          "connectivity-matrix-id": 308
                       },
                       {
                          "vnm-id": "108",
                          "src": {
                             "src": "101",
                             "src-vn-ap-id": "10101",
                          },
                          "dest": {
                             "dest": "880",
                             "dest-vn-ap-id": "88001",
                          },
                          "connectivity-matrix-id": "108"
                       }
                    ]
                 },
                 {
                    "vn-id": "2",
                    "vn-topology-id": "te-topology:abstract2",
                    "abstract-node": "D2",
                    "vn-member": [
                       {
                          "vnm-id": "105",
                          "src": {
                             "src": "101",
                             "src-vn-ap-id": "10102",
                          },
                          "dest": {
                             "dest": "550",
                             "dest-vn-ap-id": "55002",
                          },
                          "connectivity-matrix-id": 105
                       }
                    ]
                 },
                 {
                    "vn-id": "3",
                    "vn-topology-id": "te-topology:abstract3",
                    "abstract-node": "D3",
                    "vn-member": [
                       {
                          "vnm-id": "104",
                          "src": {
                             "src": "101",
                          },
                          "dest": {
                             "dest": "440",
                             "multi-dest": true
                          }
                       },
                       {
                          "vnm-id": "107",
                          "src": {
                             "src": "101",
                             "src-vn-ap-id": "10103",
                          },
                          "dest": {
                             "dest": "770",
                             "dest-vn-ap-id": "77003",
                             "multi-dest": true
                          },
                          "connectivity-matrix-id": 107,
                          "if-selected":true,
                       },
                       {
                          "vnm-id": "204",
                          "src": {
                             "src": "202",
                             "multi-src": true,
                          },
                          "dest": {
                             "dest": "440",
                          },
                       },
                       {
                          "vnm-id": "304",
                          "src": {
                             "src": "303",
                             "src-vn-ap-id": "30303",
                             "multi-src": true,
                          },
                          "dest": {
                             "dest": "440",
                             "src-vn-ap-id": "44003",
                          },
                          "connectivity-matrix-id": 304,
                          "if-selected":true,
                       },
                    ]
                 },

              ]
           }

        }
     }
```

7.2. TE-topology JSON

```
     {
          "networks": {
            "network": [
              {
                "network-types": {
                  "te-topology": {}
                },
                "network-id": "abstract1",
                "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:

- ap:
  - ap-id
  - max-bandwidth
  - avl-bandwidth
- vn-ap:
  - vn-ap-id
  - vn
  - abstract-node
  - ltp
- vn
  - vn-id
  - vn-topology-id
  - abstract-node
- vnm-id
  - src
  - src-vn-ap-id
  - dest

- dest-vn-ap-id

- connectivity-matrix-id

9. IANA Considerations

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

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

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

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

10. Acknowledgments

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

Thanks to Andy Bierman for YANGDIR review.

11. References

11.1. Normative References

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", draft-ietf-teas-yang-te-25 (work in progress), July 2020.

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

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

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

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

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

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

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

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

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

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

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

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

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <https://www.rfc-editor.org/info/rfc8776>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

11.2. Informative References

[I-D.ietf-ccamp-l1csm-yang] Lee, Y., Lee, K., Zheng, H., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang-13 (work in progress), November 2020.

[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-04 (work in progress), November 2020.

[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-05 (work in progress), November 2020.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/info/rfc7926>.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/info/rfc8299>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/info/rfc8309>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[RFC8454] Lee, Y., Belotti, S., Dhody, D., Ceccarelli, D., and B. Yoon, "Information Model for Abstraction and Control of TE Networks (ACTN)", RFC 8454, DOI 10.17487/RFC8454, September 2018, <https://www.rfc-editor.org/info/rfc8454>.

[RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <https://www.rfc-editor.org/info/rfc8466>.

Appendix A. Performance Constraints

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

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

Appendix B. Contributors Addresses

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

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

Haomian Zheng
Huawei Technologies
Email: zhenghaomian@huawei.com

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

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

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

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

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

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

Email: dhruv.ietf@gmail.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Igor Bryskin
Individual

Email: i_bryskin@yahoo.com

Bin Yeong Yoon
ETRI

Email: byyun@etri.re.kr

TEAS Working Group
Internet Draft
Category: Informational
Expires: August 25, 2021

Haomian Zheng
Xianlong Luo
Yi Lin
Huawei Technologies
Yang Zhao
China Mobile
Yunbin Xu
CAICT
Sergio Belotti
Dieter Beller
Nokia
February 21, 2021

Interworking of GMPLS Control and Centralized Controller System

draft-ietf-teas-gmpls-controller-inter-work-05

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

Copyright Notice

Conventions used in this document

Table of Contents

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).

```
                          +-------------------+
                          |   Orchestrator    |
                          +-------------------+
                            ^       ^       ^
                            |       |       |
              +-------------+       |       +-------------+
              |                     |RESTConf/YANG models |
              V                     V                     V
         +----------+          +----------+          +----------+
         |Controller|          |Controller|          |Controller|
         +----------+          +----------+          +----------+
             ^  ^                  ^  ^                   ^
             |  |                  |  |                   |
      Netconf|  |PCEP       Netconf|  |PCEP               | IF*
       /YANG |  |            /YANG |  |                   |
             V  V                  V  V                   V
         .----------.  Inter-  .----------.  Inter-  .----------.
        /            \ domain /            \ domain /            \
       |              | link |     LMP      | link |     LMP      |
       |              |======|   OSPF-TE    |======|   OSPF-TE    |
       |              |      |   RSVP-TE    |      |   RSVP-TE    |
        \            /        \            /        \            /
         `__________'          `__________'          `__________'
      Non-GMPLS domain 1      GMPLS domain 2        GMPLS domain 3
```

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. [RFC8795] 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 [RFC8795]. 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 hierachy 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.

```
+-----------------+   +----------------+   +-----------------+
|Client           |   |                |   |           Client|
|Signal   Domain 1|   |    Domain 2    |   |Domain 3   Signal|
|  |              |   |                |   |              |  |
|+-+-+            |   |                |   |            +-+-+|
|| | |  +--+  +--+|   |+--+  +--+  +--+|   |+--+  +--+  | | ||
|| | |  |  |  |  ||   ||  |  |  |  |  ||   ||  |  |  |  | | ||
|| ******************************************************** ||
||   |  |  |  |  ||   ||  |  |  |  |  ||   ||  |  |  |  |   ||
|+---+  +--+  +--+|   |+--+  +--+  +--+|   |+--+  +--+  +---+|
+-----------------+   +----------------+   +-----------------+
 |   .           .     .              .     .           .   |
 |   .<-S-LSP 1->.     .<- S-LSP 2 -->.     .<-S-LSP 3->.   |
 |               .     .              .     .               |
 |-------------->.---->.------------->.---->.-------------->|
 |<--------------.<----.<-------------.<----.<--------------|
 |       End-to-end RSVP-TE session for LSP stitching       |
```

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.

```
     +-----------------+   +----------------+   +-----------------+
     |Client           |   |                |   |           Client|
     |Signal   Domain 1|   |    Domain 2    |   |Domain 3   Signal|
     |(GE)             |   |                |   |            (GE) |
     |  |   ODU0 tunnel|   |                |   |              |  |
     |+-+-+       ^    |   |                |   |            +-+-+|
     || | |  +--+ |+--+|   |+--+  +--+  +--+|   |+--+  +--+  | | ||
     || | |  |  | ||  ||   ||  |  |  |  |  ||   ||  |  |  |  | | ||
     || ******************************************************** ||
     ||   |  |  |  |  || . ||  |  |  |  |  || . ||  |  |  |  |   ||
     |+---+  +--+  +--+| . |+--+  +--+  +--+| . |+--+  +--+  +---+|
     +-----------------+ . +----------------+ . +-----------------+
      .                  .                    .                  .
      .<-Path Segment 1->.<--Path Segment 2-->.<-Path Segment 3->.
      .                  .                    .                  .
```

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

GMPLS can interwork with centralized controller system in multi-layer networks.

```
      +--------------+
      | Multi-layer  |
      |SDN Controller|
      +-----+--+-----+
            |  |                            Higher-layer Network
            |  |                           .--------------------.
            |  |   +--------------+       /                      \
            |  |   | Higher-layer |      |   +--+   Link   +--+   |
            |  +-->|SDN Controller+----->|   |  |**********|  |   |
            |      +--------------+      |   +--+          +--+   |
            |                             \    .            .    /
            |                               `--.------------.---`
            |                                  .            .
            |                              .---.------------.---.
            |      +--------------+       /    .            .    \
            |      | Lower-layer  |      |   +--+   +--+   +--+   |
            +----->|SDN Controller+----->|   | ============== |   |
                   +--------------+      |   +--+   +--+   +--+   |
                                          \         H-LSP        /
                                            `-------------------`
                                             Lower-layer Network
```

Figure 4: Example of GMPLS-SDN interworking in multi-layer network

An example with two layers of network is shown in Figure 4. In this example, the GMPLS control plane is enabled in each layer network, and interworks with the SDN controller of its domain (higher-layer SDN controller and lower-layer SDN controller, respectively). The multi-layer SDN controller, which acts as the Orchestrator, is used to coordinate the control of the multi-layer network.

7.3.1. Multi-layer Path Computation

[RFC5623] describes three inter-layer path computation models and four inter-layer path control models:

- 3 Path computation:

  - Single PCE path computation model

  - Multiple PCE path computation with inter-PCE communication model

  - Multiple PCE path computation without inter-PCE communication model

- 4 Path control:

  - PCE-VNTM cooperation model

  - Higher-layer signaling trigger model

  - NMS-VNTM cooperation model (integrated flavor)

  - NMS-VNTM cooperation model (separate flavor)

Section 4.2.4 of [RFC5623] also provides all the possible combinations of inter-layer path computation and inter-layer path control models.

To apply [RFC5623] in multi-layer network with GMPLS-SDN interworking, the higher-layer SDN controller and the lower-layer SDN controller can act as the PCE Hi and PCE Lo respectively, and typically, the multi-layer SDN controller can act as a VNTM because it has the abstracted view of both the higher-layer and lower-layer networks.

Table 1 shows all possible combinations of path computation and path control models in multi-layer network with GMPLS-SDN interworking:

Table 1: Combinations of path computation and path control models

Path computation \ Path control	Single PCE (Not applicable)	Multiple PCE with inter-PCE	Multiple PCE w/o inter-PCE
PCE-VNTM cooperation	--	Yes	Yes
Higher-layer signaling trigger	--	Yes	Yes
NMS-VNTM cooperation (integrated flavor)	--	Yes	No
NMS-VNTM cooperation (separate flavor)	--	No	Yes

Not applicable because there are multiple PCEs

Typical models to be used

Note that:

- Since there is one PCE in each layer network, the path computation model "Single PCE path computation" is not applicable.

- For the other two path computation models "Multiple PCE with inter-PCE" and "Multiple PCE w/o inter-PCE", the possible combinations are the same as defined in [RFC5623]. More specifically:

  o The path control models "NMS-VNTM cooperation (integrated flavor)" and "NMS-VNTM cooperation (separate flavor)" are the typical models to be used in multi-layer network with GMPLS-SDN interworking. This is because in these two models, the path computation is triggered by the NMS or VNTM. And in SDN centralized control system, the path computation requests are typically from the multi-layer SDN controller (acts as VNTM).

  o For the other two path control models "PCE-VNTM cooperation" and "Higher-layer signaling trigger", the path computation is triggered by the NEs, i.e., NE performs PCC functions. These two models are still possible to be used, although they are not the main methods.

7.3.2. Cross-layer Path Creation

In a multi-layer network, a lower-layer LSP in the lower-layer network can be created, which will construct a new link in the higher-layer network. Such lower-layer LSP is called Hierarchical LSP, or H-LSP for short, see [RFC6107].

The new link constructed by the H-LSP then can be used by the higher-layer network to create new LSPs.

As described in [RFC5212], two methods are introduced to create the H-LSP: the static (pre-provisioned) method and the dynamic (triggered) method.

1) Static (pre-provisioned) method

In this method, the H-LSP in the lower layer network is created in advance. After that, the higher layer network can create LSPs using the resource of the link constructed by the H-LSP.

The multi-layer SDN controller is responsible to decide the creation of H-LSP in the lower layer network if it acts as a VNTM. It then requests the lower-layer SDN controller to create the H-LSP via, for example, MPI interface under the ACTN architecture. See Section 3.3.2 of [TE-Tunnel].

The lower-layer SDN controller can trigger the GMPLS control plane to create the H-LSP. As a typical example, the PCInitiate message can be used for the communication between the lower-layer SDN controller and the source node of the H-LSP.

And the source node of the H-LSP can trigger the RSVP-TE signaling procedure to create the H-LSP, as described in [RFC6107].

2) Dynamic (triggered) method

In this method, the signaling of LSP creation in the higher layer network will trigger the creation of H-LSP in the lower layer network dynamically, if it is necessary.

In this case, after the cross-layer path is computed, the multi-layer SDN controller requests the higher-layer SDN controller for the cross-layer LSP creation. As a typical example, the MPI interface under the ACTN architecture could be used.

The higher-layer SDN controller can trigger the GMPLS control plane to create the LSP in the higher-layer network. As a typical example, the PCInitiate message can be used for the communication between the higher-layer SDN controller and the source node of the Higher-layer LSP, as described in Section 4.3 of [RFC8282]. At least two sets of

ERO information should be included to indicate the routes of higher-layer LSP and lower-layer H-LSP.

The source node of the Higher-layer LSP follows the procedure defined in Section 4 of [RFC6001], to trigger the GMPLS control plane in both higher-layer network and lower-layer network to create the higher-layer LSP and the lower-layer H-LSP.

On success, the source node of the H-LSP should report the information of the H-LSP to the lower-layer SDN controller via, for example, PCRpt message.

7.3.3. Link Discovery

If the higher-layer network and the lower-layer network are under the same GMPLS control plane instance, the H-LSP can be an FA-LSP. Then the information of the link constructed by this FA-LSP, called FA, can be advertised in the routing instance, so that the higher-layer SDN controller can be aware of this new FA. [RFC4206] and the following updates to it (including [RFC6001] and [RFC6107]) describe the detail extensions to support advertisement of an FA.

If the higher-layer network and the lower-layer network are under separated GMPLS control plane instances, after an H-LSP is created in the lower-layer network, the link discovery procedure defined in LMP protocol ([RFC4204]) will be triggered in the higher-layer network to discover the information of the link constructed by the H-LSP. The information of this new link will be advertised to the higher-layer SDN controller.

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].

7.4.1. Span Protection

Span protection refers to the protection of the link between two neighboring switches. The main protocol requirements include:

- Link management: Link property correlation on the link protection type;

- Routing: announcement of the link protection type;

- Signaling: indication of link protection requirement for that LSP.

GMPLS already supports the above requirements, and there are no new requirements in the scenario of interworking between GMPLS and centralized controller system.

7.4.2. LSP Protection

The LSP protection includes end-to-end and segment LSP protection. For both cases:

- In the provisioning phase:

  The disjoint path computation can be done by the centralized controller system, as it has the global topology and resource view. And the path creation can be done by the procedure described in Section 7.2.

