Instantiation of IETF Network Slices in Service Providers Networks
draft-barguil-teas-network-slices-instantation-03

Abstract

Network Slicing (NS) is an integral part of Service Provider networks. The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture and YANG-based service models for network slice (NS) instantiation.

This document describes the relationship between IETF Network Slice models for requesting the IETF Network Slices and (e.g., Layer-3 Service Model, Layer-2 Service Model) and Network Models (e.g., Layer-3 Network Model, Layer-2 Network Model) used during their realizations. In addition, this document describes the communication between the IETF Network Slice Controller and the network controllers for the realization of IETF network slices.

The IETF Network Slice YANG model provides the customer-oriented view of the network slice. Thus, once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controllers. The network models are analyzed to satisfy the IETF Network Slice requirements, and the gaps in existing models are reported.

The document also provides operational and security considerations when deploying network slices in Service Provider networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
1. Introduction .................................................. 3
1.1. Terminology ............................................... 3
2. Reference Architecture and Components ......................... 4
2.1. Possible architectural options for IETF Network Slice Controller .................................................. 4
2.2. Possible relationship of IETF Network Slice service model with other models ........................................ 7
3. IETF Network Slice Requirements and Data Models ............... 8
4. IETF Network Slice Procedure .................................. 9
5. Network Controller Operation .................................. 10
5.1. LxVPN Service Models ...................................... 10
5.2. LxVPN Network Models ..................................... 11
5.3. Traffic Engineering Models ................................ 11
5.4. Traffic Engineering Service Mapping ........................ 11
6. Operational Considerations .................................... 11
6.1. Availability ............................................... 12
6.2. Downlink throughput / Uplink throughput .................... 12
6.3. Protection scheme ......................................... 12
6.4. Delay ................................................... 13
6.5. Packet loss rate ......................................... 13
1. Introduction

The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture.

The IETF Network Slice YANG service model provides the customer-oriented view of the network slice. Once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controller.

Several Service Models and Network Models, including Layer-3 Service Model (L3SM), Layer-2 Service Model (L2SM) and Network Models which may be utilized for IETF Network Slicing, are analyzed and identified gaps on existing models are reported.

This document describes the architecture and communication process between the Network Slice Controller and a network controller for IETF network slice creation.

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, as being defined in [I-D.ietf-teas-ietf-network-slices].

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 and Components

As described in [I-D.ietf-teas-ietf-network-slices], the IETF Network Slice Controller (NSC) is a functional entity for control and management of IETF network slices. As shown in Figure A, NSC from its Northbound Interface (NBI) exposes set of APIs that allow a higher level system to request an IETF network slice. The NSC NBI supports the request for enabling of an IETF Network Slice (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 Southbound Interface (SBI) with one or more Network Controllers for the realization of the requested IETF Network Slice.

This document focuses on how IETF Network Slice Controller (NSC) can be implemented in the operator’s network.

```
+------------------------------------------+  
|         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 Network Slice Controller as a module of the Hierarchical SDN controller.

2.1. Possible architectural options for IETF Network Slice Controller

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-slices] includes a three-level hierarchy and expresses how each level relates with the ACTN architecture framework.

Figure 2 defines depicts a possible architecture using those concepts. It starts from a top consumer or high-level operational systems. Next, the IETF Network Slice Controller function might be
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.

Figure 2 IETF Network Slice Controller as a module of the Hierarchical SDN controller.

In other implementations, the IETF Network Slice Controller 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.
Figure 3 The IETF Network Slice Controller as a stand-alone entity.

As another implementation possibility, the IETF 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.

Figure 4 IETF Network Slice Controller as a module of the Network controller.
2.2. Possible relationship of IETF Network Slice service model with other models

IETF Network Slice service is expected to serve as input from where deriving some other models in the network. According to the architectural options before, different relationships could be considered. Figure 5 reflects a couple of options.

Thus, the IETF Network Slice model (e.g., as defined in [RefNBIdraft]) could feed existing service models, such as L2SM or L3SM, or could feed existing network models (e.g., EVPN, L3VPN, etc). Existing models both for service or network level could require some
extensions themselves, or their application in conjunction with some other complementary models (e.g., TE model) to accomplish the service objectives and expectations as declared in the IETF Network Slice model.

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 below the slice use cases reported:

<table>
<thead>
<tr>
<th>Network Slice Requirements for 5G service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Availability</strong></td>
</tr>
<tr>
<td>Deterministic communication</td>
</tr>
<tr>
<td>Downlink throughput per network slice</td>
</tr>
<tr>
<td>Energy efficiency</td>
</tr>
<tr>
<td>Group communication support</td>
</tr>
<tr>
<td>Isolation level</td>
</tr>
<tr>
<td>Maximum supported packet size</td>
</tr>
<tr>
<td>Mission critical support</td>
</tr>
<tr>
<td>Performance monitoring</td>
</tr>
<tr>
<td>Slice quality of service parameters</td>
</tr>
<tr>
<td>Support for non-IP traffic</td>
</tr>
<tr>
<td>Uplink throughput per network slice</td>
</tr>
<tr>
<td>User data access</td>
</tr>
<tr>
<td>Delay tolerance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NFV-based services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming and outgoing bandwidth</td>
</tr>
<tr>
<td>Qos metrics</td>
</tr>
<tr>
<td>Directionality</td>
</tr>
<tr>
<td>MTU</td>
</tr>
<tr>
<td>Protection scheme</td>
</tr>
<tr>
<td>Connectivity mode</td>
</tr>
</tbody>
</table>
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]: A Yang Data Model for IETF Network Slice NBI.

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

4. IETF Network Slice Procedure

An IETF Network Slice may use several underlying technologies. The creation of a new IETF Network Slice will be initiated with following three steps:

1. A higher level system requests connections with specific characteristics via the NBI.

2. This request will be processed by an IETF NSC which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.

3. A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.
5. Network Controller Operation

As a functional entity responsible for managing a network domain, the network controller, can expose its northbound interface based on YANG models. The IETF Network Slice Controller can use the network controller’s NBI during the realization of IETF Network Slice. The following network models can be used for realization of IETF Network slices:

* **LxVPN Network models:**
  - These models describe a VPN service from the network point of view. It supports the creation of Layer 3 and Layer 2 services using several control planes.

* **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.

5.1. LxVPN Service 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 Service models could be a realization of IETF Network slice requests.

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

* [RFC8049]: YANG Data Model for L3VPN Service Delivery
5.2. LxVPN Network Models

Similar to the Service Models, the framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The 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

5.3. 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.

5.4. 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.

6. Operational Considerations

This section outlines the compliance and operational aspects of Network Controller models with IETF Network slice requirements. Section 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.
6.1. Availability

As per [draft-ietf-teas-te-service-mapping-yang], 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%).

6.2. Downlink throughput / Uplink throughput.

The LxVPN Models ([I-D.ietf-opsawg-l3sm-l3nm] and [I-D.ietf-opsawg-l2nm]) allow to specify the bandwidth at the interface level between the slice and the customer. In addition, the Service Mapping model [draft-ietf-teas-te-service-mapping-yang] allows to bind a VPN to a given LSP, which have its bandwidth requirements. Additionally, TE models can force a give bandwidth in the connection between Provider Edges.

Previous comment applies to the incoming and outgoing bandwidth parameters required for the NFV-based services use case in [I-D.contreras-teas-slice-nbi]. The Network sharing use case has Maximum and Guaranteed Bit Rate parameters. These parameters can be mapped to the TE tunnel models when setting up LSPs [draft-ietf-teas-yang-te].

6.3. Protection scheme

Protection schemes are mechanisms to define how to setup resources for a given connection. TE tunnel models [draft-ietf-teas-yang-te] includes protection and restoration as two main attributes. The parameters included in the containers for protection and restoration cover the requirements of the IETF NS related with protection schemes. Similarly, TE models cover the parameter ‘recovery time’ for the network sharing use case.
6.4. Delay

Delay is a critical parameter for several IETF NS types. Every use-case defined in [I-D.contreras-teas-slice-nbi] contains delay constraints. 5G use cases require 'delay tolerance', NFV-based services have the delay information within 'QoS metrics' and 'Bounded latency' in the network sharing use case.

During the realization of the IETF Network Slice, these parameters are part of the requirements of a TE tunnel configuration [draft-ietf-teas-yang-te]. They can be included within the 'path-metric-bounds' parameter, so the created LSP fulfils the given metrics bounds like 'path-metric-delay-average' or 'path-metric-delay-minimum'.

6.5. Packet loss rate

The packet loss rate indicates the maximum rate for lost packets that the service tolerates in the link. During the realization of the IETF Network Slice, this attribute will influence the tunnel selection and the value is included in the [draft-ietf-teas-yang-te] document as the 'path-metric-loss'. The 'path-metric-loss' is a metric type, which measures the percentage of packet loss of all links traversed by a P2P path. This parameter is required for 5G services and network sharing use-case, while it is part of the 'QoS metrics' for the NFV-based services.

7. Network Slice Procedure

Draft [draft-contreras-teas-slice-controller-models] shows the internal structure of an IETF Network Slice Controller which can be divided into two components:

* 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.
The details of IETF network slice mapper and realize are provided below for various implementation of NCS.

7.1. IETF Network Slice requested to Hierarchical Network Controller

Referring to Figure 1 in an integrated architecture, the IETF Network Slice Controller (NCS) is part of a Hierarchical SDN controller module, the NSC’s and the Hierarchical Network Controller should share the same internal data and the same NBI. Thus, the H-SDN module must be able to:

* Map: The customer request received using the [draft-wd-teas-ietf-network-slice-nbi-yang] 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 IETF Network Slice ID ns-id used of the [draft-wd-teas-ietf-network-slice-nbi-yang] must be directly mapped to the transport-instance-id at the VPN-Node level.