- In the protection switchover phase:

  The existing standards provide the distributed way to trigger the protection switchover. For example, data plane Automatic Protection Switching (APS) mechanism, or GMPLS Notify mechanism described in [RFC4872] and [RFC4873]. In the scenario of interworking between GMPLS and centralized controller system, it is recommended to still use these distributed mechanisms rather than centralized mechanism (i.e., the controller triggers the protection switchover) in the scenario of interworking between GMPLS and centralized controller system. This can significantly shorten the protection switching time.

7.4.3. LSP Restoration

- Pre-planned LSP rerouting (including shared-mesh restoration):

  In pre-planned protecting, the protecting LSP is established only in the control plane in the provisioning phase, and will be activated in the data plane once failure occurs.

  In the scenario of interworking between GMPLS and centralized controller system, the route of protecting LSP can be computed by the centralized controller system. This takes the advantage of making better use of network resource, especially for the resource sharing in shared-mesh restoration.

- Full LSP rerouting:

In full LSP rerouting, the normal traffic will be switched to an alternate LSP that is fully established only after failure occurrence.

As described in [RFC4872] and [RFC4873], the alternate route can be computed on demand when failure occurrence, or pre-computed and stored before failure occurrence.

In a fully distributed scenario, the pre-computation method offers faster restoration time, but has the risk that the pre-computed alternate route may become out of date due to the changes of the network.

In the scenario of interworking between GMPLS and centralized controller system, the pre-computation of the alternate route could be taken place in the centralized controller (and may be stored in the controller or the head-end node of the LSP). In this way, any changes in the network can trigger the refreshment of the alternate route by the centralized controller. This makes sure that the alternate route will not become out of date.

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.

[RFC4206] Kompella, K. and Rekhter Y., "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, 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.

[RFC6001] Papadimitriou D., Vigoureux M., Shiomoto K., Brungard D. and Le Roux JL., "Generalized MPLS (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)", RFC 6001, October 2010.

[RFC6107] Shiomoto K. and Farrel A., "Procedures for Dynamically Signaled Hierarchical Label Switched Paths", RFC 6107, February 2011.

[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.

[RFC8282] Oki E., Takeda T., Farrel A. and Zhang F., "Extensions to the Path Computation Element Communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 8282, December 2017.

[RFC8453] Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", RFC 8453, August 2018.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., Gonzalez De Dios, O., "YANG Data Model for Traffic Engineering (TE) Topologies", RFC8795, August 2020.

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 witching (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.

[RFC5212] Shiomoto K., Papadimitriou D., Le Roux JL., Vigoureux M. and Brungard D., "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, July 2008.

[RFC5623] Oki E., Takeda T., Le Roux JL. and Farrel A., "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.

[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.

[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, 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-ietf-pce-stateful-interdomain, work in progress.

12. Authors' Addresses

Haomian Zheng
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan
Guangdong, 523808 China
Email: zhenghaomian@huawei.com

Xianlong Luo
Huawei Technologies
G1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan
Guangdong, 523808 China
Email: luoxianlong@huawei.com

Yunbin Xu
CAICT
Email: xuyunbin@caict.ac.cn

Yang Zhao
China Mobile
Email: zhaoyangyjy@chinamobile.com

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

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

Yi Lin
Huawei Technologies
H1, Huawei Xiliu Beipo Village, Songshan Lake
Dongguan
Guangdong, 523808 China
Email: yi.lin@huawei.com

teas
Internet-Draft
Intended status: Informational
Expires: August 26, 2021

R. Rokui
Nokia
S. Homma
NTT
K. Makhijani
Futurewei
LM. Contreras
Telefonica
J. Tantsura
Juniper Networks

February 22, 2021

Definition of IETF Network Slices
draft-ietf-teas-ietf-network-slice-definition-01

Abstract

This document provides a definition of the term "IETF Network Slice" for use within the IETF and specifically as a reference for other IETF documents that describe or use aspects of network slices.

The document also describes the characteristics of an IETF network slice, related terms and their meanings, and explains how IETF network slices can be used in combination with end-to-end network slices or independent of them.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A number of use cases benefit from network connections that along with the connectivity provide assurance of meeting a specific set of objectives wrt network resources use. In this document, as detailed in the subsequent sections, we refer to this connectivity and resource commitment as an IETF Network Slice. Services that might benefit from the network slices include but not limited to:

- 5G services (e.g. eMBB, URLLC, mMTC)(See [TS.23.501-3GPP])

- Network wholesale services

- Network infrastructure sharing among operators

- NFV connectivity and Data Center Interconnect

The use cases are further described in [I-D.nsdt-teas-ns-framework].

This document defines the concept of IETF network slices that provide connectivity coupled with a set of specific commitments of network resources between a number of endpoints over a shared network infrastructure. Since the term network slice is rather generic, the qualifying term 'IETF' is used in this document to limit the scope of network slice to network technologies described and standardized by the IETF.

IETF network slices are created and managed within the scope of one or more network technologies (e.g., IP, MPLS, optical). They are intended to enable a diverse set of applications that have different requirements to coexist on the shared network infrastructure. A request for an IETF network slice is technology-agnostic so as to allow a consumer to describe their network connectivity objectives in a common format, independent of the underlying technologies used.

2. Terms and Abbreviations

The terms and abbreviations used in this document are listed below.

- NS: Network Slice

- NSC: Network Slice Controller

- NBI: NorthBound Interface

- SBI: SouthBound Interface

- SLI: Service Level Indicator

- SLO: Service Level Objective

- SLA: Service Level Agreement

The above terminology is defined in greater details in the remainder of this document.

3. Definition and Scope of IETF Network Slice

The definition of a network slice in IETF context is as follows:

An IETF network slice is a logical network topology connecting a number of endpoints using a set of shared or dedicated network resources that are used to satisfy specific Service Level Objectives (SLOs).

An IETF network slice combines the connectivity resource requirements and associated network behaviors such as bandwidth, latency, jitter, and network functions with other resource behaviors such as compute and storage availability. IETF network slices are independent of the underlying infrastructure connectivity and technologies used. This is to allow an IETF network slice consumer to describe their network connectivity and relevant objectives in a common format, independent of the underlying technologies used.

IETF network slices may be combined hierarchically, so that a network slice may itself be sliced. They may also be combined sequentially so that various different networks can each be sliced and the network slices placed into a sequence to provide an end-to-end service. This form of sequential combination is utilized in some services such as in 3GPP's 5G network [TS.23.501-3GPP].

An IETF network slice is technology-agnostic, and the means for IETF network slice realization can be chosen depending on several factors such as: service requirements, specifications or capabilities of underlying infrastructure. The structure and different characteristics of IETF network slices are described in the following sections.

Term "Slice" refers to a set of characteristics and behaviours that separate one type of user-traffic from another. IETF network slice assumes that an underlying network is capable of changing the configurations of the network devices on demand, through in-band signaling or via controller(s) and fulfilling all or some of SLOs to all of the traffic in the slice or to specific flows.

4. IETF Network Slice System Characteristics

The following subsections describe the characteristics of IETF network slices.

4.1. Objectives for IETF Network Slices

An IETF network slice is defined in terms of several quantifiable characteristics or service level objectives (SLOs). SLOs along with terms Service Level Indicator (SLI) and Service Level Agreement (SLA) are used to define the performance of a service at different levels.

A Service Level Indicator (SLI) is a quantifiable measure of an aspect of the performance of a network. For example, it may be a measure of throughput in bits per second, or it may be a measure of latency in milliseconds.

A Service Level Objective (SLO) is a target value or range for the measurements returned by observation of an SLI. For example, an SLO may be expressed as "SLI <= target", or "lower bound <= SLI <= upper bound". A network slice is expressed in terms of the set of SLOs that are to be delivered for the different connections between endpoints.

A Service Level Agreement (SLA) is an explicit or implicit contract between the consumer of an IETF network slice and the provider of the slice. The SLA is expressed in terms of a set of SLOs and may include commercial terms as well as the consequences of missing/ violating the SLOs they contain.

Additional descriptions of IETF network slice attributes is covered in [I-D.contreras-teas-slice-nbi].

4.1.1. Service Level Objectives

SLOs define a set of network attributes and characteristics that describe an IETF network slice. SLOs do not describe 'how' the IETF network slices are implemented or realized in the underlying network layers. Instead, they are defined in terms of dimensions of operation (time, capacity, etc.), availability, and other attributes. An IETF network slice can have one or more SLOs associated with it. The SLOs are combined in an SLA. The SLOs are defined for sets of two or more endpoints and apply to specific directions of traffic flow. That is, they apply to specific source endpoints and specific connections between endpoints within the set of endpoints and connections in the network slice.

4.1.2. Minimal Set of SLOs

This document defines a minimal set of SLOs and later systems or standards could extend this set as per Section 4.1.3.

SLOs can be categorized in to 'Directly Measurable Objectives' or 'Indirectly Measurable Objectives'. Objectives such as guaranteed minimum bandwidth, guaranteed maximum latency, maximum permissible delay variation, maximum permissible packet loss rate, and availability are 'Directly Measurable Objectives'. While 'Indirectly Measurable Objectives' include security, geographical restrictions, maximum occupancy level objectives. The later standard might define other SLOs as needed.

Editor's Note TODO: replace Minimal set to most commonly used objectives to describe network behavior. Other directly or indirectly measurable objectives may be requested by that consumer of an IETF network slice.

The definition of these objectives are as follows:

Guaranteed Minimum Bandwidth

Minimum guaranteed bandwidth between two endpoints at any time. The bandwidth is measured in data rate units of bits per second and is measured unidirectionally.

Guaranteed Maximum Latency

Upper bound of network latency when transmitting between two endpoints. The latency is measured in terms of network characteristics (excluding application-level latency). [RFC2681] and [RFC7679] discuss round trip times and one-way metrics, respectively.

Maximum Permissible Delay Variation

Packet delay variation (PDV) as defined by [RFC3393], s the difference in the one-way delay between sequential packets in a flow. This SLO sets a maximum value PDV for packets between two endpoints.

Maximum permissible packet loss rate

The ratio of packets dropped to packets transmitted between two endpoints over a period of time. See [RFC7680]

Availability

The ratio of uptime to the sum of uptime and downtime, where uptime is the time the IETF network slice is available in accordance with the SLOs associated with it.

Security

An IETF network slice consumer may request that the network applies encryption or other security techniques to traffic flowing between endpoints.

Note that the use of security or the violation of this SLO is not directly observable by the IETF network slice consumer and cannot be measured as a quantifiable metric.

Also note that the objective may include request for encryption (e.g., [RFC4303]) between the two endpoints explicitly to meet architecture recommendations as in [TS33.210] or for compliance with [HIPAA] and/or [PCI].

Editor's Note: Please see more discussion on security in Section 10.

4.1.3. Other Objectives

Additional SLOs may be defined to provide additional description of the IETF network slice that a consumer requests.

If the IETF network slice consumer service is traffic aware, other traffic specific characteristics may be valuable including MTU, traffic-type (e.g., IPv4, IPv6, Ethernet or unstructured), or a higher-level behavior to process traffic according to user-application (which may be realized using network functions).

Maximal occupancy for an IETF network slice should be provided. Since it carries traffic for multiple flows between the two endpoints, the objectives should also say if they are for the entire connection, group of flows or on per flow basis. Maximal occupancy should specify the scale of the flows (i.e. maximum number of flows to be admitted) and optionally a maximum number of countable resource units, e.g IP or MAC addresses a slice might consume.

4.2. IETF Network Slice Endpoints

As noted in Section 3, an IETF network slice describes connectivity between multiple endpoints across the underlying network. These connectivity types are: point-to-point, point-to-multipoint, multipoint-to-point multipoint-to-point, or multipoint-to-multipoint.

Figure 1 shows an IETF network slice along with its NSEs.

The characteristics of IETF network slice endpoints (NSEs) are as follows:

- The IETF network slice endpoints (NSEs) are conceptual points of connection to IETF network slice. As such, they serve as the IETF network slice ingress/egress points.

- Each endpoint could map to a device, application or a network function. A non-exhaustive list of devices, applications or network functions might include but not limited to: routers, switches, firewalls, WAN, 4G/5G RAN nodes, 4G/5G Core nodes, application acceleration, Deep Packet Inspection (DPI), server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and TCP optimizers.

- An NSE should be identified by a unique ID in the context of an IETF network slice consumer.

- In addition to an identifier, each NSE should contain a subset of attributes such as IPv4/IPv6 addresses, encapsulation type (i.e., VLAN tag, MPLS Label etc.), interface/port numbers, node ID etc.

- A combination of NSE unique ID and NSE attributes defines an NSE in the context of the IETF network slice controller.

- During the realization of the IETF network slice, in addition to SLOs, all or subset of IETF NSE attributes will be utilized by IETF network slice controller (NSC) to find the optimal realization in the IETF network.

- Similarly to IETF network slices, the IETF network slice endpoints are logical entities that are mapped to services/tunnels/paths endpoints in IETF network slice during its initialization and realization.

Note that there are various IETF TE terms such as access points (AP) defined in [RFC8453], Termination Point (TP) defined in [RFC8345], and Link Termination Point (LTP) defined in [RFC8795] which are tightly coupled with TE network type and various realization techniques. At the time of realization of the IETF network slice, the NSE could be mapped to one or more of these based on the network slice realization technique in use.

```
                        |----------------------------------|
                   NSE1 |                                  | NSE2
                  O.....|                                  |.....O
                    .   |                                  |  .
                    .   |                                  |  .
                    .   |                                  |  .
                        |                                  |
                   NSEm |                                  | NSEn
                  O.....|                                  |.....O
                        |                                  |
                        |----------------------------------|

                  <------------ IETF Network Slice -------------->
                           between endpoints NSE1 to NSEn

            Legend:
                 NSE: IETF Network Slice Endpoint
                   O: Represents IETF Network Slice Endpoints
```

Figure 1: An IETF Network Slice Endpoints (NSE)

4.2.1. IETF Network Slice Connectivity Types

The IETF Network Slice connection types can be point to point (P2P), point to multipoint (P2MP), multi-point to point (MP2P), or multi-point to multi-point (MP2MP). They will requested by the higher level operation system.

4.3. IETF Network Slice Composition

Operationally, an IETF network slice may be decomposed in two or more IETF network slices as specified below. Decomposed network slices are then independently realized and managed.

- Hierarchical (i.e., recursive) composition: An IETF network slice can be further sliced into other network slices. Recursive composition allows an IETF network slice at one layer to be used by the other layers. This type of multi-layer vertical IETF network slice associates resources at different layers.

- Sequential composition: Different IETF network slices can be placed into a sequence to provide an end-to-end service. In sequential composition, each IETF network slice would potentially support different dataplanes that need to be stitched together.

5. IETF Network Slice Structure

Editor’s note: This content of this section merged with Relationship with E2E slice discussion.

An IETF network slice is a set of connections among various endpoints to form a logical network that meets the SLOs agreed upon.

```
           |------------------------------------------------|
 NSE1 O....|                                                |.....O NSE2
   .       |                                                |     .
   .       |               IETF Network  Slice              |     .
   .       |    (SLOs e.g.  B/W > x bps, Delay < y ms)      |     .
 NSEm O....|                                                |.....O NSEn
           |------------------------------------------------|

== == == == == == == == == == == == == == == == == == == == == == == == ==

                      .--.               .--.
            [EP1]    (    )- .          (    )- .    [EP2]
              .    .’ IETF    ’  SLO  .’  IETF   ’     .
              .   (  Network-1 ) ... (  Network-p )    .
                  `___________’      `___________’
            [EPm]                                    [EPn]

Legendy
  NSE: IETF Network Slice Endpoints
  EP: Serivce/tunnels/path Endpoints used to realize the
      IETF Network Slice
```

Figure 2: IETF Network slice

Figure 2 illustrates a case where an IETF network slice provides connectivity between a set of IEFT network slice endpoints (NSE) pairs with specific SLOs (e.g. guaranteed minimum bandwidth of x bps and guaranteed delay of no more than y ms). The IETF network slice endpoints are mapped to the underlay IETF networks endpoints (EP). Also, the IETF network slice endpoints on the same IETF network slice may belong to the same or different address spaces.

IETF Network slice structure fits into a broader concept of end-to-end network slices. A network operator may be responsible for delivering services over a number of technologies (such as radio networks) and for providing specific and fine-grained services (such as CCTV feed or High definition realtime traffic data). That operator may need to combine slices of various networks to produce an end-to-end network service. Each of these networks may include multiple physical or virtual nodes and may also provide network functions beyond simply carrying of technology-specific protocol data units.An end-to-end network slice is defined by the 3GPP as a complete logical network that provides a service in its entirety with a specific assurance to the consumer [TS.23.501-3GPP].

An end-to-end network slice may be composed from other network slices that include IETF network slices. This composition may include the hierarchical (or recursive) use of underlying network slices and the sequential (or stitched) combination of slices of different networks.

6. IETF Network Slice Stakeholders

An IETF network slice and its realization involves the following stakeholders and it is relevant to define them for consistent terminology.

Consumer: A consumer is the requester of an IETF network slice. Consumers may request monitoring of SLOs. A consumer may manage the IETF network slice service directly by interfacing with the IETF network slice controller or indirectly through an orchestrator.

Orchestrator: An orchestrator is an entity that composes different services, resource and network requirements. It interfaces with the IETF network slice controllers.

IETF Network Slice Controller (NSC): It realizes an IETF network lice in the underlying network, maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed between different types of IETF network slice controllers and different types of orchestrators. An IETF network slice operator (or slice operator for short) manages one or more IETF network slices using the IETF network slice Controller(s).

Network Controller: is a form of network infrastructure controller that offers network resources to NSC to realize a particular network slice. These may be existing network controllers associated with one or more specific technologies that may be adapted to the function of realizing IETF network slices in a network.

7. IETF Network Slice Controller Interfaces

The interworking and interoperability among the different stakeholders to provide common means of provisioning, operating and monitoring the IETF network slices is enabled by the following communication interfaces (see Figure 3).

NSC Northbound Interface (NBI): The NSC Northbound Interface is an interface between a consumer's higher level operation system (e.g., a network slice orchestrator) and the NSC. It is a technology agnostic interface. The consumer can use this interface to communicate the requested characteristics and other requirements (i.e., the SLOs) for the IETF network slice, and the NSC can use the interface to report the operational state of an IETF network slice to the consumer.

NSC Southbound Interface (SBI): The NSC Southbound Interface is an interface between the NSC and network controllers. It is technology-specific and may be built around the many network models defined within the IETF.

```
                   +-------------------------------------------+
                   | Consumer higher level operation system    |
                   |   (e.g E2E network slice orchestrator)    |
                   +-------------------------------------------+
                                        A
                                        | NSC NBI
                                        V
                   +-------------------------------------------+
                   |    IETF Network Slice Controller (NSC)    |
                   +-------------------------------------------+
                                        A
                                        | NSC SBI
                                        V
                   +-------------------------------------------+
                   |           Network Controllers             |
                   +-------------------------------------------+
```

Figure 3: Interface of IETF Network Slice Controller

8. Realizing IETF Network Slice

Realization of IETF network slices is out of scope of this document. It is a mapping of the definition of the IETF network slice to the underlying infrastructure and is necessarily technology-specific and achieved by the NSC over the SBI.

The realization can be achieved in a form of either physical or logical connectivity through VPNs (see, for example, [I-D.ietf-teas-enhanced-vpn], a variety of tunneling technologies such as Segment Routing, MPLS, etc. Accordingly, endpoints may be realized as physical or logical service or network functions.

9. Isolation in IETF Network Slices

An IETF network slice consumer may request, that the IETF Network Slice delivered to them is isolated from any other network slices of services delivered to any other consumers. It is expected that the changes to the other network slices of services do not have any negative impact on the delivery of the IETF network slice.

9.1. Isolation as a Service Requirement

Isolation may be an important requirement of IETF network slices for some critical services. A consumer may express this request as an SLO.

This requirement can be met by simple conformance with other SLOs. For example, traffic congestion (interference from other services) might impact on the latency experienced by an IETF network slice. Thus, in this example, conformance to a latency SLO would be the primary requirement for delivery of the IETF network slice service, and isolation from other services might be only a means to that end.

It should be noted that some aspects of isolation may be measurable by a consumer who have the information about the traffic on a number of IETF network slices or other services.

9.2. Isolation in IETF Network Slice Realization

Delivery of isolation is achieved in the realization of IETF network slices, with existing, in-development, and potential new technologies in IETF. It depends on how a network operator decides to operate their network and deliver services.

Isolation may be achieved in the underlying network by various forms of resource partitioning ranging from dedicated allocation of resources for a specific IETF network slice, to sharing or resources with safeguards. For example, traffic separation between different IETF network slices may be achieved using VPN technologies, such as L3VPN, L2VPN, EVPN, etc. Interference avoidance may be achieved by network capacity planning, allocating dedicated network resources, traffic policing or shaping, prioritizing in using shared network resources, etc. Finally, service continuity may be ensured by reserving backup paths for critical traffic, dedicating specific network resources for a selected number of network slices, etc.

10. Security Considerations

This document specifies terminology and has no direct effect on the security of implementations or deployments. In this section, a few of the security aspects are identified.

- Conformance to security constraints: Specific security requests from consumer defined IETF network slices will be mapped to their realization in the unerlay networks. It will be required by underlay networks to have capabilities to conform to consumer's requests as some aspects of security may be expressed in SLOs.

- IETF network slice controller authentication: Unerlying networks need to be protected against the attacks from an adversary NSC as they can destablize overall network operations. It is particularly critical since an IETF network slice may span across different networks, therefore, IETF NSC should have strong authentication with each those networks. Futhermore, both SBI and NBI need to be secured.

- Specific isolation criteria: The nature of conformance to isolation requests means that it should not be possible to attack an IETF network slice service by varying the traffic on other services or slices carried by the same underlay network. In general, isolation is expected to strengthen the IETF network slice security.

- Data Integrity of an IETF network slice: A consumer wanting to secure their data and keep it private will be responsible for applying appropriate security measures to their traffic and not depending on the network operator that provides the IETF network slice. It is expected that for data integrity, a consumer is responsible for end-to-end encryption of its own traffic.

Note: see NGMN document [NGMN_SEC] on 5G network slice security for discussion relevant to this section.

11. IANA Considerations

This memo includes no request to IANA.

12. Acknowledgment

The entire TEAS NS design team and everyone participating in those discussion has contributed to this draft. Particularly, Eric Gray, Xufeng Liu, Jie Dong, Adrian Farrel, and Jari Arkko for a thorough review among other contributions.

13. Informative References

[HIPAA] HHS, "Health Insurance Portability and Accountability Act - The Security Rule", February 2003, <https://www.hhs.gov/hipaa/for-professionals/security/index.html>.

[I-D.contreras-teas-slice-nbi] Contreras, L., Homma, S., and J. Ordonez-Lucena, "Considerations for defining a Transport Slice NBI", draft-contreras-teas-slice-nbi-01 (work in progress), March 2020.

[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+) Services", draft-ietf-teas-enhanced-vpn-05 (work in progress), February 2020.

[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.nsdt-teas-ns-framework] Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-02 (work in progress), March 2020.

[NGMN_SEC] NGMN Alliance, "NGMN 5G Security - Network Slicing", April 2016, <https://www.ngmn.org/wp-content/uploads/Publications/2016/160429_NGMN_5G_Security_Network_Slicing_v1_0.pdf>.

[PCI] PCI Security Standards Council, "PCI DSS", May 2018, <https://www.pcisecuritystandards.org>.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <https://www.rfc-editor.org/info/rfc2681>.

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

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <https://www.rfc-editor.org/info/rfc3393>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

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

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[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>.

[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>.

[TS.23.501-3GPP]
3rd Generation Partnership Project (3GPP), "3GPP TS 23.501 (V16.2.0): System Architecture for the 5G System (5GS); Stage 2 (Release 16)", September 2019, <http://www.3gpp.org/ftp//Specs/archive/23_series/23.501/23501-g20.zip>.

[TS33.210]
3GPP, "3G security; Network Domain Security (NDS); IP network layer security (Release 14).", December 2016, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2279>.

Authors' Addresses

Reza Rokui
Nokia
Canada

Email: reza.rokui@nokia.com

Shunsuke Homma
NTT
Japan

Email: shunsuke.homma.ietf@gmail.com

Kiran Makhijani
Futurewei
USA

Email: kiranm@futurewei.com

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

Jeff Tantsura
Juniper Networks

Email: jefftant.ietf@gmail.com

TEAS Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2021

Y. Lee, Ed.
Samsung Electronics
D. Dhody, Ed.
G. Fioccola
Q. Wu, Ed.
Huawei Technologies
D. Ceccarelli
Ericsson
J. Tantsura
Apstra

February 21, 2021

Traffic Engineering (TE) and Service Mapping Yang Model
draft-ietf-teas-te-service-mapping-yang-07

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

Copyright Notice

Table of Contents

1. Introduction

Data models are a representation of objects that can be configured or monitored within a system. Within the IETF, YANG [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-l1csm-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.

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

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].

1.1.1. Requirements Language

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

1.2. Tree diagram

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

1.3. Prefixes in Data Node Names

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

Prefix	YANG module	Reference
tsmt	ietf-te-service-mapping-types	[RFCXXXX]
l1csm	ietf-l1csm	[I-D.ietf-ccamp-l1csm-yang]
l2vpn-svc	ietf-l2vpn-svc	[RFC8466]
l3vpn-svc	ietf-l3vpn-svc	[RFC8299]
l1-tsm	ietf-l1csm-te-service-mapping	[RFCXXXX]
l2-tsm	ietf-l2sm-te-service-mapping	[RFCXXXX]
l3-tsm	ietf-l3sm-te-service-mapping	[RFCXXXX]
vn	ietf-vn	[I-D.ietf-teas-actn-vn-yang]
nw	ietf-network	[RFC8345]
te-types	ietf-te-types	[RFC8776]
te	ietf-te	[I-D.ietf-teas-yang-te]
l2vpn-ntw	ietf-l2vpn-ntw	[I-D.ietf-opsawg-l2nm]
l3vpn-ntw	ietf-l3vpn-ntw	[I-D.ietf-opsawg-l3sm-l3nm]
rt	ietf-routing	[RFC8349]
sr-policy	ietf-sr-policy	[I-D.ietf-spring-sr-policy-yang]

Table 1: Prefixes and corresponding YANG modules

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

2. TE and Service Related Parameters

While L1/L2/L3 service models (L1CSM, L2SM, L3SM) are intended to provide service-specific parameters for VPN service instances, there are a number of TE Service related parameters that are not included in these service models.

Additional 'service parameters and policies' that are not included in the aforementioned service models are addressed in the YANG models defined in this document.

2.1. VN/Tunnel Selection Requirements

In some cases, the service requirements may need addition TE tunnels to be established. This may occur when there are no suitable existing TE tunnels that can support the service requirements, or when the operator would like to dynamically create and bind tunnels to the VPN such that they are not shared by other VPNs, for example, for network slicing. The establishment of TE tunnels is subject to the network operator's policies.

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

- New VN/Tunnel Binding - A customer could request a VPN service based on VN/Tunnels that are not shared with other existing or future services. This might be to meet VPN isolation requirements. Further, the YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the ACTN VN. The VN (and TE tunnels) could be bound to the VPN and not used for any other VPN. Under this mode, the following sub-categories can be supported:

  1. Hard Isolation with deterministic characteristics: A customer could request a VPN service using a set of TE Tunnels with deterministic characteristics requirements (e.g., no latency variation) and where that set of TE Tunnels must not be shared with other VPN services and must not compete for bandwidth or other network resources with other TE Tunnels.

  2. Hard Isolation: This is similar to the above case but without the deterministic characteristics requirements.

  3. Soft Isolation: The customer requests a VPN service using a set of TE tunnels which can be shared with other VPN services.

- VN/Tunnel Sharing - A customer could request a VPN service where new tunnels (or a VN) do not need to be created for each VPN and can be shared across multiple VPNs. Further, the mapping YANG model described in Section 5 of this document can be used to describe the mapping between the VPN service and the tunnels in use. No modification of the properties of a tunnel (or VN) is allowed in this mode: an existing tunnel can only be selected.

- VN/Tunnel Modify - This mode allows the modification of the properties of the existing VN/tunnel (e.g., bandwidth).

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

2.2. TE Policy

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

2.2.1. Availability Requirement

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

The availability level will need to be translated into network specific policies such as the protection/reroute policy associated with a VN or Tunnel. The means by which this is achieved is not in the scope of this document.

3. YANG Modeling Approach

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

```
   +--------------+        +----------------------+         +----------+
   |    LxSM      |o-------|                      | . . . . | ACTN VN  |
   +--------------+ augment|                      |         +----------+
                           |                      |         +----------+
   +--------------+        | Augmented LxSM Model | . . . . | TE-topo  |
   | TE & Service |------->|                      |         +----------+
   | Mapping Types| import |                      |         +----------+
   +--------------+        |                      | . . . . | TE-tunnel|
                           +----------------------+         +----------+
                                                   reference
```

Figure 1: Augmented LxSM Model

The Augmented LxSM model (where x=1,2,3) augments the basic LxSM model while importing the common TE 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 behavior of the network. It also allows for the customers to access operational state information about how their services are instantiated with the underlying VN, TE topology or TE tunnels provided that the MDSC operator is willing to share that information. This mapping will facilitate a seamless service management operation with underlay-TE network visibility.

As seen in Figure 1, the augmented LxSM service model records a mapping between the customer service models and the ACTN VN YANG model. Thus, when the MDSC receives a service request it creates a VN that meets the customer's service objectives with various constraints via TE-topology model [RFC8795], and this relationship is recorded by the Augmented LxSM Model. The model also supports a mapping between a service model and TE-topology or a TE-tunnel.