```
+-------------------+------------------+
|                  |                  |
|                  |                  |
|                  |                  |
+-------------------v------------------+

+-------------------+------------------+
|                  |                  |
|                  |                  |
|                  |                  |
+-------------------v------------------+

IETF Network Slice Request:
draft-wd-teas-ietf-network-slice-nbi-yang
* network-slice-id

Hierarchical Network Controller/Orchestrator

IETF Network Slice Controller

IETF Network Slice Realizer: LXNM

VPN-id
* transport-instance-id

Network Controller

Network Elements

Network Elements

Figure 9 Workflow for the slice request in an integrated architecture.
7.2. IETF Network Slice requested to Network Slice Controller

Referring to Figure 2 when the Network Slice Controller is a stand-alone controller module, the NSC’s should perform the same two tasks described in section 6.1:

* Map: Process the customer request. The customer request can be sent using the [draft-liu-teas-transport-network-slice-yang]. 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.

```
+-----------------------+-----------------------+
| IETF Network Slice Request |
| draft-liu-teas-transport-network-slice-yang |
| network-id |
+-----------------------+-----------------------+

```

```
<table>
<thead>
<tr>
<th>IETF Network Slice Controller</th>
</tr>
</thead>
</table>
+-----------------------+-----------------------+
| IETF Network Slice Realizer: LXNM |
| VPN-id |
| * Underlay-transport |
| * transport-instance-id |
+-----------------------+-----------------------+

```

```
| Network Controller |
+-------------------+-------------------+

```

```
| Network Elements |
|------------------|------------------|

```

Figure 10 Workflow for the slice request in an stand-alone architecture.
7.3. Network Slice Controller as part of the domain controller

The Network Slice Controller can be a module of the Network controller. In that case, two options are available. One is to share the same device data model in the NBI and SBI of the SDN controller. The direct translation would reduce the service logic implemented at the SDN controller level, grouping the mapping and translation into a single task:

* Realize: As the device models are part of the network controller’s NBI thus, the realization can be done by the network controller applying a simple service logic to send the Network elements.

![Diagram]

Figure 11 Workflow for the slice request in an stand-alone architecture.

A second option introduces a more complex logic in the network controller and creates an abstraction layer to process the transport slices. In that case, the controller should receive network slices creation requests and maintain the whole set of implemented slices:

* Map & Realize: The mapping and realization can be done by the Domain controller applying the service logic to create policies directly on the Network elements.
Figure 12 Workflow for the slice request in an stand-alone architecture.

8. 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 clearly 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 that implements an IETF network slice

* Malicious clients attempting to leak traffic of the slice.

9. IANA Considerations

This document is informational and does not require IANA allocations.

10. 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 available. 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.

11. Contributors

Daniel King:daniel@olddog.co.uk>
12. Acknowledgements

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

13. Normative References

[I-D.contreras-teas-slice-nbi]

[I-D.ietf-opsawg-l2nm]

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

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

[I-D.ietf-teas-te-service-mapping-yang]
[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I.,
and O. G. D. Dios, "A YANG Data Model for Traffic
Engineering Tunnels, Label Switched Paths and Interfaces",
Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-
29, 7 February 2022,
<https://datatracker.ietf.org/doc/html/draft-ietf-teas-
yang-te-29>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,

and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,

Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,

Access Control Model", STD 91, RFC 8341,
DOI 10.17487/RFC8341, March 2019,

Abstraction and Control of TE Networks (ACTN)", RFC 8453,
DOI 10.17487/RFC8453, August 2018,

Data Model for Layer 2 Virtual Private Network (L2VPN)
Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October

L. Geng, "A Framework for Automating Service and Network
Management with YANG", RFC 8969, DOI 10.17487/RFC8969,
Annex. Example of relationship between IETF NBI model parameters and L3SM model parameters

This annex presents an initial analysis of the relationship between IETF NBI model parameters and L3SM service model parameters.

The L3SM service parameters are defined in section 6.2 of RFC 8299. The following parameters are considered, so far:

* **Bandwidth.** This parameter indicates the bandwidth requirement between each CE and PE participating in the service, then referring essentially to the required WAN link bandwidth. It is expressed in terms of bits per second and individually specified for both input and output. Despite it is not stated in RFC 8299, this parameter can be interpreted as the CIR/PIR expected for the CE - PE connection.

* **MTU.** This parameter indicates the maximum PDU size expected for the layer-3 service. It is relevant since packets could be discarded in case the customer sends packets with longer MTU than the one expressed by this parameter.

* **QoS.** Regarding QoS, two different kind of parameters are detailed.
  - **QoS classification policy.** This policy is used to classify the traffic received from the customer, and it is expressed as a set of ordered rules. It is used for marking the input traffic (from CE to PE) when the customer flows match any of the rules in the list, setting the appropriate target class of service (target-class-id).
  - **QoS profile.** This profile defines the traffic-scheduling to be applied to the flows for either Site-to-WAN, WAN-to-Site, or both directions. It contains the following information per class of service: rate-limit, latency, jitter and guaranteed bandwidth.

* **Multicast.** This parameter identifies if the service is multicast, and if so, what is the role of the site in the customer multicast service topology (i.e., source, receiver, or both). It also defines the kind of multicast relationship with the customer (i.e., as a router requiring PIM, host requiring either IGMP or MLD, or both), as well as the support of IPv4, IPv6 or both.

On the other hand, the IETF NS NBI YANG model supports a number of SLOs and SLEs in the form of network slice service policy attributes. Such policy can apply to per-network slice, per-connection group or
per-connection individually (over-writing of attributes is allowed as more granular information is provided). The following SLO attributes are detailed:

* One-way / Two-way bandwidth, indicating the guaranteed minimum bandwidth between any two NSEs (unidirectional / bidirectional).

* One-way / Two-way latency, indicating the guaranteed minimum latency between any two NSEs (unidirectional / bidirectional).

* One-way / Two-way delay variation, indicating the maximum permissible delay variation of the slice (unidirectional / bidirectional).

* One-way / Two-way packet loss, indicating the maximum permissible packet loss rate between endpoints (unidirectional / bidirectional).

Additionally, the following SLEs are defined:

* MTU, referring to the the maximum PDU size that the customer may use.

* Security, indicating if encryption or other security measures are required between two endpoints.

* Isolation, as a way of indicating the isolation level expected by the customer in the allocation of network resources.

* Maximum occupancy level, to express the amount of flows to be admitted (and optionally a maximum number of countable resource units such as IP or MAC addresses).

Thus, an initial mapping between L3SM and IETF NS NBI model can be performed as indicated in the following table.
### Table 1: Mapping of IETF NS NBI and L3SM service attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3SM (RFC 8299)</td>
<td>IETF NSC NBI YANG model</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Sum of bandwidth SLO per NSE counting all connections</td>
</tr>
<tr>
<td>MTU</td>
<td>MTU attribute in SLE</td>
</tr>
<tr>
<td>QoS</td>
<td></td>
</tr>
<tr>
<td>- QoS classification policy</td>
<td>Defined in the model as network-access-qos-policy-name to be applied per access-point</td>
</tr>
<tr>
<td>- QoS profile</td>
<td></td>
</tr>
<tr>
<td>- rate-limit</td>
<td>Defined in the model as incoming/outgoing rate-limits per end-point (or access-point)</td>
</tr>
<tr>
<td>- latency</td>
<td>One-way / Two-way latency SLO</td>
</tr>
<tr>
<td>- jitter</td>
<td>One-way / Two-way delay variation SLO</td>
</tr>
<tr>
<td>- bandwidth</td>
<td>One-way / Two-way bandwidth SLO</td>
</tr>
<tr>
<td>Multicast</td>
<td>The need of replication can be inferred from ns-connectivity-type. Further details are not available (e.g. source or receiver role)</td>
</tr>
</tbody>
</table>

The following consideration can be made.

* While the QoS profile in L3SM applies per service class, the parameters in IETF NS NBI apply per connection. So if per-class granularity is required in an IETF network slice, then different connections have to be defined between the same end-points, one per service class.

* A number of attributes are not defined in L3SM such as packet loss, isolation or security. Then L3SM could not be sufficient to realize IETF network slices with such specific needs, unless those other objectives and expectations are provided by other means (e.g., realizing the L3SM thorough technologies guaranteeing dedicated resource allocation such as OTN).
Abstract

Network Slicing (NS) is an integral part of Service Provider networks. The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture and YANG-based service models for network slice (NS) instantiation.

This document describes the relationship between IETF Network Slice models for requesting the IETF Network Slices and (e.g., Layer-3 Service Model, Layer-2 Service Model) and Network Models (e.g., Layer-3 Network Model, Layer-2 Network Model) used during their realizations. In addition, this document describes the communication between the IETF Network Slice Controller and the network controllers for the realization of IETF network slices.

The IETF Network Slice YANG model provides the customer-oriented view of the network slice. Thus, once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controllers. The network models are analyzed to satisfy the IETF Network Slice requirements, and the gaps in existing models are reported.

The document also provides operational and security considerations when deploying network slices in Service Provider networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

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

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

Table of Contents

1. Introduction ............................................. 3
   1.1. Terminology ....................................... 3
2. Reference Architecture and Components ............................ 4
   2.1. Possible architectural options for IETF Network Slice
        Controller ....................................... 4
   2.2. Possible relationship of IETF Network Slice service model
        with other models .................................. 7
3. IETF Network Slice Requirements and Data Models ................. 8
4. IETF Network Slice Procedure .................................. 9
5. Network Controller Operation .................................. 10
   5.1. LxVPN Service Models ................................ 10
   5.2. LxVPN Network Models ................................ 11
   5.3. Traffic Engineering Models ......................... 11
   5.4. Traffic Engineering Service Mapping ................ 11
6. Operational Considerations .................................. 11
   6.1. Availability ..................................... 12
   6.2. Downlink throughput / Uplink throughput .............. 12
   6.3. Protection scheme .................................. 12
   6.4. Delay ............................................. 13
   6.5. Packet loss rate ................................... 13
1. Introduction

The IETF has produced several YANG data models to support the Software-Defined Networking and network slice architecture.

The IETF Network Slice YANG service model provides the customer-oriented view of the network slice. Once the IETF Network Slice controller (NSC) receives a request, it needs to map it to accomplish the specific parameters expected by the network controller.

Several Service Models and Network Models, including Layer-3 Service Model (L3SM), Layer-2 Service Model (L2SM) and Network Models which may be utilized for IETF Network Slicing, are analyzed can satisfy the IETF Network Slice requirements. In addition, identified gaps on existing models are reported.

This document describes the architecture and communication process between the Network Slice Controller and a network controller for IETF network slice creation.

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, as being defined in [I-D.ietf-teas-ietf-network-slices].