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

3.1. Forward Compatibility

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

It is possible that new service models will be defined at some future time and that it will be desirable to map them to underlying TE constructs in the same way as the three existing models are augmented.

3.2. TE and Network Models

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

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

```
   +---------------+        +----------------------+          +-----------+
   |     LxNM      |o-------|                      | . . . .  | ACTN VN   |
   +---------------+ augment|                      |          +-----------+
                            |                      |          +-----------+
   +---------------+        | Augmented LxNM Model | . . . .  | TE-topo   |
   | TE & Service  |------->|                      |          +-----------+
   | Mapping Types | import |                      |          +-----------+
   +---------------+        |                      | . . . .  | TE-tunnel |
                            +----------------------+          +-----------+
                                                   reference
```

Figure 2: Augmented LxNM Model

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

4. L3VPN Architecture in the ACTN Context

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

```
                            +-----------------------------+
                            |  Customer Service Manager   |
                            |  +-----------------------+  |
                            |  |          CNC          +  |
                            |  +-+-------------------+-+  |
                            +----|-------------------|---+
                                 |                   |
                                 |CMI(Augmented L3SM)|CMI(VN)
                                 |                   |
                +----------------|-------------------|----+
                | +--------------|-----------------+ |    |
                | | MDSC         |                 | |    |
                | |              |                 | |    |
                | |  +-----------+-------------+   | |    |
   TE-Svc-Map<------+ Service Mapping Function |   | |    |
                | |  +-----------+-------------+   | |    |
                | |              |                 | |    |
                | +-------+------|-----------------+ |    |
                |         |      |                   |    |
                |         |      |CMI(VN)            |    |
                |         |      |                   |    |
                |         |   +--|-------------------|--+ |
                |         |   |  |        MDSC       |  | |
                |         |   | ++-------------------++ | |
                |         |   | +   Service Mapping   +---->TE-Svc-Map
                |         |   | ++----------+---------+ | |
                |         |   +--|----------|-----------+ |
                +---------|------|----------|-------------+
                          |      |          |
                          | +----+--------+ |
                          | |             | |
      MPI(VPN / TE models)| |             | |MPI(TE / L1 models)
                          | |             | |
                    +-----|-|---+   +-----|-|----+
         IP/MPLS    |  +--+-+-+ |   |  +--+-+-+  | Optical Domain
         Domain     |  | PNC1 | |   |  | PNC2 |  | Controller
         Controller |  +--+---+ |   |  +--+---+  |
                    +-----|-----+   +-----|------+
                          |               |
                          V               |  SBI
              +---------------------+     |
             /    IP/MPLS Network    \    |
            +-------------------------+   |
                                          V
                               +---------------------+
                              /    Optical Network    \
                             +-------------------------+
```

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

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

- CNC: The Customer Network Controller is responsible for generating service requests. In the context of an L3VPN, the CNC uses the Augmented L3SM to express the service request and communicate it to the network operator.

- MDSC: This entity is responsible for coordinating a L3VPN service request (expressed via the Augmented L3SM) with the IP/MPLS PNC and the Transport PNC. For TE services, one of the key responsibilities of the MDSC is to coordinate with both the IP PNC and the Transport PNC for the mapping of the Augmented L3VPN Service Model to the ACTN VN model. In the VN/TE-tunnel binding case, the MDSC will need to coordinate with the Transport PNC to dynamically create the TE-tunnels in the transport network as needed. These tunnels are added as links in the IP/MPLS Layer topology. The MDSC coordinates with IP/MPLS PNC to create the TE-tunnels in the IP/MPLS layer, as part of the ACTN VN creation.

o PNC: The Provisioning Network Controller is responsible for configuring and operating the network devices. Figure 3 shows two distinct PNCs.

  * IP/MPLS PNC (PNC1): This entity is responsible for device configuration to create PE-PE L3VPN tunnels for the VPN customer and for the configuration of the L3VPN VRF on the PE nodes. Each network element would select a tunnel based on the configuration.

  * Transport PNC (PNC2): This entity is responsible for device configuration for TE tunnels in the transport networks.

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

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 3, the MDSC may be used recursively. For example, the CNC might map a L3SM request to a VN request that it sends to a recursive MDSC.

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

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

2. The MDSC considers the service request and local policy to determine if it needs to create a new VN or any TE Topology, and if that is the case, ACTN VN YANG [I-D.ietf-teas-actn-vn-yang] is used to configure a new VN based on this VPN and map the VPN service to the ACTN VN. In case an existing tunnel is to be used, each device will select which tunnel to use and populate this mapping information.

3. The MDSC interacts with both the IP/MPLS PNC and the Transport PNC to create a PE-PE tunnel in the IP network mapped to a TE tunnel in the transport network by providing the inter-layer access points and tunnel requirements. The specific service information is passed to the IP/MPLS PNC for the actual VPN configuration and activation.

   A. The Transport PNC creates the corresponding TE tunnel matching with the access point and egress point.
   B. The IP/MPLS PNC maps the VPN ID with the corresponding TE tunnel ID to bind these two IDs.

4. The IP/MPLS PNC creates/updates a VRF instance for this VPN customer. This is not in the scope of this document.

4.1. Service Mapping

Augmented L3SM and L2SM can be used to request VPN service creation including the creation of sites and corresponding site network access connection between CE and PE. A VPN-ID is used to identify each VPN service ordered by the customer. The ACTN VN can be used further to establish PE-to-PE connectivity between VPN sites belonging to the same VPN service. A VN-ID is used to identify each virtual network established between VPN sites.

Once the ACTN VN has been established over the TE network (maybe a new VN, maybe modification of an existing VN, or maybe the use of an unmodified existing VN), the mapping between the VPN service and the ACTN VN service can be created.

4.2. Site Mapping

The elements in Augmented L3SM and L2SM define site location parameters and constraints such as distance and access diversity that can influence the placement of network attachment points (i.e, virtual network access points (VNAP)). To achieve this, a central directory can be set up to establish the mapping between location parameters and constraints and network attachment point location. Suppose multiple attachment points are matched, the management system can use constraints or other local policy to select the best candidate network attachment points.

After a network attachment point is selected, the mapping between VPN site and VNAP can be established as shown in Table 1.

Site	Site Network Access	Location (Address, Postal Code, State, City,Country Code)	Access Diversity (Constraint-Type, Group-id,Target Group-id)	PE
SITE1	ACCESS1	(,,US,NewYork,)	(10,PE-Diverse,10)	PE1
SITE2	ACCESS2	(,,CN,Beijing,)	(10,PE-Diverse,10)	PE2
SITE3	ACCESS3	(,,UK,London,)	(12,same-PE,12)	PE4
SITE4	ACCESS4	(,,FR,Paris,)	(20,Bearer-Diverse,20)	PE7

Table 2: : Mapping Between VPN Site and VNAP

5. Applicability of TE-Service Mapping in Generic context

As discussed in the Introduction Section, the models presented in this document are also applicable generically outside of the ACTN architecture. [RFC8309] defines Customer Service Model between Customer and Service Orchestrator and Service Delivery Model between Service Orchestrator and Network Orchestrator(s). TE-Service mapping models defined in this document can be regarded primarily as Customer Service Model and secondarily as Service Deliver Model.

6. YANG Data Trees

6.1. Service Mapping Types

```
module: ietf-te-service-mapping-types
  +--rw te-mapping-templates
     +--rw te-mapping-template* [id]
        +--rw id                    te-mapping-template-id
        +--rw description?          string
        +--rw map-type?             identityref
        +--rw path-constraints
        |  +--rw te-bandwidth
```

```
           |  |  +--rw (technology)?
           |  |     +--:(generic)
           |  |        +--rw generic?   te-bandwidth
           |  +--rw link-protection?          identityref
           |  +--rw setup-priority?           uint8
           |  +--rw hold-priority?            uint8
           |  +--rw signaling-type?           identityref
           |  +--rw path-metric-bounds
           |  |  +--rw path-metric-bound* [metric-type]
           |  |     +--rw metric-type    identityref
           |  |     +--rw upper-bound?   uint64
           |  +--rw path-affinities-values
           |  |  +--rw path-affinities-value* [usage]
           |  |     +--rw usage    identityref
           |  |     +--rw value?   admin-groups
           |  +--rw path-affinity-names
           |  |  +--rw path-affinity-name* [usage]
           |  |     +--rw usage            identityref
           |  |     +--rw affinity-name* [name]
           |  |        +--rw name    string
           |  +--rw path-srlgs-lists
           |  |  +--rw path-srlgs-list* [usage]
           |  |     +--rw usage     identityref
           |  |     +--rw values*   srlg
           |  +--rw path-srlgs-names
           |  |  +--rw path-srlgs-name* [usage]
           |  |     +--rw usage    identityref
           |  |     +--rw names*   string
           |  +--rw disjointness?             te-path-disjointness
           +--rw optimizations
              +--rw (algorithm)?
                 +--:(metric) {path-optimization-metric}?
                 |  +--rw optimization-metric* [metric-type]
                 |  |  +--rw metric-type
                 |  |  |       identityref
                 |  |  +--rw weight?                           uint8
                 |  |  +--rw explicit-route-exclude-objects
                 |  |  |     ...
                 |  |  +--rw explicit-route-include-objects
                 |  |        ...
                 |  +--rw tiebreakers
                 |     +--rw tiebreaker* [tiebreaker-type]
                 |           ...
                 +--:(objective-function)
                          {path-optimization-objective-function}?
                    +--rw objective-function
                       +--rw objective-function-type?   identityref
```

6.2. Service Models

6.2.1. L3SM

```
  module: ietf-l3sm-te-service-mapping
    augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:vpn-services
              /l3vpn-svc:vpn-service:
      +--rw te-service-mapping!
         +--rw te-mapping
            +--rw map-type?                  identityref
            +--rw te-policy
            |  +--rw color?               uint32
            |  +--rw protection-type?     identityref
            |  +--rw availability-type?   identityref
            +--rw (te)?
            |  +--:(vn)
            |  |  +--rw vn*                  -> /vn:vn/vn/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*           te:tunnel-ref
            |     +--rw sr-policy*
            |             [policy-color-ref policy-endpoint-ref]
            |             {sr-policy}?
            |        +--rw policy-color-ref       leafref
            |        +--rw policy-endpoint-ref    leafref
            +--rw te-mapping-template-ref?
                    -> /tsmt:te-mapping-templates/te-mapping-template/id
                    {template}?
    augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
              /l3vpn-svc:site-network-accesses
              /l3vpn-svc:site-network-access:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
    augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
              /l3vpn-svc:service/l3vpn-svc:qos/l3vpn-svc:qos-profile
              /l3vpn-svc:qos-profile/l3vpn-svc:custom/l3vpn-svc:classes
              /l3vpn-svc:class:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
```

```
            +--rw ltp?     te-types:te-tp-id
    augment /l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site
              /l3vpn-svc:site-network-accesses
              /l3vpn-svc:site-network-access/l3vpn-svc:service
              /l3vpn-svc:qos/l3vpn-svc:qos-profile
              /l3vpn-svc:qos-profile/l3vpn-svc:custom/l3vpn-svc:classes
              /l3vpn-svc:class:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
```

6.2.2. L2SM

```
  module: ietf-l2sm-te-service-mapping
    augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:vpn-services
              /l2vpn-svc:vpn-service:
      +--rw te-service-mapping!
         +--rw te-mapping
            +--rw map-type?                 identityref
            +--rw te-policy
            |  +--rw color?              uint32
            |  +--rw protection-type?    identityref
            |  +--rw availability-type?  identityref
            +--rw (te)?
            |  +--:(vn)
            |  |  +--rw vn*                 -> /vn:vn/vn/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*          te:tunnel-ref
            |     +--rw sr-policy*
            |             [policy-color-ref policy-endpoint-ref]
            |             {sr-policy}?
            |        +--rw policy-color-ref       leafref
            |        +--rw policy-endpoint-ref    leafref
            +--rw te-mapping-template-ref?
                    -> /tsmt:te-mapping-templates/te-mapping-template/id
                    {template}?
    augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
              /l2vpn-svc:site-network-accesses
              /l2vpn-svc:site-network-access:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
    augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
              /l2vpn-svc:service/l2vpn-svc:qos/l2vpn-svc:qos-profile
              /l2vpn-svc:qos-profile/l2vpn-svc:custom/l2vpn-svc:classes
              /l2vpn-svc:class:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
    augment /l2vpn-svc:l2vpn-svc/l2vpn-svc:sites/l2vpn-svc:site
              /l2vpn-svc:site-network-accesses
              /l2vpn-svc:site-network-access/l2vpn-svc:service
              /l2vpn-svc:qos/l2vpn-svc:qos-profile
              /l2vpn-svc:qos-profile/l2vpn-svc:custom/l2vpn-svc:classes
              /l2vpn-svc:class:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
```

6.2.3. L1CSM

```
  module: ietf-l1csm-te-service-mapping
    augment /l1csm:l1-connectivity/l1csm:services/l1csm:service:
      +--rw te-service-mapping!
         +--rw te-mapping
            +--rw map-type?                  identityref
            +--rw te-policy
            |  +--rw color?               uint32
            |  +--rw protection-type?     identityref
            |  +--rw availability-type?   identityref
            +--rw (te)?
            |  +--:(vn)
            |  |  +--rw vn*                  -> /vn:vn/vn/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*           te:tunnel-ref
            |     +--rw sr-policy*
            |             [policy-color-ref policy-endpoint-ref]
            |             {sr-policy}?
            |        +--rw policy-color-ref       leafref
            |        +--rw policy-endpoint-ref    leafref
            +--rw te-mapping-template-ref?
                    -> /tsmt:te-mapping-templates/te-mapping-template/id
                    {template}?
    augment /l1csm:l1-connectivity/l1csm:access/l1csm:unis/l1csm:uni:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
```

6.3. Network Models

6.3.1. L3NM

```
  module: ietf-l3nm-te-service-mapping
    augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
              /l3vpn-ntw:vpn-service:
      +--rw te-service-mapping!
         +--rw te-mapping
            +--rw map-type?                 identityref
            +--rw te-policy
            |  +--rw color?              uint32
            |  +--rw protection-type?     identityref
            |  +--rw availability-type?   identityref
            +--rw (te)?
            |  +--:(vn)
            |  |  +--rw vn*                    -> /vn:vn/vn/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*           te:tunnel-ref
            |     +--rw sr-policy*
            |             [policy-color-ref policy-endpoint-ref]
            |             {sr-policy}?
            |        +--rw policy-color-ref       leafref
            |        +--rw policy-endpoint-ref    leafref
            +--rw te-mapping-template-ref?
                    -> /tsmt:te-mapping-templates/te-mapping-template/id
                    {template}?
    augment /l3vpn-ntw:l3vpn-ntw/l3vpn-ntw:vpn-services
              /l3vpn-ntw:vpn-service/l3vpn-ntw:vpn-nodes
              /l3vpn-ntw:vpn-node/l3vpn-ntw:vpn-network-accesses
              /l3vpn-ntw:vpn-network-access:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
```

6.3.2. L2NM

```
  module: ietf-l2nm-te-service-mapping
    augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
              /l2vpn-ntw:vpn-service:
      +--rw te-service-mapping!
         +--rw te-mapping
            +--rw map-type?                   identityref
            +--rw te-policy
            |  +--rw color?                uint32
            |  +--rw protection-type?     identityref
            |  +--rw availability-type?   identityref
            +--rw (te)?
            |  +--:(vn)
            |  |  +--rw vn*                   -> /vn:vn/vn/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*           te:tunnel-ref
            |     +--rw sr-policy*
            |             [policy-color-ref policy-endpoint-ref]
            |             {sr-policy}?
            |        +--rw policy-color-ref       leafref
            |        +--rw policy-endpoint-ref    leafref
            +--rw te-mapping-template-ref?
                    -> /tsmt:te-mapping-templates/te-mapping-template/id
                    {template}?
    augment /l2vpn-ntw:l2vpn-ntw/l2vpn-ntw:vpn-services
              /l2vpn-ntw:vpn-service/l2vpn-ntw:vpn-nodes
              /l2vpn-ntw:vpn-node/l2vpn-ntw:vpn-network-accesses
              /l2vpn-ntw:vpn-network-access:
      +--rw (te)?
         +--:(vn)
         |  +--rw vn-ap*   -> /vn:ap/ap/vn-ap/vn-ap-id
         +--:(te)
            +--rw ltp?     te-types:te-tp-id
```

7. YANG Data Models

The YANG codes are as follows:

7.1. ietf-te-service-mapping-types

```
 <CODE BEGINS> file "ietf-te-service-mapping-types@2021-02-22.yang"
 module ietf-te-service-mapping-types {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-te-service-mapping-types";
   prefix tsmt;

   /* Import te-types */

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

   /* Import network model */

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

   /* Import TE model */

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

   /* Import VN model */

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

   /* Import Routing */

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

   /* Import SR Policy */

   import ietf-sr-policy {
     prefix sr-policy;
     reference
```

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

   organization
     "IETF Traffic Engineering Architecture and Signaling (TEAS)
      Working Group";
   contact
     "WG Web:   <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
      variuous service models (e.g., L1CSM, L2SM, L3SM, etc.)

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

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

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

      The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
      NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
      'MAY', and 'OPTIONAL' in this document are to be interpreted as
      described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2021-02-22 {
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   /*
    * Features
    */

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

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

   /*
    * Identity for map-type
    */

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

   identity new {
     base map-type;
     description
       "The new VN/tunnels are binded to the service.";
   }

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

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

   identity soft-isolation {
     base new;
     description
       "Soft-isolation.";
   }

   identity select {
     base map-type;
     description
       "The VPN service selects an existing tunnel with no
        modification.";
   }

   identity modify {
     base map-type;
     description
       "The VPN service selects an existing tunnel and allows to modify
        the properties of the tunnel (e.g., b/w)";
   }

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

   /*
    * Identity for availability-type
    */

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

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

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

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

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

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

   /*
    * Typedef
    */

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

   /*
    * Groupings
    */

   grouping te-ref {
     description
       "The reference to TE.";
     choice te {
       description
         "How the VPN is mapped to a VN, Topology, Tunnel, SR Policy
          etc.";
       case vn {
         leaf-list vn {
           type leafref {
             path "/vn:vn/vn:vn/vn:vn-id";
           }
           description
             "The reference to VN";
           reference
             "RFC 8453: Framework for Abstraction and Control of TE
              Networks (ACTN)";
         }
       }
       case te-topo {
         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 as defined in RFC 8453";
           reference
             "RFC 8795: YANG Data Model for Traffic Engineering (TE)
              Topologies
              RFC 8453: Framework for Abstraction and Control of TE
              Networks (ACTN)";
         }
         leaf abstract-node {
           type leafref {
             path "/nw:networks/nw:network/nw:node/nw:node-id";
           }
           description
             "A reference to the abstract node in TE Topology";
           reference
             "RFC 8795: YANG Data Model for Traffic Engineering (TE)
              Topologies";
         }
       }
       case te-tunnel {
         leaf-list te-tunnel {
           type te:tunnel-ref;
           description
             "Reference to TE Tunnels";
           reference
             "I-D.ietf-teas-yang-te: A YANG Data Model for Traffic
              Engineering Tunnels and Interfaces";
         }
         list sr-policy {
           if-feature "sr-policy";
           key "policy-color-ref policy-endpoint-ref";
           description
             "SR Policy";
           leaf policy-color-ref {
             type leafref {
               path
                 "/rt:routing/sr-policy:segment-routing"
               + "/sr-policy:traffic-engineering/sr-policy:policies"
               + "/sr-policy:policy/sr-policy:color";
             }
             description
               "Reference to sr-policy color";
           }
           leaf policy-endpoint-ref {
             type leafref {
               path
                 "/rt:routing/sr-policy:segment-routing"
               + "/sr-policy:traffic-engineering/sr-policy:policies"
               + "/sr-policy:policy/sr-policy:endpoint";
             }
             description
               "Reference to sr-policy endpoint";
           }
         }
       }
     }
     leaf te-mapping-template-ref {
       if-feature "template";
       type leafref {
         path "/tsmt:te-mapping-templates/"
            + "tsmt:te-mapping-template/tsmt:id";
       }
       description
         "An identifier to the TE Mapping Template where the TE
          constraints and optimization criteria are specified.";
     }
   }

   //grouping

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

   //grouping

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

   //grouping

   grouping te-mapping {
     description
       "Mapping between Services and TE";
     container te-mapping {
       description
         "Mapping between Services and TE";
       leaf map-type {
         type identityref {
           base map-type;
         }
         description
           "Isolation Requirements, Tunnel Bind or
            Tunnel Selection";
       }
       container te-policy {
         uses te-policy;
         description
           "Desired Underlying TE Policy";
       }
       uses te-ref;
     }
   }

   //grouping

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

7.2. Service Models

7.2.1. ietf-l3sm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l3sm-te-service-mapping@2021-02-22.yang"
 module ietf-l3sm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping";
   prefix l3-tsm;

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

   organization
     "IETF Traffic Engineering Architecture and Signaling (TEAS)
      Working Group";
   contact
     "WG Web:   <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) 2020 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject to
      the license terms contained in, the Simplified BSD License set
      forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).
      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      The key words ’MUST’, ’MUST NOT’, ’REQUIRED’, ’SHALL’, ’SHALL
      NOT’, ’SHOULD’, ’SHOULD NOT’, ’RECOMMENDED’, ’NOT RECOMMENDED’,
      ’MAY’, and ’OPTIONAL’ in this document are to be interpreted as
      described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2021-02-22 {
     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 tsmt:te-mapping;
     }
   }

   //augment

   augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
         + "/l3vpn-svc:site-network-accesses"
         + "/l3vpn-svc:site-network-access" {
     description
       "This augment is only valid for TE mapping of L3SM network-access
        to TE endpoints";
     uses tsmt:te-endpoint-ref;
   }

   //augment

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

   augment "/l3vpn-svc:l3vpn-svc/l3vpn-svc:sites/l3vpn-svc:site"
         + "/l3vpn-svc:site-network-accesses"
         + "/l3vpn-svc:site-network-access"
         + "/l3vpn-svc:service/l3vpn-svc:qos/l3vpn-svc:qos-profile"
         + "/l3vpn-svc:qos-profile/l3vpn-svc:custom"
         + "/l3vpn-svc:classes/l3vpn-svc:class" {
     description
       "This augment is for per-class in site-network-access for custom
        QoS profile";
     uses tsmt:te-endpoint-ref;
   }
 }
 <CODE ENDS>
```

7.2.2. ietf-l2sm-te-service-mapping

```
  <CODE BEGINS> file "ietf-l2sm-te-service-mapping@2021-02-22.yang"
  module ietf-l2sm-te-service-mapping {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping";
    prefix l2-tsm;

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

    organization
      "IETF Traffic Engineering Architecture and Signaling (TEAS)
       Working Group";
    contact
      "WG Web:   <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) 2021 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

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

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

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
       'MAY', and 'OPTIONAL' in this document are to be interpreted as
       described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.";

    revision 2021-02-22 {
      description
        "Initial revision.";
      reference
        "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
    }

    /*
     * Augmentation to L2SM
     */

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

    //augment

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

    //augment

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

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

7.2.3. ietf-l1csm-te-service-mapping

```
 <CODE BEGINS> file "ietf-l1csm-te-service-mapping@2021-02-22.yang"
 module ietf-l1csm-te-service-mapping {
   yang-version 1.1;
   namespace
     "urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping";
   prefix l1-tsm;

   import ietf-te-service-mapping-types {
     prefix tsmt;
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }
   import ietf-l1csm {
     prefix l1csm;
     reference
       "I-D.ietf-ccamp-l1csm-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) 2021 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject to
      the license terms contained in, the Simplified BSD License set
      forth in Section 4.c of the IETF Trust’s Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).
      This version of this YANG module is part of RFC XXXX; see the
      RFC itself for full legal notices.

      The key words ’MUST’, ’MUST NOT’, ’REQUIRED’, ’SHALL’, ’SHALL
      NOT’, ’SHOULD’, ’SHOULD NOT’, ’RECOMMENDED’, ’NOT RECOMMENDED’,
      ’MAY’, and ’OPTIONAL’ in this document are to be interpreted as
      described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.";

   revision 2021-02-22 {
     description
       "Initial revision.";
     reference
       "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
   }

   /*
    * Augmentation to L1CSM
    */

   augment "/l1csm:l1-connectivity/l1csm:services/l1csm:service" {
     description
       "L1CSM augmented to include TE parameters and mapping";
     container te-service-mapping {
       presence "Indicates L1 service to TE mapping";
       description
         "Container to augment L1CSM to TE parameters and mapping";
       uses tsmt:te-mapping;
     }
   }

   //augment

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

   //augment
 }
 <CODE ENDS>
```

7.3. Network Models

7.3.1. ietf-l3nm-te-service-mapping

```
  <CODE BEGINS> file "ietf-l3nm-te-service-mapping@202-02-22.yang"
  module ietf-l3nm-te-service-mapping {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping";
    prefix l3nm-tsm;

    import ietf-te-service-mapping-types {
      prefix tsmt;
      reference
        "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
    }
    import ietf-l3vpn-ntw {
      prefix l3vpn-ntw;
      reference
        "I-D.ietf-opsawg-l3sm-l3nm: A Layer 3 VPN Network YANG Model";
    }

    organization
      "IETF Traffic Engineering Architecture and Signaling (TEAS)
       Working Group";
    contact
      "WG Web:   <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
       Network Model (L3NM) to the TE and VN.

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

       Redistribution and use in source and binary forms, with or
       without modification, is permitted pursuant to, and subject to
       the license terms contained in, the Simplified BSD License set
       forth in Section 4.c of the IETF Trust’s Legal Provisions
       Relating to IETF Documents
       (https://trustee.ietf.org/license-info).
       This version of this YANG module is part of RFC XXXX; see the
       RFC itself for full legal notices.

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
       'MAY', and 'OPTIONAL' in this document are to be interpreted as
       described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.";

    revision 2021-02-22 {
      description
        "Initial revision.";
      reference
        "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
    }

    /*
     * Augmentation to L3NM
     */

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

    //augment

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

    //augment
  }
  <CODE ENDS>
```

7.3.2. ietf-l2nm-te-service-mapping

```
  <CODE BEGINS> file "ietf-l2nm-te-service-mapping@2021-02-22.yang"
  module ietf-l2nm-te-service-mapping {
    yang-version 1.1;
    namespace
      "urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping";
    prefix l2nm-tsm;

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

    organization
      "IETF Traffic Engineering Architecture and Signaling (TEAS)
       Working Group";
    contact
      "WG Web:   <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
       Network Model (L2NM) to the TE and VN.

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

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

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

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
       NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
       'MAY', and 'OPTIONAL' in this document are to be interpreted as
       described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
       they appear in all capitals, as shown here.";

    revision 2021-02-22 {
      description
        "Initial revision.";
      reference
        "RFC XXXX:  Traffic Engineering and Service Mapping Yang Model";
    }

    /*
     * Augmentation to L2NM
     */

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

    //augment

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

    //augment
  }
  <CODE ENDS>
```

8. Security Considerations

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

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

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

- /l3vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.

- /l3vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.

- /l2vpn-svc/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Service mapping.

- /l2vpn-svc/sites/site/site-network-accesses/site-network-access/te/ - configure TE Endpoint mapping.

- /l1-connectivity/services/service/te-service-mapping/te-mapping/ - configure TE Service mapping.

- /l1-connectivity/access/unis/uni/te/ - configure TE Endpoint mapping.

- /l3vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Network mapping.

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

- /l2vpn-ntw/vpn-services/vpn-service/te-service-mapping/te-mapping/ - configure TE Network mapping.

- /l2vpn-ntw/vpn-services/vpn-service/vpn-nodes/vpn-node/vpn-network-accesses/vpn-network-access/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 six URIs in the "IETF XML Registry" [RFC3688]. Following the format in RFC 3688, the following registrations are requested -

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

URI: urn:ietf:params:xml:ns:yang:ietf-l3sm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2sm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l1csm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l3nm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2nm-te-service-mapping
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

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

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

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

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

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

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

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

10. Acknowledgements

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

11. References

11.1. Normative References

[I-D.ietf-ccamp-l1csm-yang]
Lee, Y., Lee, K., Zheng, H., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang-13 (work in progress), November 2020.

[I-D.ietf-opsawg-l2nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm-01 (work in progress), November 2020.

[I-D.ietf-opsawg-l3sm-l3nm]
barguil, s., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm-05 (work in progress), October 2020.

[I-D.ietf-spring-sr-policy-yang]
Raza, K., Sawaya, R., Shunwan, Z., Voyer, D., Durrani, M., Matsushima, S., and V. Beeram, "YANG Data Model for Segment Routing Policy", draft-ietf-spring-sr-policy-yang-00 (work in progress), September 2020.

[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-10 (work in progress), November 2020.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", draft-ietf-teas-yang-te-25 (work in progress), July 2020.

[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>.

[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>.

[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>.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/info/rfc8299>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

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

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

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[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>.

[RFC8776] Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "Common YANG Data Types for Traffic Engineering", RFC 8776, DOI 10.17487/RFC8776, June 2020, <https://www.rfc-editor.org/info/rfc8776>.

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/info/rfc8795>.

11.2. Informative References

[I-D.ietf-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-06 (work in progress), August 2020.

[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>.

Appendix A. Examples

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

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

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

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

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

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

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

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

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

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

Appendix B. Discussion

- While the support to bind a tunnel to the VPN is supported. We do not have a mechanism to map traffic to a path. The input can come from the user. E.g. the enterprise customer can tell, the traffic from source X on port Y should go with delay less than Z. Further discussion is required on how and where to model these.

- Support for Calendaring and scheduling TE resources.