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 and Components

As described in [I-D.ietf-teas-ietf-network-slices], the IETF Network Slice Controller (NSC) is a functional entity for control and management of IETF network slices. As shown in Figure A, NSC from its Northbound Interface (NBI) exposes set of APIs that allow a higher level system to request an IETF network slice. The NSC NBI supports the request for enabling of an IETF Network Slice (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 Southbound Interface (SBI) with one or more Network Controllers for the realization of the requested IETF Network Slice.

This document focuses on how IETF Network Slice Controller (NSC) can be implemented in the operator’s network.

+------------------------------------------+
|         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 Network Slice Controller as a module of the Hierarchical SDN controller.

2.1. Possible architectural options for IETF Network Slice Controller

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-slices] includes a three-level hierarchy and expresses how each level relates with the ACTN architecture framework.

Figure 2 defines depicts a possible architecture using those concepts. It starts from a top consumer or high-level operational systems. Next, the IETF Network Slice Controller function might be
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.

![Diagram](image)

**Figure 2 IETF Network Slice Controller as a module of the Hierarchical SDN controller.**

In other implementations, the IETF Network Slice Controller 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.
As another implementation possibility, the IETF 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.
2.2. Possible relationship of IETF Network Slice service model with other models

IETF Network Slice service is expected to serve as input from where deriving some other models in the network. According to the architectural options before, different relationships could be considered. Figure 5 reflects a couple of options.

Thus, the IETF Network Slice model (e.g., as defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang]) could feed existing service models, such as L2SM or L3SM, or could feed existing network...
models (e.g., EVPN, L3VPN, etc). Existing models both for service or network level could require some extensions themselves, or their application in conjunction with some other complementary models (e.g., TE model) to accomplish the service objectives and expectations as declared in the IETF Network Slice model.

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 below the slice use cases reported:

<table>
<thead>
<tr>
<th>Network Slice Requirements for 5G service</th>
</tr>
</thead>
</table>
| 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  
  Delay tolerance |

<table>
<thead>
<tr>
<th>NFV-based services</th>
</tr>
</thead>
</table>
| Incoming and outgoing bandwidth  
  Qos metrics  
  Directionality  
  MTU  
  Protection scheme  
  Connectivity mode |
To accomplish those requirements, a set of YANG data models have been proposed. Those Yang models 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.

* [I-D.ietf-teas-ietf-network-slice-nbi-yang]: A Yang Data Model for IETF Network Slice NBI.
* [I-D.liu-teas-transport-network-slice-yang]: Transport Network Slice YANG Data Model.

4. IETF Network Slice Procedure

An IETF Network Slice may use several underlying technologies. The creation of a new IETF Network Slice will be initiated with following three steps:

1. A higher level system requests connections with specific characteristics via the NBI.

2. This request will be processed by an IETF NSC which specifies a mapping between northbound request to any IETF Services, Tunnels, and paths models.

3. A series of requests for creation of services, tunnels and paths will be sent to the network to realize the transport slice.
5. Network Controller Operation

As a functional entity responsible for managing a network domain, the network controller, can expose its northbound interface based on YANG models. The IETF Network Slice Controller can use the network controller’s NBI during the realization of IETF Network Slice. The following network models can be used for realization of IETF Network slices:

* LxVPN Network models:
  - These models describe a VPN service from the network point of view. It supports the creation of Layer 3 and Layer 2 services using several control planes.

* 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.

5.1. LxVPN Service 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 Service models could be a realization of IETF Network slice requests.

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

* [RFC8049]: YANG Data Model for L3VPN Service Delivery
5.2. LxVPN Network Models

Similar to the Service Models, the framework defined in [RFC8969] compiles a set of YANG data models for automating network services. The Network models could be reused for the realization of Network slice requests.

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

* [RFC9182]: A Layer 3 VPN Network YANG Model
* [I-D.ietf-opsawg-l2nm]: A Layer 2 VPN Network YANG Model

5.3. 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.

5.4. 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.

6. Operational Considerations

This section outlines the compliance and operational aspects of Network Controller models with IETF Network slice requirements. Section 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.
6.1. Availability

As per [draft-ietf-teas-te-service-mapping-yang], 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%).

6.2. Downlink throughput / Uplink throughput.

The LxVPN Models ([RFC9182] and [I-D.ietf-opsawg-l2nm]) allow to specify the bandwidth at the interface level between the slice and the customer. In addition, the Service Mapping model [draft-ietf-teas-te-service-mapping-yang] allows to bind a VPN to a given LSP, which have its bandwidth requirements. Additionally, TE models can force a give bandwidth in the connection between Provider Edges.

Previous comment applies to the incoming and outgoing bandwidth parameters required for the NFV-based services use case in [I-D.contreras-teas-slice-nbi]. The Network sharing use case has Maximum and Guaranteed Bit Rate parameters. These parameters can be mapped to the TE tunnel models when setting up LSPs [draft-ietf-teas-yang-te].

6.3. Protection scheme

Protection schemes are mechanisms to define how to setup resources for a given connection. TE tunnel models [draft-ietf-teas-yang-te] includes protection and restoration as two main attributes. The parameters included in the containers for protection and restoration cover the requirements of the IETF NS related with protection schemes. Similarly, TE models cover the parameter ‘recovery time’ for the network sharing use case.
6.4. Delay

Delay is a critical parameter for several IETF NS types. Every use-case defined in [I-D.contreras-teas-slice-nbi] contains delay constraints. 5G use cases require ‘delay tolerance’, NFV-based services have the delay information within ‘QoS metrics’ and ‘Bounded latency’ in the network sharing use case.

During the realization of the IETF Network Slice, these parameters are part of the requirements of a TE tunnel configuration [draft-ietf-teas-yang-te]. They can be included within the ‘path-metric-bounds’ parameter, so the created LSP fulfils the given metrics bounds like ‘path-metric-delay-average’ or ‘path-metric-delay-minimum’.

6.5. Packet loss rate

The packet loss rate indicates the maximum rate for lost packets that the service tolerates in the link. During the realization of the IETF Network Slice, this attribute will influence the tunnel selection and the value is included in the [draft-ietf-teas-yang-te] document as the ‘path-metric-loss”. The ‘path-metric-loss’ is a metric type, which measures the percentage of packet loss of all links traversed by a P2P path. This parameter is required for 5G services and network sharing use-case, while it is part of the ‘QoS metrics’ for the NFV-based services.

7. Relationship between IETF NBI model parameters and L3SM and L2SM model parameters

This section presents an initial analysis of the relationship between IETF NBI model parameters and L3SM and L2SM service model parameters.

The L3SM service parameters are defined in section 6.2 of RFC 8299. The following parameters are considered, so far:

* Bandwidth. This parameter indicates the bandwidth requirement between each CE and PE participating in the service, then referring essentially to the required WAN link bandwidth. It is expressed in terms of bits per second and individually specified for both input and output. Despite it is not stated in RFC 8299, this parameter can be interpreted as the CIR/PIR expected for the CE – PE connection.

* MTU. This parameter indicates the maximum PDU size expected for the layer-3 service. It is relevant since packets could be discarded in case the customer sends packets with longer MTU than the one expressed by this parameter.
* QoS. Regarding QoS, two different kind of parameters are detailed.

- QoS classification policy. This policy is used to classify the traffic received from the customer, and it is expressed as a set of ordered rules. It is used for marking the input traffic (from CE to PE) when the customer flows match any of the rules in the list, setting the appropriate target class of service (target-class-id).

- QoS profile. This profile defines the traffic-scheduling to be applied to the flows for either Site-to-WAN, WAN-to-Site, or both directions. It contains the following information per class of service: rate-limit, latency, jitter and guaranteed bandwidth.

* Multicast. This parameter identifies if the service is multicast, and if so, what is the role of the site in the customer multicast service topology (i.e., source, receiver, or both). It also defines the kind of multicast relationship with the customer (i.e., as a router requiring PIM, host requiring either IGMP or MLD, or both), as well as the support of IPv4, IPv6 or both.

Similarly L2SM model parameters are described in section 5.9 and 5.10 of RFC 8466.

* Bandwidth. This parameter is related to the bandwidth between both CE and PE and can be expressed as CIR/EIR/PIR, in the ingress or egress direction, taking the CE as the point of reference.

* MTU. This parameter refers to the maximum layer-2 PDU frame size.

* QoS. The specification of the QoS follows a similar structure to the one described in the case of L3SM. Some differences apply, for instance, at the time of QoS classification, which is performed on top of layer-2 parameters (e.g., MAC addresses).

* BUM traffic. This parameter allows to determine if a site acts as source, receiver, or both.

* Availability. This parameter in the L2SM model relates to the capability of supporting multi-homing.
On the other hand, the IETF NS NBI YANG model supports a number of SLOs and SLEs in the form of network slice service policy attributes. Such policy can apply to per-network slice, per-connection group or per-connection individually (over-writing of attributes is allowed as more granular information is provided). The following SLO attributes are detailed:

* One-way / Two-way bandwidth, indicating the guaranteed minimum bandwidth between any two NSEs (unidirectional / bidirectional).

* One-way / Two-way latency, indicating the guaranteed minimum latency between any two NSEs (unidirectional / bidirectional).

* One-way / Two-way delay variation, indicating the maximum permissible delay variation of the slice (unidirectional / bidirectional).

* One-way / Two-way packet loss, indicating the maximum permissible packet loss rate between endpoints (unidirectional / bidirectional).

Additionally, the following SLEs are defined:

* MTU, referring to the the maximum PDU size that the customer may use.

* Security, indicating if encryption or other security measures are required between two endpoints.

* Isolation, as a way of indicating the isolation level expected by the customer in the allocation of network resources.

* Maximum occupancy level, to express the amount of flows to be admitted (and optionally a maximum number of countable resource units such as IP or MAC addresses).

Thus, an initial mapping between L3SM, L2SM and IETF NS NBI model can be performed as indicated in the following table.
<table>
<thead>
<tr>
<th>L3SM (RFC 8299)</th>
<th>L2SM (RFC 8466)</th>
<th>IETF NSC NBI YANG model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>Bandwidth (CIR, PIR)</td>
<td>Sum of bandwidth SLO per NS counting all connections</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTU (later 3 service)</td>
<td>MTU (later 2 service)</td>
<td>MTU attribute in SLE</td>
</tr>
<tr>
<td>QoS</td>
<td>QoS</td>
<td>QoS</td>
</tr>
<tr>
<td>QoS classification</td>
<td>QoS classification</td>
<td>Defined in the model as network-access-qos-policy-name to be applied per access-point int</td>
</tr>
<tr>
<td>QoS profile</td>
<td>QoS profile</td>
<td></td>
</tr>
<tr>
<td>- rate-limit</td>
<td>- rate-limit</td>
<td>Defined in the model as incoming/outgoing rate-limit int</td>
</tr>
<tr>
<td>latency</td>
<td>latency</td>
<td>One-way / Two-way latency SLO</td>
</tr>
<tr>
<td>jitter</td>
<td>jitter</td>
<td>One-way / Two-way delay</td>
</tr>
<tr>
<td>bandwidth</td>
<td>bandwidth</td>
<td>One-way / Two-way bandwidth SLO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast</td>
<td>Broadcast, Unknown,</td>
<td>The need of replication can be inferred from Unicast and Multicast (BUM) ns-connectivity-type. Further details are not available (source or receiver role)</td>
</tr>
</tbody>
</table>
Table 1 Mapping of IETF NS NBI and LxSM service attributes.

The following consideration can be made.

* While the QoS profile in L3SM and L2SM applies per service class, the parameters in IETF NS NBI apply per connection. So if per-class granularity is required in an IETF network slice, then different connections have to be defined between the same endpoints, one per service class.
* A number of attributes are not defined in L3SM nor L2SM such as packet loss, isolation or security. Then L3SM and L2SM could not be sufficient to realize IETF network slices with such specific needs, unless those other objectives and expectations are provided by other means (e.g., realizing the L3SM through technologies guanareling dedicated resource allocation such as OTN).

8. Network Slice Procedure

Draft [draft-contreras-teas-slice-controller-models] shows the internal structure of an IETF Network Slice Controller which can be divided into two components:

* 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 8: IETF Network Slice Controller Structure
The details of IETF network slice mapper and realize are provided below for various implementation of NCS.

8.1. IETF Network Slice requested to Hierarchical Network Controller

Referring to Figure 1 in an integrated architecture, the IETF Network Slice Controller (NCS) is part of a Hierarchical SDN controller module, the NSC’s and the Hierarchical Network Controller should share the same internal data and the same NBI. Thus, the H-SDN module must be able to:

* Map: The customer request received using the [draft-wd-teas-ietf-network-slice-nbi-yang] 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 IETF Network Slice ID ns-id used of the [draft-wd-teas-ietf-network-slice-nbi-yang] must be directly mapped to the transport-instance-id at the VPN-Node level.
IETF Network Slice Request:
draft-wd-teas-ietf-network-slice-nbi-yang
* network-slice-id

Hierarchical Network Controller/Orchestrator

IETF Network Slice Realizer: LXNM
VPN-id
* transport-instance-id

Network Controller

Network Elements

Figure 9 Workflow for the slice request in an integrated architecture.

8.2. IETF Network Slice requested to Network Slice Controller

Referring to Figure 2 when the Network Slice Controller is a stand-alone controller module, the NSC’s should perform the same two tasks described in section 6.1:

* Map: Process the customer request. The customer request can be sent using the [draft-liu-teas-transport-network-slice-yang]. 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.

![Diagram of network slice request workflow]

**Figure 10** Workflow for the slice request in an stand-alone architecture.

### 8.3. Network Slice Controller as part of the domain controller

The Network Slice Controller can be a module of the Network controller. In that case, two options are available. One is to share the same device data model in the NBI and SBI of the SDN controller. The direct translation would reduce the service logic implemented at the SDN controller level, grouping the mapping and translation into a single task:

* Realize: As the device models are part of the network controller’s NBI thus, the realization can be done by the network controller applying a simple service logic to send the Network elements.
A second option introduces a more complex logic in the network controller and creates an abstraction layer to process the transport slices. In that case, the controller should receive network slices creation requests and maintain the whole set of implemented slices:

* Map & Realize: The mapping and realization can be done by the Domain controller applying the service logic to create policies directly on the Network elements.
9. 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 clearly 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 [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 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.

10. IANA Considerations

This document is informational and does not require IANA allocations.

11. 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 available. 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.

12. Contributors

Daniel King, daniel@olddog.co.uk

13. Acknowledgements

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).
14. Normative References

[I-D.contreras-teas-slice-nbi]
Contreras, L. M., Homma, S., Ordonez-Lucena, J. A.,
Tantsura, J., and H. Nishihara, "IETF Network Slice Use
Cases and Attributes for Northbound Interface of IETF
Network Slice Controllers", Work in Progress, Internet-
Draft, draft-contreras-teas-slice-nbi-06, 7 March 2022,
<https://datatracker.ietf.org/doc/html/draft-contreras-
teas-slice-nbi-06>.

[I-D.ietf-opsawg-l2nm]
Boucadair, M., Dios, O. G. D., Barguil, S., and L. A.
Munoz, "A YANG Network Data Model for Layer 2 VPNS", Work
in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-19, 2
June 2022, <https://datatracker.ietf.org/doc/html/draft-
ietf-opsawg-l2nm-19>.

[I-D.ietf-teas-ietf-network-slices]
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhihani,
K., Contreras, L. M., and J. Tantsura, "Framework for IETF
Network Slices", Work in Progress, Internet-Draft, draft-
ietf-teas-ietf-network-slices-12, 30 June 2022,
<https://datatracker.ietf.org/doc/html/draft-ietf-
teas-ietf-network-slices-12>.

[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 Data Model", Work in Progress, Internet-
Draft, draft-ietf-teas-te-service-mapping-yang-11, 11 July
2022, <https://datatracker.ietf.org/doc/html/draft-ietf-
teas-te-service-mapping-yang-11>.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I.,
and O. G. D. Dios, "A YANG Data Model for Traffic
Engineering Tunnels, Label Switched Paths and Interfaces",
Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-
29, 7 February 2022,
<https://datatracker.ietf.org/doc/html/draft-ietf-teas-
yang-te-29>.
[I-D.ietf-teas-ietf-network-slice-nbi-yang]

[I-D.liu-teas-transport-network-slice-yang]


Authors’ Addresses

Samier Barguil
Telefonica
Distrito T
28050 Madrid
Spain
Email: samier.barguilgiraldo.ext@telefonica.com

Luis M. Contreras
Telefonica
Distrito T
28050 Madrid
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

Victor Lopez
Nokia
Calle de María Tubau, 9
28050 Madrid
Spain
Email: victor.lopez@nokia.com

Reza Rokui
Ciena
Canada
Email: rrokui@ciena.com

Oscar Gonzalez de Dios
Telefonica
Distrito T
28050 Madrid
Spain
Email: oscar.gonzalezdedios@telefonica.com

[Page 26]
Realizing Network Slices in IP/MPLS Networks

draft-bestbar-teas-ns-packet-08

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 co-exist on the same network while ensuring 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 domains and nodes to provide forwarding treatment (scheduling and drop policy) on to packets that carry the respective Diffserv code point. This document adopts a similar approach to Diffserv and proposes a scalable approach to realize network slicing in IP/MPLS networks. The solution does not mandate Diffserv to be enabled in the network to provide a specific forwarding treatment, but can co-exist with and complement it when enabled.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 6 August 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction ................................................. 3
1.1. Terminology ............................................. 5
1.2. Acronyms and Abbreviations .............................. 6
2. Network Resource Slicing Membership ......................... 7
3. IETF Network Slice Realization ............................... 8
3.1. Network Topology Filters ................................. 9
3.2. IETF Network Slice Service Request ..................... 9
3.3. Slice-Flow Aggregation Mapping .......................... 10
3.4. Path Placement over NRP Topology ....................... 10
3.5. NRP Policy Installation ................................. 10
3.6. Path Instantiation ..................................... 11
3.7. Service Mapping ....................................... 11
3.8. Network Slice-Flow Aggregate Relationships ............ 11
4. Network Resource Partition Modes ............................ 12
4.1. Data plane Network Resource Partition Mode ............. 12
4.2. Control Plane Network Resource Partition Mode .......... 13
4.3. Data and Control Plane Network Resource Partition Mode 15
5. Network Resource Partition Instantiation .................. 15
5.1. NRP Policy Definition ................................. 15
5.1.1. Network Resource Partition Data Plane Selector .... 16
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 enhance the delivery of 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 (e.g., RSVP, SR, or other) agnostic solution that a Service Provider can deploy to realize network slicing in IP/MPLS networks.

[I-D.ietf-teas-ietf-network-slices] provides the definition of a network slice for use within the IETF and discusses the general framework for requesting and operating IETF Network Slices, their characteristics, and the necessary system components and interfaces. It also discusses 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-Flow Aggregate which comprises of one or more IETF network slice traffic streams. It also describes the Network Resource Partition (NRP) and the NRP Policy
that can be used to instantiate control and data plane behaviors on select topological elements associated with the NRP that supports a Slice-Flow Aggregate – refer Section 5.1 for further details.

The IETF Network Slice Controller is responsible for the aggregation of multiple IETF network traffic streams into a Slice-Flow Aggregate, and for maintaining the mapping required between them. The mechanisms used by the controller to determine the mapping of one or more IETF network slice to a Slice-Flow Aggregate 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 Diffserv (DS) domain [RFC2475], packets requiring the same forwarding treatment (scheduling and drop policy) are classified and marked with the respective Class Selector (CS) Codepoint (or the Traffic Class (TC) field for MPLS packets [RFC5462]) at the DS domain ingress nodes. Such packets are said to belong to a Behavior Aggregates (BA) that has a common set of behavioral characteristics or a common set of delivery requirements. At transit nodes, the CS is inspected to determine the specific forwarding treatment to be applied before the packet is forwarded. A similar approach is adopted in this document to realize network slicing. The solution proposed in this document does not mandate Diffserv to be enabled in the network to provide a specific forwarding treatment.

When logical networks associated with an NRP are realized on top of a shared physical network infrastructure, it is important to steer traffic on the specific network resources partition that is allocated for a given Slice-Flow Aggregate. In packet networks, the packets of a specific Slice-Flow Aggregate may be identified by one or more specific fields carried within the packet. An NRP ingress boundary node (where Slice-Flow Aggregate traffic enters the NRP) populates the respective field(s) in packets that are mapped to a Slice-Flow Aggregate in order to allow interior NRP nodes to identify and apply the specific Per NRP Hop Behavior (NRP-PHB) associated with the Slice-Flow Aggregate. The NRP-PHB defines the scheduling treatment and, in some cases, the packet drop probability.