Appendix C. Contributor Addresses

Adrian Farrel
Old Dog Consulting

EMail: adrian@olddog.co.uk

Italo Busi
Huawei Technologies

EMail: Italo.Busi@huawei.com

Haomian Zheng
Huawei Technologies

EMail: zhenghaomian@huawei.com

Anton Snitser
Sedonasys

EMail: antons@sedonasys.com

SAMIER BARGUIL GIRALDO
Telefonica

EMail: samier.barguilgiraldo.ext@telefonica.com

Oscar Gonzalez de Dios
Telefonica

EMail: oscar.gonzalezdedios@telefonica.com

Carlo Perocchio
Ericsson

EMail: carlo.perocchio@ericsson.com

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

Authors' Addresses

Young Lee (editor)
Samsung Electronics

Email: younglee.tx@gmail.com

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

Email: dhruv.ietf@gmail.com

Giuseppe Fioccola
Huawei Technologies

Email: giuseppe.fioccola@huawei.com

Qin Wu (editor)
Huawei Technologies

Email: bill.wu@huawei.com

Daniele Ceccarelli
Ericsson
Torshamnsgatan,48
Stockholm, Sweden

Email: daniele.ceccarelli@ericsson.com

Jeff Tantsura
Apstra

Email: jefftant.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2021

B. Wu
D. Dhody
Huawei Technologies
L. Han
China Mobile
R. Rokui
Nokia

February 21, 2021

A Yang Data Model for IETF Network Slice NBI
draft-wd-teas-ietf-network-slice-nbi-yang-02

Abstract

This document provides a YANG data model for the IETF Network Slice NBI (Northbound Interface). The model can be used by a higher level system to request configuration, and management IETF Network Slices from the IETF Network Slice Controller (NSC).

The YANG modules in this document conforms to the Network Management Datastore Architecture (NMDA) defined in RFC 8342.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document provides a YANG [RFC7950] data model for the IETF Network Slice NBI.

The YANG model discussed in this document is defined based on the description of the IETF Network Slice in [I-D.ietf-teas-ietf-network-slice-definition] and [I-D.nsdt-teas-ns-framework], which is used to operate IETF Network Slice during the IETF Network Slice instantiation. This YANG model supports various oprations on IETF Network Slices such as creation, modification, deletion, and monitoring of IETF Network Slices.

The IETF Network Slice Controller (NSC) provides a Northbound Interface (NBI) that allows consumers of network slices to request and monitor IETF network slices. Consumers operate on abstract IETF network slices, with details related to their realization hidden.

The NSC takes requests from a management system or other application via an NBI. This interface carries data objects the IETF network slice user provides, describing the needed IETF network slices in terms of topology, applicable service level objectives (SLO), and any monitoring and reporting requirements that may apply. The NBI conveys the generic IETF network slice requirements. These may then be realized using an SBI within the NSC.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the consumer, which is classified as Customer Service Model in [RFC8309].

It will be up to the management system or NSC (IETF Network Slice controller) to take this model as an input and use other management system or specific configuration models to configure the different network elements to deliver an IETF Network Slice. The YANG models can be used with network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The details of how the IETF network slices are realized by the NSC is out of scope for this document.

The IETF Network Slice operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture [RFC8342].

2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

o client

o configuration data

o state data

This document makes use of the following terminology introduced in the YANG 1.1 Data Modeling Language [RFC7950]:

- augment

- data model

- data node

This document also makes use of the following terminology introduced in the IETF Network Slice definition draft [I-D.ietf-teas-ietf-network-slice-definition]:

- NBI: Northbound Interface

- NS: IETF Network Slice

- NSC: IETF Network Slice Controller

- NSE: Network Slice Endpoint

- SLO: Service Level Objective

This document defines the following new terminology:

- IETF Network Slice Member (Network-Slice-Member): In the context of an IETF Network Slice, an IETF Network-Slice-Member is an abstract entity which represents a particular connection between a pair of NSEs. An IETF Network Slice can has one or multiple members.

2.1. Tree Diagrams

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

3. IETF Network Slice NBI Model Usage

The intention of the IETF Network Slice NBI model is to allow the consumer, e.g. a higher level management system, to request and monitor IETF Network Slices. In particular, the model allows consumers to operate in an abstract, technology-agnostic manner, with realization details hidden.

According to the [I-D.ietf-teas-ietf-network-slice-definition] description, the NBI model is applicable to use case such as (but not limited to) Network wholesale services, Network infrastructure sharing among operators, NFV connectivity and Data Center Interconnect and 5G E2E network slice.

As Figure 1 shows, in all these use-cases, the NBI model is used by the higher management system (i.e the consumer of the IETF network slice controller) to communicate with IETF Network Slice controller for life cycle manage of IETF Network Slices including both enablement and monitoring. For example, in 5G E2E network slicing use-case the E2E network slice orchestrator acts as the higher layer system to request the IETF Network Slices. The interface is used to support dynamic IETF Network Slice creation and its lifecycle management to facilitate end-to-end network slice services.

```
             +-----------------------------------------+
             |   IETF Network Slice Consumer           |
             |                                         |
             +----------------+------------------------+
                              |
                              |
                              |IETF Network Slice NBI YANG
                              |
        +---------------------+---------------------------+
        |      IETF Network Slice Controller (NSC)        |
        +-------------------------------------------------+
```

Figure 1: IETF Network Slice NBI Model Context

4. IETF Network Slice NBI Model Overview

As defined in [I-D.ietf-teas-ietf-network-slice-definition], an IETF network slice is a logical network connecting a number of endpoints with specified SLOs. The connectivity can be point-to-point, multipoint-to-point, point-to-multipoint or multipoint-to-multipoint. In addition, a minimum set of SLOs is defined, including but not limited to bandwidth, delay, and etc. An example of an IETF network slice is shown in Figure 2 .

```
          +----------------------------------------------+
          |                                              |
     NSE1 O------------------+                           |
          .                  +---------------------------O NSE2
          .                  |                           .
          .                  |multipoint-to-multipoint   .
                             |
                             +---------------------------O NSEn
     NSEm O------------------+                           |
          |                                              |
          +----------------------------------------------+

          |                                              |
          |<-----------An IETF Network Slice  ---------->|
          |         between endpoints NSE1 to NSEn       |

    Legend:
         NSE: IETF Network Slice Endpoint
           O: Represents IETF Network Slice Endpoints
```

Figure 2: An IETF Network Slice Example

Draft [I-D.ietf-teas-ietf-network-slice-definition] introduces the IETF network slice endpoints (NSEs) which are conceptual points of connection to IETF network slice. As such, they are ingress/egress point where the traffic enters/exits the IETF network slice. In other words, they are the edge of the IETF network slices.

When IETF network slice controller (NSC) receives a message via its NBI for creation/modification of an IETF network slice, it uses the provided IETF network slice endpoints to map them to appropriate services/tunnels/paths endpoints in the underlay IETF network. It then uses services/tunnels/paths endpoints to realize the IETF network slice.

The IETF Network Slice ("ietf-network-slice") is defined to manage network slices in the IETF network. In particular, the 'ietf-network-slice' module can be used to create, modify, and monitor network slices of an IETF network.

The 'ietf-network-slice' module uses two main nodes: list 'ietf-network-slice' and container 'ns-templates' (see Figure 3).

The 'ietf-network-slice' list includes the set of IETF Network slices managed within IETF network. 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice. Under the "ietf-network-slice", list "ns-endpoint" is used to abstract the NSEs, e.g. NSEs in the example above.

The 'ns-templates' container is used by the NSC to maintain a set of common network slice templates that apply to one or several IETF Network Slices.

The figure below describes the overall structure of the YANG module:

```
   module: ietf-network-slice
     +--rw ietf-network-slices
        +--rw ns-templates
        |  +--rw slo-template* [id]
        |     +--rw id                      string
        |     +--rw template-description?   string
        +--rw ietf-network-slice* [ns-id]
           +--rw ns-id                 string
           +--rw ns-description?       string
           +--rw ns-tag*               string
           +--rw ns-topology?          identityref
           +--rw (ns-slo-policy)?
           |  +--:(standard)
           |  |  +--rw slo-template?   leafref
           |  +--:(custom)
           |     +--rw slo-policy
           |        +--rw policy-description?   string
           |        +--rw ns-metric-bounds
           |           +--rw ns-metric-bound* [metric-type]
           |              +--rw metric-type          identityref
           |              +--rw metric-unit          string
           |              +--rw value-description?   string
           |              +--rw boundary?            uint64
           +--rw status
           |  +--rw admin-enabled?   boolean
           |  +--ro oper-status?     operational-type
           +--rw ns-endpoint* [ep-id]
           |  +--rw ep-id                string
           |  +--rw ep-description?      string
           |  +--rw ep-role?             identityref
           |  +--rw location
           |  |  +--rw altitude?    int64
           |  |  +--rw latitude?    decimal64
           |  |  +--rw longitude?   decimal64
           |  +--rw node-id?             string
           |  +--rw ep-ip?               inet:host
           |  +--rw ns-match-criteria
           |  |  +--rw ns-match-criteria* [match-type]
           |  |     +--rw match-type    identityref
           |  |     +--rw value?        string
           |  +--rw ep-network-access* [network-access-id]
           |  |  +--rw network-access-id             string
           |  |  +--rw network-access-description?   string
           |  |  +--rw network-access-node-id?       string
           |  |  +--rw network-access-tp-id?         string
           |  |  +--rw network-access-tp-ip?         inet:host
           |  +--rw ep-rate-limit
           |  |  +--rw incoming-throughput
           |  |  |  +--rw maximum-throughput?   te-types:te-bandwidth
           |  |  +--rw outgoing-throughput
           |  |     +--rw maximum-throughput?   te-types:te-bandwidth
           |  +--rw ep-protocol
           |  +--rw status
           |  |  +--rw admin-enabled?   boolean
           |  |  +--ro oper-status?     operational-type
           |  +--ro ep-monitoring
           |     +--ro incoming-utilized-bandwidth?
           |     |       te-types:te-bandwidth
           |     +--ro incoming-bw-utilization        decimal64
           |     +--ro outgoing-utilized-bandwidth?
           |     |       te-types:te-bandwidth
           |     +--ro outgoing-bw-utilization        decimal64
           +--rw ns-member* [ns-member-id]
              +--rw ns-member-id             uint32
              +--rw ns-member-description?   string
              +--rw src
              |  +--rw src-ep-id?   leafref
              +--rw dest
              |  +--rw dest-ep-id?   leafref
              +--rw monitoring-type?         ns-monitoring-type
              +--ro ns-member-monitoring
                 +--ro latency?      yang:gauge64
                 +--ro jitter?       yang:gauge32
                 +--ro loss-ratio?   decimal64
```

Figure 3

5. IETF Network Slice Templates

The ’ns-templates’ container (Figure 3) is used by service provider of the NSC to define and maintain a set of common IETF Network Slice templates that apply to one or several IETF Network Slices. The exact definition of the templates is deployment specific to each network provider. The model includes only the identifiers of SLO-templates. When creation of IETF Network slice, the SLO policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the upper layer management system:

```
   {
     "ietf-network-slices": {
       "ns-templates": {
         "slo-template": [
          {
            "id":"GOLD-template",
            "template-description": "Bandwidth: 1 Gbps, delay 100ms "
           },
          {
            "id":"PLATINUM-template",
            "template-description": "Bandwidth: 1 Gbps, delay 50ms "
           },
          ],
         }
      }
   }
```

6. IETF Network Slice Modeling Description

The 'ietf-network-slice' is the data structure that abstracts an IETF Network Slice of the IETF network. Each 'ietf-network-slice' is uniquely identified by an identifier: 'ns-id'.

An IETF Network Slice has the following main parameters:

- "ns-id": Is an identifier that is used to uniquely identify the IETF Network Slice within NSC.

- "ns-description": May be provided to help identify an IETF Network Slice.

- "ns-topology": Indicates the network topology for the IETF Network Slice: Hub-Spoke, Any-to-Any, and Custom.

- "status": Enable the control of the operative and administrative status of the IETF Network Slice, can be used as indicator to detect network slice anomalies.

- "ns-tag": The list is to show the correlation between higher level function and the IETF network slices. If provided, this parameter may be used by IETF Network Slice Controller (NSC) during the realization. It may also be used by NSC for monitoring and assurance of the IETF network slices where NSC can notify the higher system by issuing the notifications. It is noted that a single higher level consumer might have multiple IETF Network Slices for a single application. This attribute may be used by NSC to also correlated multiple IETF network slices for a single application.

- "ns-slo-policy": Defines SLO policy for the "ietf-network-slice". More description are provided in Section 6.1

The "ns-endpoint" is an abstrac entity that represents a set of matching rules applied to an IETF network edge device or a customer network edge device involved in the IETF Network Slice and each 'ns-endpoint' belongs to a single 'ietf-network-slice'. More description are provided in Section 6.3

6.1. IETF Network Slice Topology

An IETF Network Slice can be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), or multipoint-to-multipoint (MP2MP) based on the consumer's traffic pattern requirements.

Therefore, the "ns-topology" under the node "ietf-network-slice" is required for configuration. The model supports any-to-any, Hub and Spoke (where Hubs can exchange traffic), and the different combinations. New topologies could be added via augmentation. By default, the any-to-any topology is used.

In addition, "ep-role" under the node "ns-endpoint" also needs to be defined, which specifies the role of the NSE in a particular Network Slice topology. In the any-to-any topology, all NSEs MUST have the same role, which will be "any-to-any-role". In the Hub-and-Spoke topology, NSEs MUST have a Hub role or a Spoke role.

6.2. IETF Network Slice SLO Policy

As defined in [I-D.ietf-teas-ietf-network-slice-definition], the SLO policy of an IETF Network Slice defines the minimum IETF Network Slice SLO attributes, and additional attributes can be added as needed.

"ns-slo-policy" is used to represent a specific SLO policy. During the creation of an IETF Network Slice, the policy can be specified either by a standard SLO template or a customized SLO policy.

The model allows multiple SLO attributes to be combined to meet different SLO requirements. For example, some NSs are used for video services and require high bandwidth, some NSs are used for key business services and request low latency and reliability, and some NSs need to provide connections for a large number of NSEs. That is, not all SLO attributes must be specified to meet the particular requirements of a slice.

"ns-metric-bounds" contains all these variations, which includes a list of "ns-metric-bound" and each "ns-metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and the YANG module supports the following options:

"network-slice-slo-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. The unit is data rate per second. And the bandwidth is unidirectional.

"network-slice-slo-one-way-delay": Indicates the maximum one-way latency between two NSE. The unit is micro seconds.

"network-slice-slo-two-way-delay": Indicates the maximum round trip latency between two NSE. The unit is micro seconds.

"network-slice-slo-jitter": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one- way delay between sequential packets in a flow.

"network-slice-slo-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"network-slice-slo-availability": Is defined as the ratio of up-time to total_time(up-time+down-time), where up-time is the time the IETF Network Slice is available in accordance with the SLOs associated with it.

Some other Network Slice objectives, such as MTU and security which can be added when needed. MTU specifies the maximum packet length that the network slice guarantee to be able to carry across.

Note: About the definition of SLO parameters, the author is discussing to reuse the TE-Types grouping definition as much as possible, to avoid duplication of definitions.