If Diffserv is enabled within the network, the Slice-Flow Aggregate traffic can further carry a Diffserv CS to enable differentiation of forwarding treatments for packets within a Slice-Flow Aggregate.
For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same Slice-Flow Aggregate by carrying an identifier in an MPLS Label Stack Entry (LSE). 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 NRP.

This document covers different modes of NRPs and discusses how each mode can ensure proper placement of Slice-Flow Aggregate paths and respective treatment of Slice-Flow Aggregate traffic.

1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slices].

The following terminology is used in the document:

IETF Network Slice: refer to the definition of 'IETF network slice' in [I-D.ietf-teas-ietf-network-slices].

IETF Network Slice Controller (NSC): refer to the definition in [I-D.ietf-teas-ietf-network-slices].

Network Resource Partition: the set of network resources that are used to support a Slice-Flow Aggregate to meet the requested SLOs and SLEs.

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

Network Resource Partition Policy (NRP): 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 an NRP Policy results in the creation of an NRP.

NRP Identifier (NRP-ID): an identifier that is globally unique within an NRP domain and that can be used in the control or management plane to identify the resources associated with the NRP.
NRP Capable Node:
a node that supports one of the NRP modes described in this
document.

NRP Incapable Node:
a node that does not support any of the NRP modes described in
this document.

Slice-Flow Aggregate Path:
a path that is setup over the NRP that is associated with a
specific Slice-Flow Aggregate.

Slice-Flow Aggregate Packet:
a packet that traverses over the NRP that is associated with a
specific Slice-Flow Aggregate.

NRP Topology:
a set of topological elements associated with a Network Resource
Partition.

NRP state aware TE (NRP-TE):
a mechanism for TE path selection that takes into account the
available network resources associated with a specific NRP.

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

NRP-PHB: NRP Per Hop Behavior as described in Section 5.1.3

FAS: Flow Aggregate Selector

FASL: Flow Aggregate Selector Label as described in Section 5.1.1

SLA: Service Level Agreements

SLO: Service Level Objectives

SLE: Service Level Expectations
2. Network Resource Slicing Membership

An NRP that supports a Slice-Flow Aggregate can be instantiated over 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. The NRP topology may be comprised of dedicated and/or shared network resources (e.g., in terms of processing power, storage, and bandwidth).

The physical network resources may be fully dedicated to a specific Slice-Flow Aggregate. For example, traffic belonging to a Slice-Flow Aggregate can traverse dedicated network resources without being subjected to contention from traffic of other Slice-Flow Aggregates. Dedicated physical network resource slicing allows for simple partitioning of the physical network resources amongst Slice-Flow Aggregates without the need to distinguish packets traversing the dedicated network resources since only one Slice-Flow Aggregate traffic stream can traverse the dedicated resource at any time.
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 NRPs that support multiple Slice-Flow Aggregates. The shared physical 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. IETF Network Slice Realization

Figure 1 describes the steps required to realize an IETF network slice service in a provider network using the solution proposed in this document. Each of the steps is further elaborated on in a subsequent section.
3.1. Network Topology Filters

The Physical Network may be filtered into a number of Policy Filter Topologies. Filter actions may include selection of specific nodes and links according to their capabilities and are based on network-wide policies. The resulting topologies can be used to host IETF Network Slices and provide a useful way for the network operator to know that all of the resources they are using to plan a network slice meet specific SLOs. This step can be done offline during planning activity, or could be performed dynamically as new demands arise.

Section 5.1.4 describes how topology filters can be associated with the NRP instantiated by the NRP Policy.

3.2. IETF Network Slice Service Request

The customer requests an IETF Network Slice Service specifying the CE-AC-PE points of attachment, the connectivity matrix, and the SLOs/SLEs as described in [I-D.ietf-teas-ietf-network-slices]. These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice customer and the provider.

This defines the traffic flows that need to be supported when the slice is realized. Depending on the mechanism and encoding of the Attachment Circuit (AC), the IETF Network Slice Service may also include information that will allow the operator’s controllers to configure the PEs to determine what customer traffic is intended for this IETF Network Slice.
IETF Network Slice Service Requests are likely to arrive at various times in the life of the network, and may also be modified.

3.3. Slice-Flow Aggregation Mapping

A network may be called upon to support very many IETF Network Slices, and this could present scaling challenges in the operation of the network. In order to overcome this, the IETF Network Slice streams may be aggregated into groups according to similar characteristics.

A Slice-Flow Aggregate is a construct that comprises the traffic flows of one or more IETF Network Slices. The mapping of IETF Network Slices into an Slice-Flow Aggregate is a matter of local operator policy is a function executed by the Controller. The Slice-Flow Aggregate may be preconfigured, created on demand, or modified dynamically.

3.4. Path Placement over NRP Topology

Depending on the underlying network technology, the paths are selected in the network in order to best deliver the SLOs for the different services carried by the Slice-Flow Aggregate. The path placement function (carried on ingress node or by a controller) is performed on the Policy Filtered Topology that is selected to support the Slice-Flow Aggregate.

Note that this step may indicate the need to increase the capacity of the underlying Policy Filter Topology or to create a new Policy Filter Topology.

3.5. NRP Policy Installation

A Controller function programs the physical network with policies for handling the traffic flows belonging to the Slice-Flow Aggregate. These policies instruct underlying routers how to handle traffic for a specific Slice-Flow Aggregate: the routers correlate markers present in the packets that belong to the Slice-Flow Aggregate. The way in which the NRP Policy is installed in the routers and the way that the traffic is marked is implementation specific. The NRP Policy instantiation in the network is further described in Section 5.
3.6. Path Instantiation

Depending on the underlying network technology, a Controller function may install the forwarding state specific to the Slice-Flow Aggregate so that traffic is routed along paths derived in the Path Placement step described in Section 3.4. The way in which the paths are instantiated is implementation specific.

3.7. Service Mapping

The edge points (PEs) can be configured to support the network slice service by mapping the customer traffic to Slice-Flow Aggregates, possibly using information supplied when the IETF network slice service was requested. The edge points MAY also be instructed to mark the packets so that the network routers will know which policies and routing instructions to apply.

3.8. Network Slice-Flow Aggregate Relationships

The following describes the generalization relationships between the IETF network slice and different parts of the solution as described in Figure 1.

- A customer may request 1 or more IETF Network Slices.

- Any given Attachment Circuit (AC) may support the traffic for 1 or more IETF Network Slice, but if there is more than one IETF Network Slice using a single AC, the IETF Network Slice Service request must include enough information to allow the edge nodes to demultiplex the traffic for the different IETF Network Slices.

- By definition, multiple IETF Network Slices may be mapped to a single Slice-Flow Aggregate. However, it is possible for a Slice-Flow Aggregate to contain just a single IETF Network Slice.

- The physical network may be filtered to multiple Policy Filter Topologies. Each such Policy Filter Topology facilitates planning the placement and support of the Slice-Flow Aggregate by presenting only the subset of links and nodes that meet specific criteria. Note, however, that a network operator does not need to derive any Policy Filter Topologies, choosing to operate directly on the full physical network.

- It is anticipated that there may be very many IETF Network Slices supported by a network operator over a single physical network. A network may support a limited number of Slice-Flow Aggregates, with each of the Slice-Flow Aggregates grouping any number of the IETF Network Slices streams.
4. Network Resource Partition Modes

An NRP Policy can be used to dictate if the network resource partitioning of the shared network resources among multiple Slice-Flow Aggregates can be achieved:

a) in data plane only,
b) in control plane only, or
c) in both control and data planes.

4.1. Data plane Network Resource Partition 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 Codepoint (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 NRP mode is applied, packets need to be forwarded on the specific NRP that supports the Slice-Flow Aggregate to ensure the proper forwarding treatment dictated in the NRP Policy is applied (refer to Section 5.1 below). In this case, a Flow Aggregate Selector (FAS) MUST be carried in each packet to identify the Slice-Flow Aggregate that it belongs to.

The ingress node of an NRP domain MAY also add an FAS to each Slice-Flow Aggregate packet. The transit nodes within an NRP domain MAY use the FAS to associate packets with a Slice-Flow Aggregate and to determine the Network Resource Partition Per Hop Behavior (NRP-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-Flow Aggregate.

When data plane only NRP mode is used, routers may rely on a network state independent view of the topology to determine the best paths. In this case, the best path selection dictates the forwarding path of packets to the destination. The FAS field carried in each packet determines the specific NRP-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. An NRP Policy may be used to allow links along the least latency path to share its data plane resources amongst multiple Slice-Flow
Aggregates. In this case, the packets that are steered on a specific NRP carry the FAS that enables routers (along with the Diffserv CS) to determine the NRP-PHB to enforce on the Slice-Flow Aggregate traffic streams.

4.2. Control Plane Network Resource Partition Mode

Multiple NRPs can be realized over the same set of physical resources. Each NRP is identified by an identifier (NRP-ID) that is globally unique within the NRP domain. The NRP state reservations for each NRP can be maintained on the network element or on a controller.

The network reservation states for a specific partition can be represented in a topology that contains all or a subset of the physical network elements (nodes and links) and reflect the network state reservations in that NRP. The logical network resources that appear in the NRP topology can reflect a part, whole, or in-excess of the physical network resource capacity (e.g., when oversubscription is desirable).

For example, the physical link bandwidth can be divided into fractions, each dedicated to an NRP that supports a Slice-Flow Aggregate. The topology associated with the NRP supporting a Slice-Flow Aggregate can be used by routing protocols, or by the ingress/PCE when computing NRP state aware TE paths.

To perform NRP state aware Traffic Engineering (NRP-TE), the resource reservation on each link needs to be NRP aware. The NRP reservations state can be managed locally on the device or off device (e.g. on a controller). Details of required IGP extensions to support NRP-TE are described in [I-D.bestbar-lsr-slice-aware-te].

The same physical link may be member of multiple slice policies that instantiate different NRPs. The NRP reservable or utilized bandwidth on such a link is updated (and may be advertised) whenever new paths are placed in the network. The NRP reservation state, in this case, MAY be maintained on each device or off the device on a resource reservation manager that holds reservation states for those links in the network.