The following shows an example where a network slice policy can be configured:

```
   {
     "ietf-network-slices": {
       "ietf-network-slice": {
         "slo-policy": {
           "policy-description":"video-service-policy",
           "ns-metric-bounds": {
               "ns-metric-bound": [
                {
                   "metric-type": "network-slice-slo-bandwidth",
                   "metric-unit": "mbps"
                   "boundary": "1000"
                },
                {
                   "metric-type": "network-slice-slo-availability",
                   "boundary": "99.9%"
                },
               ],
            }
         }
       }
     }
   }
```

6.3. IETF Network Slice Endpoint (NSE)

An IETF Network Slice Endpoint has several characteristics:

- "ep-id": Uniquely identifies the NSE within Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice.

- "location": is NSE location information that facilities NSC easy identification of a NSE.

- "ep-role": Is a topology role of a NSE belonging to an IETF network slice, as described in Section 6.1. The "ep-role" leaf defines the role of the endpoint in a particular NS topology. In the NS any-to-any topology, all NSEs MUST have the same role, which will be "any-to-any-role".

- "node-id": is NSE node information that facilities NSC easy identification of a NSE.

- "ep-ip": is NSE IP information that facilities NSC easy identification of a NSE.

- "ns-match-criteria": Is used to define matching policies to apply on a given NSE.

- "ep-network-access": Is the list that includes the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received.

- "ep-rate-limit": Is to set rate-limiting policies to apply on a given NSE, including ingress and egress traffic to ensure access security. When applied in the incoming direction, the rate-limit is applicable to the traffic from the NSE to the IETF scope Network that passes through the external interface. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the NSE of that particular NS.

- "ep-protocol": Specify the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols,etc.

- "status": Enable the control of the operative and administrative status of the NSE, can be used as indicator to detect NSE anomalies.

An NSE belong to a single IETF Network Slice. An IETF Network Slice involves two or more NSEs. An IETF Network Slice can be modified by adding new "ns-endpoint" or removing existing "ns-endpoint".

A NSE is used to define the matching rule on the customer traffic that can be injected to an IETF Network Slice. "network-slice-match-criteria" is defined to support different options. Classification can be based on many criteria, such as:

- Physical interface: Indicates all the traffic received from the interface belongs to the IETF Network Slice.

- Logical interface: For example, a given VLAN ID is used to identify an IETF Network Slice.

- Encapsulation in the traffic header: For example, a source IP address is used to identify an IETF Network Slice.

To illustrate the use of NSE parameters, the below are two examples. How the NSC realize the mapping is out of scope for this document.

- NSE mapping to PE example: As shown in Figure 4 , consumer of the IETF network slice would like to connect two NSEs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice endpoints are mapped to physical interfaces of PE nodes. The IETF network slice controller (NSC) uses "node-id" (PE device ID), "ep-network-access" (Two PE interfaces) to map the interfaces and corresponding services/tunnels/paths.

```
              NSE1                                     NSE2
       (With PE1 parameters)                       (with PE2 parameters)
               o<--------- IETF Network Slice 1 ------->o
               +     |                            |     +
               +     |<----------- S1 ----------->|     +
               +     |                            |     +
               +     |    |<------ T1 ------>|    |     +
                 +   v    v                  v    v   +
                   + +----+                  +----+ +
    +-----+    |     | PE1|==================| PE2|          +-----+
    |     |----------X    |                  |    |     |    |     |
    |     |    |     |    |                  |    X----------|     |
    |     |----------X    |                  |    |     |    |     |
    +-----+    |     |    |==================|    |     |    +-----+
               AC    +----+                  +----+     AC
    Customer         Provider                Provider        Customer
    Edge 1           Edge 1                  Edge 2           Edge 2

  Legend:
       O: Representation of the IETF network slice endpoints (NSE)
       +: Mapping of NES to PE or CE nodes on IETF network
       X: Physical interfaces used for realization of IETF network slice
       S1: L0/L1/L2/L3 services used for realization of IETF network slice
       T1: Tunnels used for realization of IETF network slice
```

Figure 4

- NSE mapping to CE example: As shown in Figure 5 , consumer of the IETF network slice would like to connect two NSEs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice endpoints (NSE) might be mapped to tunnels endpoints on CE nodes (see 3GPP TS 28.541 V17.1.0 section 6.3.17 EP_Transport). The IETF network slice controller (NSC) uses "node-id" (CE device ID) , "ep-ip" (CE tunnel endpoint IP), "network-slice-match-criteria" (VLAN interface), "ep-network-access" (Two nexthop interfaces) to map underlay services/tunnels/paths.

```
              NSE3                                     NSE4
       (With CE1 parameters)                       (with CE2 parameters)
               o<--------- IETF Network Slice 2 ------->o
               +      |                            |    +
               +      |<----------- S2 ----------->|    +
               +      |                            |    +
             +        |    |<------ T2 ------>|    |      +
           +          v    v                  v    v        +
         +     AC     +----+                  +----+          +
    +-----+    |     | PE1|==================| PE2|          +-----+
    |     |----------X    |                  |    |     |    |     |
    |     |    |     |    |                  |    X----------|     |
    |     |----------X    |                  |    |     |    |     |
    +-----+    |     |    |==================|    |     |    +-----+
               AC    +----+                  +----+     AC
    Customer         Provider                Provider         Customer
    Edge 1           Edge 1                  Edge 2           Edge 2

  Legend:
       O: Representation of the IETF network slice endpoints (NSE)
       +: Mapping of NES to PE or CE nodes on IETF network
       X: Physical interfaces used for realization of IETF network slice
       S2: L0/L1/L2/L3 services used for realization of IETF network slice
       T2: Tunnels used for realization of IETF network slice
```

Figure 5

7. IETF Network Slice Monitoring

An IETF Network Slice is a connectivity with specific SLO characteristics, including bandwidth, QoS metric, etc. The connectivity is a combination of logical connections, represented by Network-Slice-Members.

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

- Per NS connection: specified in 'network-slice-member-monitoring' under the "network-slice-member"

- Per NS Endpoint: specified in 'endpoint-monitoring' under the "network-slice-endpoint"

This model does not define monitoring enabling methods. The mechanism defined in [RFC8640] and [RFC8641] can be used for either periodic or on-demand subscription.

By specifying subtree filters or xpath filters to 'ns-member' or 'ns-endpoint' ,so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the client management system could initiate modification based on the IETF Network Slice running status.

8. IETF Network Slice NBI Module

```
<CODE BEGINS> file "ietf-network-slice@2021-02-19.yang"
module ietf-network-slice {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-slice";
  prefix ietf-ns;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-te-types {
    prefix te-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: Bo Wu <lana.wubo@huawei.com>
           : Dhruv Dhody <dhruv.ietf@gmail.com>";
  description
    "This module contains a YANG module for the IETF Network Slice.

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

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

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

  revision 2021-02-19 {
    description
      "initial version.";
    reference
      "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
  }

  /* Features */
  /* Identities */

  identity network-slice-topology {
    description
      "Base identity for IETF Network Slice topology.";
  }

  identity any-to-any {
    base network-slice-topology;
    description
      "Identity for any-to-any IETF Network Slice topology.";
  }

  identity hub-spoke {
    base network-slice-topology;
    description
      "Identity for Hub-and-Spoke IETF Network Slice topology.";
  }

  identity custom {
    base network-slice-topology;
    description
      "Identity of a custom NS topology where Hubs
       can act as Spoke for certain parts of
       the network or Spokes as Hubs.";
  }

  identity endpoint-role {
    description
      "Base identity of a NSE role in an IETF Network Slice topology.";
  }

  identity any-to-any-role {
    base endpoint-role;
    description
      "Identity of any-to-any NS.";
  }
  identity spoke-role {
    base endpoint-role;
    description
      "A NSE is acting as a Spoke.";
  }

  identity hub-role {
    base endpoint-role;
    description
      "A NSE is acting as a Hub.";
  }

  identity custom-role {
    base endpoint-role;
    description
      "A NSE is custom role in the NS.";
  }

  identity network-slice-slo-metric-type {
    description
      "Base identity for Network Slice SLO metric type";
  }

  identity network-slice-slo-two-way-delay {
    base network-slice-slo-metric-type;
    description
      "SLO delay metric.";
  }

  identity network-slice-slo-one-way-delay {
    base network-slice-slo-metric-type;
    description
      "SLO delay metric.";
  }

  identity network-slice-slo-jitter {
    base network-slice-slo-metric-type;
    description
      "SLO jitter metric.";
  }

  identity network-slice-slo-loss {
    base network-slice-slo-metric-type;
    description
      "SLO loss metric .";
  }

  identity network-slice-slo-availability {
    base network-slice-slo-metric-type;
    description
      "SLO availability level.";
  }

  identity network-slice-slo-bandwidth {
    base network-slice-slo-metric-type;
    description
      "SLO bandwidth metric.";
  }

  identity network-slice-match-type {
    description
      "Base identity for Network Slice traffic match type";
  }

  identity network-slice-phy-interface-match {
    base network-slice-match-type;
    description
      "VLAN as Network Slice traffic match criteria.";
  }

  identity network-slice-vlan-match {
    base network-slice-match-type;
    description
      "VLAN as Network Slice traffic match criteria.";
  }

  identity network-slice-label-match {
    base network-slice-match-type;
    description
      "Label as Network Slice traffic match criteria.";
  }

  /*
   * Identity for availability-type
   */

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

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

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

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

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

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

  /* typedef */

  typedef operational-type {
    type enumeration {
      enum up {
        value 0;
        description
          "Operational status UP.";
      }
      enum down {
        value 1;
        description
          "Operational status DOWN";
      }
      enum unknown {
        value 2;
        description
          "Operational status UNKNOWN";
      }
    }
    description
      "This is a read-only attribute used to determine the
```

```
       status of a particular element";
  }

  typedef ns-monitoring-type {
    type enumeration {
      enum one-way {
        description
          "represents one-way monitoring type";
      }
      enum two-way {
        description
          "represents two-way monitoring type";
      }
    }
    description
      "enumerated type of monitoring on a network-slice-member ";
  }

  /* Groupings */

  grouping status-params {
    description
      "Grouping used to join operational and administrative status";
    container status {
      description
        "Container for status of administration and operational";
      leaf admin-enabled {
        type boolean;
        description
          "Administrative Status UP/DOWN";
      }
      leaf oper-status {
        type operational-type;
        config false;
        description
          "Operations status";
      }
    }
  }

  grouping network-slice-match-criteria {
    description
      "Grouping for Network Slice match definition.";
    container ns-match-criteria {
      description
        "Describes Network Slice match criteria.";
      list ns-match-criteria {
        key "match-type";
        description
          "List of Network Slice traffic criteria";
        leaf match-type {
          type identityref {
            base network-slice-match-type;
          }
          description
            "Identifies an entry in the list of match-type for
             the Network Slice.";
        }
        leaf value {
          type string;
          description
            "Describes Network Slice match criteria,e.g. IP address,
             VLAN, etc.";
        }
      }
    }
  }

  grouping network-slice-metric-bounds {
    description
      "Network Slice metric bounds grouping";
    container ns-metric-bounds {
      description
        "Network Slice metric bounds container";
      list ns-metric-bound {
        key "metric-type";
        description
          "List of Network Slice metric bounds";
        leaf metric-type {
          type identityref {
            base network-slice-slo-metric-type;
          }
          description
            "Identifies an entry in the list of metric-types
             bound for the Network Slice.";
        }
        leaf metric-unit {
          type string;
          mandatory true;
          description
            "The metric unit of the parameter.
             For example, s, ms, ns, and so on.";
        }
        leaf value-description {
          type string;
          description
```

```
            "The description of previous value. ";
        }
        leaf boundary {
          type uint64;
          default "0";
          description
            "Boundary on network-slice-member metric. A zero indicate
             an unbounded upper limit for the specific metric-type";
        }
      }
    }
  }

  grouping ep-network-accesses {
    description
      "Grouping for endpoint network access definition.";
    list ep-network-access {
      key "network-access-id";
      description
        "IETF Network Slice endpoint network access related parameters";
      leaf network-access-id {
        type string;
        description
          "unique identifier for the referred endpoint network access";
      }
      leaf network-access-description {
        type string;
        description
          "endpoint network access description";
      }
      leaf network-access-node-id {
        type string;
        description
          "EP network access node ID in the case of multi-homing.";
      }
      leaf network-access-tp-id {
        type string;
        description
          "EP network access termination port ID.";
      }
      leaf network-access-tp-ip {
        type inet:host;
        description
          "The IP address of EP network access.";
      }
    }
  }
  grouping endpoint-monitoring-parameters {
    description
      "Grouping for endpoint-monitoring-parameters.";
    container ep-monitoring {
      config false;
      description
        "Container for endpoint-monitoring-parameters.";
      leaf incoming-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Bandwidth utilization that represents the actual
           utilization of the incoming endpoint.";
      }
      leaf incoming-bw-utilization {
        type decimal64 {
          fraction-digits 5;
          range "0..100";
        }
        units "percent";
        mandatory true;
        description
          "To be used to define the bandwidth utilization
           as a percentage of the available bandwidth.";
      }
      leaf outgoing-utilized-bandwidth {
        type te-types:te-bandwidth;
        description
          "Bandwidth utilization that represents the actual
           utilization of the incoming endpoint.";
      }
      leaf outgoing-bw-utilization {
        type decimal64 {
          fraction-digits 5;
          range "0..100";
        }
        units "percent";
        mandatory true;
        description
          "To be used to define the bandwidth utilization
           as a percentage of the available bandwidth.";
      }
    }
  }

  grouping common-monitoring-parameters {
    description
      "Grouping for link-monitoring-parameters.";
    leaf latency {
      type yang:gauge64;
      units "usec";
      description
        "The latency statistics per Network Slice member.
         [RFC2681] and [RFC7679] discuss round trip times and one-way
         metrics, respectively";
    }
    leaf jitter {
      type yang:gauge32;
      description
        "The jitter statistics per Network Slice member
         as defined by [RFC3393].";
    }
    leaf loss-ratio {
      type decimal64 {
        fraction-digits 6;
        range "0 .. 50.331642";
      }
      description
        "Packet loss as a percentage of the total traffic
         sent over a configurable interval. The finest precision is
         0.000003%. where the maximum 50.331642%.";
      reference
        "RFC 7810, section-4.4";
    }
  }

  grouping geolocation-container {
    description
      "A grouping containing a GPS location.";
    container location {
      description
        "A container containing a GPS location.";
      leaf altitude {
        type int64;
        units "millimeter";
        description
          "Distance above the sea level.";
      }
      leaf latitude {
        type decimal64 {
          fraction-digits 8;
          range "-90..90";
        }
        description
          "Relative position north or south on the Earth's surface.";
      }
      leaf longitude {
        type decimal64 {
          fraction-digits 8;
          range "-180..180";
        }
        description
          "Angular distance east or west on the Earth’s surface.";
      }
    }
    // gps-location
  }

  // geolocation-container

  grouping endpoint {
    description
      "IETF Network Slice endpoint related information";
    leaf ep-id {
      type string;
      description
        "unique identifier for the referred IETF Network
         Slice endpoint";
    }
    leaf ep-description {
      type string;
      description
        "endpoint name";
    }
    leaf ep-role {
      type identityref {
        base endpoint-role;
      }
      default "any-to-any-role";
      description
        "Role of the endpoint in the IETF Network Slice.";
    }
    uses geolocation-container;
    leaf node-id {
      type string;
      description
        "Uniquely identifies an edge node within the IETF slice
         network.";
    }
    leaf ep-ip {
      type inet:host;
      description
        "The address of the endpoint IP address.";
    }
    uses network-slice-match-criteria;
    uses ep-network-accesses;
    container ep-rate-limit {
      description
        "Container for the asymmetric traffic control";
      container incoming-throughput {
        description
          "Container for the incoming traffic policy";
        leaf maximum-throughput {
          type te-types:te-bandwidth;
          description
            "If maximum-throughput is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
      }
      container outgoing-throughput {
        description
          "Container for the bandwidth policy";
        leaf maximum-throughput {
          type te-types:te-bandwidth;
          description
            "If maximum-throughput is 0, it means best effort, no
             minimum throughput is guaranteed.";
        }
      }
    }
    container ep-protocol {
      description
        "Describes protocol for the Network Slice Endpoint.";
    }
    uses status-params;
    uses endpoint-monitoring-parameters;
  }

  //network-slice-endpoint

  grouping network-slice-member {
    description
      "network-slice-member is described by this container";
    leaf ns-member-id {
      type uint32;
      description
        "network-slice-member identifier";
    }
    leaf ns-member-description {
      type string;
      description
        "network-slice-member description";
    }
    container src {
      description
        "the source of Network Slice link";
      leaf src-ep-id {
        type leafref {
          path "/ietf-network-slices/ietf-network-slice/"
             + "ns-endpoint/ep-id";
        }
        description
          "reference to source Network Slice endpoint";
      }
    }
    container dest {
      description
        "the destination of Network Slice link ";
      leaf dest-ep-id {
        type leafref {
          path "/ietf-network-slices/ietf-network-slice"
             + "/ns-endpoint/ep-id";
        }
        description
          "reference to dest Network Slice endpoint";
      }
    }
    leaf monitoring-type {
      type ns-monitoring-type;
      description
        "One way or two way monitoring type.";
    }
    container ns-member-monitoring {
      config false;
      description
        "SLO status Per network-slice endpoint to endpoint ";
      uses common-monitoring-parameters;
    }
  }

  //network-slice-member

  grouping slice-template {
    description
      "Grouping for slice-templates.";
    container ns-templates {
      description
        "Contains a set of network slice templates to
         reference in the IETF network slice.";
      list slo-template {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the SLO Template to be used.
             Local administration meaning.";
        }
        leaf template-description {
          type string;
          description
            "Description of the SLO policy template.";
        }
        description
          "List for SLO template identifiers.";
      }
    }
  }

  /* Configuration data nodes */

  container ietf-network-slices {
    description
      "IETF network-slice configurations";
    uses slice-template;
    list ietf-network-slice {
      key "ns-id";
      description
        "a network-slice is identified by a network-slice-id";
      leaf ns-id {
        type string;
        description
          "A unique network-slice identifier across an IETF NSC ";
      }
      leaf ns-description {
        type string;
        description
          "Give more description of the network slice";
      }
      leaf-list ns-tag {
        type string;
        description
          "Network Slice tag for operational management";
      }
      leaf ns-topology {
        type identityref {
          base network-slice-topology;
        }
        default "any-to-any";
        description
```

```
          "Network Slice topology.";
      }
      choice ns-slo-policy {
        description
          "Choice for SLO policy template.
           Can be standard template or customized template.";
        case standard {
          description
            "Standard SLO template.";
          leaf slo-template {
            type leafref {
              path "/ietf-network-slices"
                 + "/ns-templates/slo-template/id";
            }
            description
              "Standard SLO template to be used.";
          }
        }
        case custom {
          description
            "Customized SLO template.";
          container slo-policy {
            description
              "Contains the SLO policy.";
            leaf policy-description {
              type string;
              description
                "Description of the SLO policy.";
            }
            uses network-slice-metric-bounds;
          }
        }
      }
      uses status-params;
      list ns-endpoint {
        key "ep-id";
        uses endpoint;
        description
          "list of endpoints in this slice";
      }
      list ns-member {
        key "ns-member-id";
        description
          "List of network-slice-member in a slice";
        uses network-slice-member;
      }
    }
    //ietf-network-slice list
```

```
  }
}
<CODE ENDS>
```