Multiple NRPs that support Slice-Flow Aggregates can form a group and share the available network resources allocated to each. In this case, a node can update the reservable bandwidth for each NRP to take into consideration the available bandwidth from other NRPs in the same group.
For illustration purposes, Figure 2 describes bandwidth partitioning or sharing amongst a group of NRPs. In Figure 2a, the NRPs identified by the following NRP-IDs: NRP1, NRP2, NRP3 and NRP4 are not sharing any bandwidths between each other. In Figure 2b, the NRPs: NRP1 and NRP2 can share the available bandwidth portion allocated to each amongst them. Similarly, NRP3 and NRP4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to NRP1 or NRP2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.

(a) No bandwidth sharing between NRPs.  
(b) Sharing bandwidth between NRPs of the same group.
Figure 2: Bandwidth isolation/sharing among NRPs.

4.3. Data and Control Plane Network Resource Partition Mode

In order to support strict guarantees for Slice-Flow 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 NRP that each supports a Slice-Flow Aggregate. The ingress routers or a Path Computation Engine (PCE) may use the customized topologies and the NRP state to determine optimal path placement for specific demand flows using NRP-TE.

The data plane partitioning provides isolation for Slice-Flow Aggregate traffic, and protection when resource contention occurs due to bursts of traffic from other Slice-Flow Aggregate traffic that traverses the same shared network resource.

5. Network Resource Partition Instantiation

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

The customer of a network slice service expresses their intent by specifying requirements rather than mechanisms to realize the slice as described in Section 3.2.

The network slice controller is fed with the network slice service intent and realizes it with an appropriate Network Resource Partition Policy (NRP Policy). Multiple IETF network slices MAY be mapped to the same Slice-Flow Aggregate as described in Section 3.3.

The network wide consistent NRP Policy definition is distributed to the devices in the network as shown in Figure 1. 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-Flow Aggregate are outside the scope of this document and will be addressed in separate documents.

5.1. NRP Policy Definition

The NRP Policy is network-wide construct that is supplied to network devices, and may include rules that control the following:
* Data plane specific policies: This includes the FAS, any firewall rules or flow-spec filters, and QoS profiles associated with the NRP Policy and any classes within it.

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

* Topology membership policies: This defines the topology filter policies that dictate node/link/function membership to a specific NRP.

There is a desire for flexibility in realizing network slices to support the services across networks consisting of implementations 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 NRP Policy will facilitate the instantiation and management of the NRP on the topological elements selected by the NRP Policy topology filter. A YANG data model for the Network Resource Partition Policy instantiation on the controller and network devices is described in [I-D.bestbar-teas-yang-slice-policy].

It is also possible to distribute the NRP 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). The extensions to enable specific protocols to carry an NRP Policy definition will be described in separate documents.

5.1.1. Network Resource Partition Data Plane Selector

A router MUST be able to identify a packet belonging to a Slice-Flow Aggregate before it can apply the associated dataplane forwarding treatment or NRP-PHB. One or more fields within the packet MAY be used as an FAS to do this.

Forwarding Address Based Selector:

It is possible to assign a different forwarding address (or MPLS forwarding label in case of MPLS network) for each Slice-Flow Aggregate on a specific node in the network. [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"'. Assigning a unique forwarding address (or MPLS forwarding label)
to each Slice-Flow Aggregate allows Slice-Flow Aggregate packets destined to a node to be distinguished by the destination address (or MPLS forwarding label) that is carried in the packet.

This approach requires maintaining per Slice-Flow 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 can be reached over \(M\) different Slice-Flow Aggregates, the node assigns and advertises reachability to \(N\) unique forwarding addresses, or MPLS forwarding labels. Similarly, each node assigns a unique forwarding address (or MPLS forwarding label) for each of its \(K\) adjacencies to enable strict steering over the adjacency for each slice. The total number of control and data plane states that need to be stored and programmed in a router’s forwarding is \((N+K)\times M\) states. Hence, as \(N\), \(K\), and \(M\) parameters increase, this approach suffers from scalability challenges in both the control and data planes.

Global Identifier Based Selector:

An NRP Policy MAY include a Global Identifier FAS (GIS) field as defined in [I-D.kompella-mpls-mspl4fa] that is carried in each packet in order to associate it to the NRP supporting a Slice-Flow Aggregate, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the NRP domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding next-hop(s), and use the GIS field in the packet to infer the specific forwarding treatment that needs to be applied on the packet.

The GIS can be carried in one of multiple fields within the packet, depending on the dataplane used. For example, in MPLS networks, the GIS can be encoded within an MPLS label that is carried in the packet’s MPLS label stack. All packets that belong to the same Slice-Flow Aggregate MAY carry the same GIS in the MPLS label stack. It is also possible to have multiple GIS’s map to the same Slice-Flow Aggregate.

The GIS can be encoded in an MPLS label and may appear in several positions in the MPLS label stack. For example, the VPN service label may act as a GIS to allow VPN packets to be mapped to the Slice-Flow Aggregate. In this case, a single VPN service label acting as a GIS MAY be allocated by all Egress PEs of a VPN. Alternatively, multiple VPN service labels MAY act as GIS’s that map a single VPN to the same Slice-Flow Aggregate to allow for multiple Egress PEs to allocate different VPN service labels for a
VPN. In other cases, a range of VPN service labels acting as multiple GIS’s MAY map multiple VPN traffic to a single Slice-Flow Aggregate. An example of such deployment is shown in Figure 3.

SR Adj-SID: GIS (VPN service label) 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- |
|       |       +-------+         +-------+        +-------+
|       |       | IP   |         | Pay- |        | Load |
|       |       +-------+         +-------+        +-------+
|       |       | Pay- |         | Load |
|       |       +-------+         +-------+
|       |       +-------+         +-------+

Figure 3: GIS or VPN label at bottom of label stack.

In some cases, the position of the GIS may not be at a fixed position in the MPLS label header. In this case, the GIS label can show up in any position in the MPLS label stack. To enable a transit router to identify the position of the GIS label, a special purpose label (ideally a base special purpose label (bSPL)) can be used to indicate the presence of a GIS in the MPLS label stack. [I-D.kompella-mpls-mspl4fa] proposes a new bSPL called Forwarding Actions Identifier (FAI) that is assigned to alert of the presence of multiple actions and action data (including the presence of the GIS). The NRP ingress boundary node, in this case, imposes two labels: the FAI label and a forwarding actions label that includes the GIS to identify the Slice-Flow Aggregate packets as shown in Figure 4.

[I-D.decaene-mpls-slid-encoded-entropy-label-id] also proposes to repurpose the ELI/EL [RFC6790] to carry the Slice Identifier in order to minimize the size of the MPLS stack and ease incremental deployment.
SR Adj-SID: GIS: 1001
9012: P1-P2
9023: P2-PE2

```
+-----+        +-----+        +-----+
| PE1 | ------ | P1  | ------ | P2  |------ | PE2 |
+-----+        +-----+        +-----+
```

In packet:

```
+-----+        +-----+        +-----+        +-----+
| IP   |       | 9012 |         | 9023 |        | FAI  |
+-----+        +-----+        +-----+        +-----+
| Pay- |       | 9023 |         | FAI  |        | 1001 |
| Load | +-----+        +-----+        +-----+        +-----+ 
+-----+        +-----+        +-----+        +-----+
| FAI  |       | 1001 |         | IP   |        | Pay- |
| 1001 | +-----+        +-----+        +-----+        +-----+
| IP   |       | Pay- |        | Load |        | Load |
| Pay- | +-----+        +-----+        +-----+
| Load |
+-----+
```

Figure 4: FAI and GIS label in the label stack.

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

A detailed review of NRP scale considerations is presented in [I-D.dong-teas-nrp-scalability].

5.1.2. Network Resource Partition 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 management of available bandwidth and the 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 NRPs may also be desirable. For example, a Slice-Flow Aggregate may not be using all of the NRP reservable bandwidth; this allows other NRPs in the same group to use the available bandwidth resources for other Slice-Flow Aggregates.

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

Since network characteristics and its state can change over time, the NRP topology and its network state need 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 NRP network resources.

5.1.3. Network Resource Partition 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.

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

The Slice-Flow Aggregate traffic may be identified at NRP ingress boundary nodes by carrying a FAS 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 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-Flow Aggregate, a Slice-Flow 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 NRP.

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. Network Resource Partition Topology

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

An NRP topology can overlap or share a subset of links with another NRP topology. A number of topology filtering policies can be defined as part of the NRP Policy to limit the specific topology elements that belong to the NRP. For example, a topology filtering policy can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain links that the NRP is instantiated on in supports of the Slice-Flow Aggregate.

The NRP 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]. A YANG data model that covers generic topology filters is described in [I-D.bestbar-teas-yang-topology-filter]. Also, the Path Computation Element (PCE) Communication Protocol (PCEP) extensions to carry topology filters are defined in [I-D.xpbs-pce-topology-filter].

5.2. Network Resource Partition Boundary

A network slice originates at the edge nodes of a network slice provider. Traffic that is steered over the corresponding NRP supporting a Slice-Flow Aggregate may traverse NRP capable as well as NRP 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. Network Resource Partition Edge Nodes

NRP edge nodes sit at the boundary of a network slice provider
network and receive traffic that requires steering over network
resources specific to a NRP that supports a Slice-Flow Aggregate.
These edge nodes are responsible for identifying Slice-Flow 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 NRP 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 NRP mode is employed, the NRP ingress nodes are
responsible for adding a suitable FAS onto packets that belong to
specific Slice-Flow Aggregate. In addition, edge nodes MAY mark the
corresponding Diffserv CS to differentiate between different types of
traffic carried over the same Slice-Flow Aggregate.

5.2.2. Network Resource Partition Interior Nodes

An NRP interior node receives slice traffic and MAY be able to
identify the packets belonging to a specific Slice-Flow Aggregate by
inspecting the FAS 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-Flow Aggregate. For example, when
data plane NRP mode is applied, interior nodes can use the FAS
carried within the packet to apply the corresponding NRP-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 NRP Policy, and allow differentiation
of forwarding treatments for packets forwarded over the same NRP that
supports the Slice-Flow Aggregate.
5.2.3. Network Resource Partition Incapable Nodes

Packets that belong to a Slice-Flow Aggregate may need to traverse nodes that are NRP 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 NRP forwarding treatment once the traffic reaches devices that are NRP-capable.

When data plane NRP mode is employed, packets carry a FAS to allow slice interior nodes to identify them. To support end-to-end network slicing, the FAS MUST be maintained in the packets as they traverse devices within the network - including NRP capable and incapable devices.

For example, when the FAS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are NRP incapable without any further considerations. On the other hand when the FASL is at the top of the MPLS label stack, packets can be bypassed (or tunneled) over the NRP incapable devices towards the next device that supports NRP as shown in Figure 5.
SR Node-SID:    FASL: 1001    @@: NRP Policy enforced
1601: P1
1602: P2
1603: P3
1604: P4
1605: P5

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

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

\-----/  \-----/  \-----/  .
| P4  | ------ | P5  |
\-----/  \-----/  

Figure 5: Extending network slice over NRP incapable device(s).

5.2.4. Combining Network Resource Partition Modes

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

The usual techniques to steer traffic onto paths can be applicable when steering traffic over paths established for a specific Slice-Flow Aggregate.

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established for a Slice-Flow 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-Flow Aggregate path. External interfaces can be further partitioned (e.g., using VLANs) to allow mapping one or more VLANs to specific Slice-Flow 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 belongs to a specific Slice-Flow Aggregate. This option allows for applying a rich set of rules to identify specific packets to be mapped to a Slice-Flow Aggregate. However, it requires data plane network resources to be able to perform the additional checks in hardware.

6. Path Selection and Instantiation

6.1. Applicability of Path Selection to Slice-Flow Aggregates

The path selection in the 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 IGP's 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.

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 IGP protocols (e.g., [RFC3630] and [RFC5305]), and BGP-LS [RFC7752] to advertise such link state reservations.

When the network resource reservations are maintained for NRPs, the link state can carry per NRP state (e.g., reservable bandwidth). This allows path computation to take into account the specific network resources available for an NRP. In this case, we refer to the process of path placement and path provisioning as NRP-aware TE (NRP-TE).

6.2. Applicability of Path Control Technologies to Slice-Flow Aggregates

The NRP modes described in this document are agnostic to the technology used to setup paths that carry Slice-Flow 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 NRP that supports a Slice-Flow Aggregate.

The feasible paths can be computed using the NRP topology and network state subject the optimization metrics and constraints.

6.2.1. RSVP-TE Based Slice-Flow Aggregate Paths

RSVP-TE [RFC3209] can be used to signal LSPs over the computed feasible paths in order to carry the Slice-Flow Aggregate traffic. The specific extensions to the RSVP-TE protocol required to enable signaling of NRP-aware RSVP LSPs are outside the scope of this document.

6.2.2. SR Based Slice-Flow Aggregate Paths

Segment Routing (SR) [RFC8402] can be used to setup and steer traffic over the computed Slice-Flow Aggregate feasible paths.

The SR architecture defines a number of building blocks that can be leveraged to support the realization of NRPs that support Slice-Flow Aggregates in an SR network.

Such building blocks include:

* SR Policy with or without Flexible Algorithm.
* Steering of services (e.g. VPN) traffic over SR paths
* SR Operation, Administration and Management (OAM) and Performance Management (PM)

SR allows a headend node to steer packets onto specific SR paths using a Segment Routing Policy (SR Policy). The SR policy supports various optimization objectives and constraints and can be used to steer Slice-Flow Aggregate traffic in the SR network.

The SR policy can be instantiated with or without the IGP Flexible Algorithm (Flex-Algorithm) feature. It may be possible to dedicate a single SR Flex-Algorithm to compute and instantiate SR paths for one Slice-Flow Aggregate traffic. In this case, the SR Flex-Algorithm computed paths and Flex-Algorithm SR SIDs are not shared by other Slice-Flow Aggregates traffic. However, to allow for better scale, it may be desirable for multiple Slice-Flow Aggregates traffic to share the same SR Flex-Algorithm computed paths and SIDs. Further details on how the NRP modes presented in this document can be realized in an SR network are discussed in [I-D.bestbar-spring-scalable-ns], and [I-D.bestbar-lsr-spring-sa].


Routing protocols may need to be extended to carry additional per NRP 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 an NRP Policy may need to be automated. Multiple options are possible to facilitate automation of distribution of an NRP Policy to capable devices.

For example, a YANG data model for the NRP 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 NRP Policy YANG data model is outside the scope of this document, and is defined in [I-D.bestbar-teas-yang-slice-policy].

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 NRP is indicated primarily by the FAS, and hence an adversary may be able to utilize resources allocated to a specific NRP by injecting packets carrying the same FAS 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 NRP.

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

10. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, Prabhu Raj Villadathu Karunakaran, Jie Dong, and Mohamed Boucadair for their review of this document and for providing valuable feedback on it. The authors would also like to thank Adrian Farrel for detailed discussions that resulted in Section 3.

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-teas-yang-slice-policy]

[I-D.ietf-lsr-flex-algo]

[I-D.kompella-mpls-mspl4fa]


12.2. Informative References

[I-D.bestbar-lsr-slice-aware-te]

[I-D.bestbar-lsr-spring-sa]
[I-D.bestbar-spring-scalable-ns]

[I-D.bestbar-teas-yang-topology-filter]

[I-D.decraene-mpls-slid-encoded-entropy-label-id]

[I-D.dong-teas-nrp-scalability]

[I-D.filsfils-spring-srv6-stateless-slice-id]

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

Saad, et al. Expires 6 August 2022 [Page 31]
Internet-Draft  IP/MPLS Network Slicing  February 2022

[I-D.ietf-teas-rfc3272bis]

[I-D.xpbs-pce-topology-filter]


Authors' Addresses

Tarek Saad
Juniper Networks

Email: tsaad@juniper.net
Luis M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Reza Rokui
Nokia
Email: reza.rokui@nokia.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
Abstract

Realizing network slices may require the Service Provider to have 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. Multiple network slices can be realized on the same network while ensuring slice elasticity in terms of network resource allocation. This document describes a scalable solution to realize network slicing in IP/MPLS networks by supporting multiple services on top of a single physical network by relying on compliant domains and nodes to provide forwarding treatment (scheduling, drop policy, resource usage) on to packets that carry identifiers that indicate the slicing service that is to be applied to the packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 5 November 2022.

Copyright Notice

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

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

Table of Contents

- Introduction ............................................ 3
  1.1. Terminology ........................................ 5
  1.2. Acronyms and Abbreviations .......................... 6
- Network Resource Slicing Membership .................... 7
- IETF Network Slice Realization ........................... 8
  3.1. Network Topology Filters ............................ 9
  3.2. IETF Network Slice Service Request .................. 9
  3.3. Slice-Flow Aggregation .............................. 10
  3.4. Path Placement over NRP Filter Topology ............. 10
  3.5. NRP Policy Installation ............................. 10
  3.6. Path Instantiation ................................. 10
  3.7. Service Mapping ..................................... 11
- Network Resource Partition Modes ......................... 11
  4.1. Data plane Network Resource Partition Mode ......... 11
  4.2. Control Plane Network Resource Partition Mode ...... 12
  4.3. Data and Control Plane Network Resource Partition Mode . 14
- Network Resource Partition Instantiation ................. 14
  5.1. NRP Policy Definition ................................ 14
  5.1.1. Network Resource Partition - Flow-Aggregate Selector ........ 15
1. Introduction

Network slicing allows a Service Provider to create independent and logical networks on top of a shared physical network infrastructure. Such network slices can be offered to customers or used internally by the Service Provider to enhance the delivery of their service offerings. A Service Provider can also use network slicing to structure and organize the elements of its infrastructure. The solution discussed in this document works with any path control technology (such as RSVP-TE, or SR) that can be used by a Service Provider to realize network slicing in IP/MPLS networks.

[I-D.ietf-teas-ietf-network-slices] provides the definition of a network slice for use within the IETF and discusses the general framework for requesting and operating IETF Network Slices, their characteristics, and the necessary system components and interfaces. It also discusses 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-Flow Aggregate which comprises of one or more IETF network slice traffic streams. It also describes the Network Resource Partition (NRP) and the NRP Policy that can be used to instantiate control and data plane behaviors on select topological elements associated with the NRP that supports a Slice-Flow Aggregate – refer Section 5.1 for further details.

The IETF Network Slice Controller is responsible for the aggregation of multiple IETF network traffic streams into a Slice-Flow Aggregate, and for maintaining the mapping required between them. The mechanisms used by the controller to determine the mapping of one or more IETF network slice to a Slice-Flow Aggregate 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 Diffserv (DS) domain [RFC2475], packets requiring the same forwarding treatment (scheduling and drop policy) are classified and marked with the respective Class Selector (CS) Codepoint (or the Traffic Class (TC) field for MPLS packets [RFC5462]) at the DS domain ingress nodes. Such packets are said to belong to a Behavior Aggregate (BA) that has a common set of behavioral characteristics or a common set of delivery requirements. At transit nodes, the CS is inspected to determine the specific forwarding treatment to be applied before the packet is forwarded. A similar approach is adopted in this document to realize network slicing. The solution proposed in this document does not mandate Diffserv to be enabled in the network to provide a specific forwarding treatment.

When logical networks associated with an NRP are realized on top of a shared physical network infrastructure, it is important to steer traffic on the specific network resources partition that is allocated for a given Slice-Flow Aggregate. In packet networks, the packets of a specific Slice-Flow Aggregate may be identified by one or more specific fields carried within the packet. An NRP ingress boundary node (where Slice-Flow Aggregate traffic enters the NRP) populates the respective field(s) in packets that are mapped to a Slice-Flow Aggregate in order to allow interior NRP nodes to identify and apply the specific Per NRP Hop Behavior (NRP-PHB) associated with the Slice-Flow Aggregate. The NRP-PHB defines the scheduling treatment and, in some cases, the packet drop probability.

If Diffserv is enabled within the network, the Slice-Flow Aggregate traffic can further carry a Diffserv CS to enable differentiation of forwarding treatments for packets within a Slice-Flow Aggregate.
For example, when using MPLS as a dataplane, it is possible to identify packets belonging to the same Slice-Flow Aggregate by carrying an identifier in an MPLS Label Stack Entry (LSE). 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 NRP.

This document covers different modes of NRPs and discusses how each mode can ensure proper placement of Slice-Flow Aggregate paths and respective treatment of Slice-Flow Aggregate traffic.

1.1. Terminology

The reader is expected to be familiar with the terminology specified in [I-D.ietf-teas-ietf-network-slices].

The following terminology is used in the document:

IETF Network Slice:
refer to the definition of 'IETF network slice' in [I-D.ietf-teas-ietf-network-slices].