9. Security Considerations

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

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

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

o /ietf-network-slice/ietf-network-slices/ietf-network-slice

The entries in the list above include the whole network configurations corresponding with the slice which the higher management system requests, and indirectly create or modify the PE or P device configurations. Unexpected changes to these entries could lead to service disruption and/or network misbehavior.

10. 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-slice
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

```
            Name: ietf-network-slice
            Namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice
            Prefix: ietf-ns
            Reference: RFC XXXX
```

11. Acknowledgments

The authors wish to thank Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many other NS DT members for their helpful comments and suggestions.

12. References

12.1. Normative References

[I-D.ietf-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-ietf-teas-ietf-network-slice-definition-00 (work in progress), January 2021.

[I-D.nsdt-teas-ns-framework]
Gray, E. and J. Drake, "Framework for Transport Network Slices", draft-nsdt-teas-ns-framework-04 (work in progress), July 2020.

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

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

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

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

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

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

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

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

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

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

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

[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>.

[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>.

12.2. Informative References

[I-D.geng-teas-network-slice-mapping]
Geng, X., Dong, J., Pang, R., Han, L., Niwa, T., Jin, J., Liu, C., and N. Nageshar, "5G End-to-end Network Slice Mapping from the view of Transport Network", draft-geng-teas-network-slice-mapping-02 (work in progress), July 2020.

[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-10 (work in progress), November 2020.

[I-D.liu-teas-transport-network-slice-yang]
Liu, X., Tantsura, J., Bryskin, I., Contreras, L., WU, Q., Belotti, S., and R. Rokui, "IETF Network Slice YANG Data Model", draft-liu-teas-transport-network-slice-yang-02 (work in progress), November 2020.

[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>.

Appendix A. IETF Network Slice NBI Model Usage Example

The following example describes a simplified service configuration of two IETF Network slice instances:

- IETF Network Slice 1 on Device1, Device3, and Device4, with any-to-any connection type

- IETF Network Slice 2 on Device2, Device3, with any-to-any connection type

```
     192.0.2.2 VLAN1
+--------+
|Device1 o------/
+--------+      |       +------+
+--------+      +------o|  A   +---------------+
|Device2 o-------/-----o|      |               |
+--------+              +---+--+               |
      198.51.100.2          |                  |
       VLAN2                |              +---+--+      192.0.2.4 VLAN1
                            |              |      |           +--------+
      192.0.2.3 VLAN1       |              |   C  o-----/-----oDevice4 |
+--------+                  |              +---+--+           +--------+
|        o------/           |                  |
|        |      |       +---+--+               |
| Device3|      +------o|  B   +---------------+
|        o-------/-----o|      |
+--------+              +------+
      198.51.100.3
       VLAN2

      POST: /restconf/data/ietf-network-slice:ietf-network-slices
      Host: example.com
      Content-Type: application/yang-data+json

   {
     "ietf-network-slices": {
       "ietf-network-slice": [
         {
           "network-slice-id": 1,
           "network-slice-name": "slice1",
           "network-slice-topology": "any-to-any",
           "network-slice-endpoint": [
             {
              "endpoint-id": 11,
              "endpoint-name": "device1-ep1",
              "endpoint-role": "any-to-any-role",
              "network-slice-match-criteria": [
               {
                 "match-type": "network-slice-vlan-match",
                 "value": "1"
               }
              ]
            },
            {
             "endpoint-id": 12,
             "endpoint-name": "device3-ep1",
             "endpoint-role": "any-to-any-role",
             "network-slice-match-criteria": [
               {
                 "match-type": "network-slice-vlan-match",
                 "value": "1"
               }
              ]
            },
            {
              "endpoint-id": 13,
              "endpoint-name": "device4-ep1",
              "endpoint-role": "any-to-any-role",
              "network-slice-match-criteria": [
                {
                  "match-type": "network-slice-vlan-match",
                  "value": "1"
                }
              ]
            }
          ]
        },
       {
         "network-slice-id": 2,
         "network-slice-name": "slice2",
         "network-slice-topology": "any-to-any",
         "network-slice-endpoint": [
           {
             "endpoint-id": 21,
             "endpoint-name": "device2-ep1",
             "endpoint-role": "any-to-any-role",
             "network-slice-match-criteria": [
               {
                 "match-type": "network-slice-vlan-match",
                 "value": "2"
               }
             ]
           },
          {
            "endpoint-id": 22,
            "endpoint-name": "device3-ep2",
            "endpoint-role": "any-to-any-role",
            "network-slice-match-criteria": [
              {
                "match-type": "network-slice-vlan-match",
                "value": "2"
              }
             ]
           }
         ]
        }
      ]
     }
   }
```

Appendix B. Comparison with Other Possible Design choices for IETF Network Slice NBI

According to the 3.3.1. Northbound Inteface (NBI) [I-D.nsdt-teas-ns-framework], the IETF Network Slice NBI is a technology-agnostic interface, which is used for a consumer to express requirements for a particular IETF Network Slice. Consumers operate on abstract IETF Network Slices, with details related to their realization hidden. As classified by [RFC8309], the IETF Network Slice NBI is classified as Customer Service Model.

This draft analyzes the following existing IETF models to identify the gap between the IETF Network Slice NBI requirements.

B.1. ACTN VN Model Augmentation

The difference between the ACTN VN model and the IETF Network Slice NBI requirements is that the IETF Network Slice NBI is a technology-agnostic interface, whereas the VN model is bound to the IETF TE Topologies. The realization of the IETF Network Slice does not necessarily require the slice network to support the TE technology.

The ACTN VN (Virtual Network) model introduced in [I-D.ietf-teas-actn-vn-yang] is the abstract consumer view of the TE network. Its YANG structure includes four components:

- VN: A Virtual Network (VN) is a network provided by a service provider to a customer for use and two types of VN has defined. The Type 1 VN can be seen as a set of edge-to-edge abstract links. Each link is an abstraction of the underlying network which can encompass edge points of the customer's network, access links, intra-domain paths, and inter-domain links.

- AP: An AP is a logical identifier used to identify the access link which is shared between the customer and the IETF scoped Network.

- VN-AP: A VN-AP is a logical binding between an AP and a given VN.

- VN-member: A VN-member is an abstract edge-to-edge link between any two APs or VN-APs. Each link is formed as an E2E tunnel across the underlying networks.

The Type 1 VN can be used to describe IETF Network Slice connection requirements. However, the Network Slice SLO and Network Slice Endpoint are not clearly defined and there's no direct equivalent. For example, the SLO requirement of the VN is defined through the IETF TE Topologies YANG model, but the TE Topologies model is related to a specific implementation technology. Also, VN-AP does not define "network-slice-match-criteria" to specify a specific NSE belonging to an IETF Network Slice.

B.2. RFC8345 Augmentation Model

The difference between the IETF Network Slice NBI requirements and the IETF basic network model is that the IETF Network Slice NBI requests abstract consumer IETF Network Slices, with details related to the slice Network hidden. But the IETF network model is used to describe the interconnection details of a Network. The customer service model does not need to provide details on the Network.

For example, IETF Network Topologies YANG data model extension introduced in Transport Network Slice YANG Data Model

[I-D.liu-teas-transport-network-slice-yang] includes three major parts:

- o Network: a transport network list and an list of nodes contained in the network
- o Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
- o Support network: vertical layering relationships between IETF Network Slice networks and underlay networks

Based on this structure, the IETF Network Slice-specific SLO attributes nodes are augmented on the Network Topologies model,, e.g. isolation etc. However, this modeling design requires the slice network to expose a lot of details of the network, such as the actual topology including nodes interconnection and different network layers interconnection.

Appendix C. Appendix B IETF Network Slice Match Criteria

5G is a use case of the IETF Network Slice and 5G End-to-end Network Slice Mapping from the view of IETF Network [I-D.geng-teas-network-slice-mapping]

defines two types of Network Slice interconnection and differentiation methods: by physical interface or by TNSII (Transport Network Slice Interworking Identifier). TNSII is a field in the packet header when different 5G wireless network slices are transported through a single physical interfaces of the IETF scoped Network. In the 5G scenario, "network-slice-match-criteria" refers to TNSII.

```
       +------------------------------------------------------------+
       |          5G E2E network slice orchestrator                 |
       ++-----------------------------------------------------+-----+
        |                        |                            |
        |                IETF Network Slice NBI               |
    +---+-------+                  |                    +-----+-----+
    |           |          +------------------+         |           |
    |RAN Slice  |          |IETF Network Slice|         |Core Slice |
    |controller |          |    controller    |         | controller|
    +----+------+          +-------+----------+         +-----+-----+
         |                         |                          |
         |                         |                          |
     +---+--+         +------------+----------------+        ++-----+
     |      |         |                             |        |      |
     |      |         |                             |        |      |
     |+----+|         |                             |        |      |
     ||    ||NS1-NSE1 |         Network Slice 1     |        |+----+|
     ||gNB1|+---------+-----+-----------------------+--------+|UPF1||
     ||    |+************  /                        |NS1-NSE3|+----+|
     |+----+|NS2-NSE1 |  */                         |        |      |
     |      |         |  /*                         |        |      |
     |+----+|NS1-NSE2 | /  *                        |        |      |
     ||    |+----------     *   Network Slice 2     |NS2-NSE3|+----+|
     ||gNB2|+************************************************+|UPF2||
     ||    ||NS2-NSE2 |                             |        |+----+|
     |+----+|         |                             |               |
     |      |         |                             |        |      |
     |      |         |                             |        |      |
     +------+         +----------- -----------------+        +------+
```

As shown in the figure, gNodeB 1 and gNodeB 2 use IP gNB1 and IP gNB2 to communicate with the IETF network, respectively. In addition, the traffic of NS1 and NS2 on gNodeB 1 and gNodeB 2 is transmitted through the same access links to the IETF slice network. The IETF slice network need to to distinguish different IETF Network Slice traffic of same gNB. Therefore, in addition to using "node-id" and "port-id" to identify a Network Slice Endpont, other information is needed along with these parameters to uniquely distinguish a NSE. For example, VLAN IDs in the user traffic can be used to distinguish the NSEs of gNBs and UPFs.

Authors’ Addresses

Bo Wu
Huawei Technologies
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: lana.wubo@huawei.com

Dhruv Dhody
Huawei Technologies
Divyashree Techno Park
Bangalore, Karnataka 560066
India

Email: dhruv.ietf@gmail.com

Liuyan Han
China Mobile

Email: hanliuyan@chinamobile.com

Reza Rokui
Nokia

Email: reza.rokui@nokia.com