IETF Network Slice Controller (NSC):
refer to the definition in [I-D.ietf-teas-ietf-network-slices].

Network Resource Partition:
refer to the definition in [I-D.ietf-teas-ietf-network-slices].

Slice-Flow Aggregate:
a collection of packets that match an NRP Policy and are given the same forwarding treatment; a Slice-Flow Aggregate comprises of one or more IETF network slice traffic streams; the mapping of one or more IETF network slices to a Slice-Flow Aggregate is maintained by the IETF Network Slice Controller. The boundary nodes MAY also maintain a mapping of specific IETF network slice service(s) to a SFA.

Network Resource Partition Policy (NRP):
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 an NRP Policy results in the creation of an NRP.

NRP Identifier (NRP-ID):
an identifier that is globally unique within an NRP domain and that can be used in the control or management plane to identify the resources associated with the NRP.
NRP Capable Node:
a node that supports one of the NRP modes described in this
document.

NRP Incapable Node:
a node that does not support any of the NRP modes described in
this document.

Slice-Flow Aggregate Path:
a path that is setup over the NRP that is associated with a
specific Slice-Flow Aggregate.

Slice-Flow Aggregate Packet:
a packet that traverses over the NRP that is associated with a
specific Slice-Flow Aggregate.

NRP Filter Topology:
a set of topological elements associated with a Network Resource
Partition.

NRP state aware TE (NRP-TE):
a mechanism for TE path selection that takes into account the
available network resources associated with a specific NRP.

1.2. Acronyms and Abbreviations

BA: Behavior Aggregate
CS: Class Selector
NRP-PHB: NRP Per Hop Behavior as described in Section 5.1.3
FAS: Flow Aggregate Selector
FASL: Flow Aggregate Selector Label as described in Section 5.1.1
SLA: Service Level Agreements
SLO: Service Level Objectives
SLE: Service Level Expectations
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
AC: Attachment Circuit
CE: Customer Edge
PE: Provider Edge
PCEP: Path Computation Element (PCE) Communication Protocol (PCEP)

2. Network Resource Slicing Membership

An NRP that supports a Slice-Flow Aggregate can be instantiated over 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. The NRP topology may be comprised of dedicated and/or shared network resources (e.g., in terms of processing power, storage, and bandwidth).

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

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 NRPs that support multiple Slice-Flow Aggregates. The shared physical 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. IETF Network Slice Realization

Figure 1 describes the steps required to realize an IETF network slice service in a provider network using the solution proposed in this document. While Figure 4 of [I-D.ietf-teas-ietf-network-slices] provides an abstract architecture of an IETF Network Slice, this section intends to offer a realization of that architecture specific for IP/MPLS packet networks.

Each of the steps is further elaborated on in a subsequent section.
Program the Network
(NRP Policies and Paths)*

*: NRP Policy installation and path placement can be centralized or distributed.

Figure 1: IETF network slice realization steps.

3.1. Network Topology Filters

The Physical Network may be filtered into a number of Filter Topologies. Filter actions may include selection of specific nodes and links according to their capabilities and are based on network-wide policies. The resulting topologies can be used to host IETF Network Slices and provide a useful way for the network operator to know that all of the resources they are using to plan a network slice meet specific SLOs. This step can be done offline during planning activity, or could be performed dynamically as new demands arise.

Section 5.1.4 describes how topology filters can be associated with the NRP instantiated by the NRP Policy.

3.2. IETF Network Slice Service Request

The customer requests an IETF Network Slice Service specifying the CE-AC-PE points of attachment, the connectivity matrix, and the SLOs/SLEs as described in [I-D.ietf-teas-ietf-network-slices]. These capabilities are always provided based on a Service Level Agreement (SLA) between the network slice customer and the provider.

This defines the traffic flows that need to be supported when the slice is realized. Depending on the mechanism and encoding of the Attachment Circuit (AC), the IETF Network Slice Service may also include information that will allow the operator’s controllers to configure the PEs to determine what customer traffic is intended for this IETF Network Slice.

IETF Network Slice Service Requests are likely to arrive at various times in the life of the network, and may also be modified.
3.3. Slice-Flow Aggregation

A network may be called upon to support very many IETF Network Slices, and this could present scaling challenges in the operation of the network. In order to overcome this, the IETF Network Slice streams may be aggregated into groups according to similar characteristics.

A Slice-Flow Aggregate is a construct that comprises the traffic flows of one or more IETF Network Slices. The mapping of IETF Network Slices into an Slice-Flow Aggregate is a matter of local operator policy is a function executed by the Controller. The Slice-Flow Aggregate may be preconfigured, created on demand, or modified dynamically.

3.4. Path Placement over NRP Filter Topology

Depending on the underlying network technology, the paths are selected in the network in order to best deliver the SLOs for the different services carried by the Slice-Flow Aggregate. The path placement function (carried on ingress node or by a controller) is performed on the Filter Topology that is selected to support the Slice-Flow Aggregate.

Note that this step may indicate the need to increase the capacity of the underlying Filter Topology or to create a new Filter Topology.

3.5. NRP Policy Installation

A Controller function programs the physical network with policies for handling the traffic flows belonging to the Slice-Flow Aggregate. These policies instruct underlying routers how to handle traffic for a specific Slice-Flow Aggregate: the routers correlate markers present in the packets that belong to the Slice-Flow Aggregate. The way in which the NRP Policy is installed in the routers and the way that the traffic is marked is implementation specific. The NRP Policy instantiation in the network is further described in Section 5.

3.6. Path Instantiation

Depending on the underlying network technology, a Controller function may install the forwarding state specific to the Slice-Flow Aggregate so that traffic is routed along paths derived in the Path Placement step described in Section 3.4. The way in which the paths are instantiated is implementation specific.
3.7. Service Mapping

The edge points can be configured to support the network slice service by mapping the customer traffic to Slice-Flow Aggregates, possibly using information supplied when the IETF network slice service was requested. The edge points may also be instructed to mark the packets so that the network routers will know which policies and routing instructions to apply. The steering of traffic onto Slice-Flow Aggregate paths is further described in Section 6.

4. Network Resource Partition Modes

An NRP Policy can be used to dictate if the network resource partitioning of the shared network resources among multiple Slice-Flow Aggregates can be achieved:

a) in data plane only,

b) in control plane only, or

c) in both control and data planes.

4.1. Data plane Network Resource Partition 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) codepoint 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 NRP mode is applied, packets need to be forwarded on the specific NRP that supports the Slice-Flow Aggregate to ensure the proper forwarding treatment dictated in the NRP Policy is applied (refer to Section 5.1 below). In this case, a Flow Aggregate Selector (FAS) must be carried in each packet to identify the Slice-Flow Aggregate that it belongs to.

The ingress node of an NRP domain adds a FAS field if one is not already present in each Slice-Flow Aggregate packet. In the data plane NRP mode, the transit nodes within an NRP domain use the FAS to associate packets with a Slice-Flow Aggregate and to determine the Network Resource Partition Per Hop Behavior (NRP-PHB) that is applied to the packet (refer to Section 5.1.3 for further details). The CS is used to apply a Diffserv PHB on to the packet to allow differentiation of traffic treatment within the same Slice-Flow Aggregate.
When data plane only NRP mode is used, routers may rely on a network state independent view of the topology to determine the best paths. In this case, the best path selection dictates the forwarding path of packets to the destination. The FAS field carried in each packet determines the specific NRP-PHB treatment along the selected path.

4.2. Control Plane Network Resource Partition Mode

Multiple NRPs can be realized over the same set of physical resources. Each NRP is identified by an identifier (NRP-ID) that is globally unique within the NRP domain. The NRP state reservations for each NRP can be maintained on the network element or on a controller.

The network reservation states for a specific partition can be represented in a topology that contains all or a subset of the physical network elements (nodes and links) and reflect the network state reservations in that NRP. The logical network resources that appear in the NRP topology can reflect a part, whole, or in-excess of the physical network resource capacity (e.g., when oversubscription is desirable).

For example, the physical link bandwidth can be divided into fractions, each dedicated to an NRP that supports a Slice-Flow Aggregate. The topology associated with the NRP supporting a Slice-Flow Aggregate can be used by routing protocols, or by the ingress/PCE when computing NRP state aware TE paths.

To perform NRP state aware Traffic Engineering (NRP-TE), the resource reservation on each link needs to be NRP aware. The NRP reservations state can be managed locally on the device or off device (e.g. on a controller).

The same physical link may be member of multiple slice policies that instantiate different NRPs. The NRP reservable or utilized bandwidth on such a link is updated (and may be advertised) whenever new paths are placed in the network. The NRP reservation state, in this case, is maintained on each device or off the device on a resource reservation manager that holds reservation states for those links in the network.

Multiple NRPs that support Slice-Flow Aggregates can form a group and share the available network resources allocated to each. In this case, a node can update the reservable bandwidth for each NRP to take into consideration the available bandwidth from other NRPs in the same group.
For illustration purposes, Figure 2 describes bandwidth partitioning or sharing amongst a group of NRPs. In Figure 2a, the NRPs identified by the following NRP-IDs: NRP1, NRP2, NRP3 and NRP4 are not sharing any bandwidths between each other. In Figure 2b, the NRPs: NRP1 and NRP2 can share the available bandwidth portion allocated to each amongst them. Similarly, NRP3 and NRP4 can share amongst themselves any available bandwidth allocated to them, but they cannot share available bandwidth allocated to NRP1 or NRP2. In both cases, the Max Reservable Bandwidth may exceed the actual physical link resource capacity to allow for over subscription.

(a) No bandwidth sharing between NRPs.  
(b) Sharing bandwidth between NRPs of the same group.
4.3. Data and Control Plane Network Resource Partition Mode

In order to support strict guarantees for Slice-Flow 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 NRP that each supports a Slice-Flow Aggregate. The ingress routers or a Path Computation Engine (PCE) may use the customized topologies and the NRP state to determine optimal path placement for specific demand flows using NRP-TE.

The data plane partitioning provides isolation for Slice-Flow Aggregate traffic, and protection when resource contention occurs due to bursts of traffic from other Slice-Flow Aggregate traffic that traverses the same shared network resource.

5. Network Resource Partition Instantiation

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

The customer of a network slice service expresses their intent by specifying requirements rather than mechanisms to realize the slice as described in Section 3.2.

The network slice controller is fed with the network slice service intent and realizes it with an appropriate Network Resource Partition Policy (NRP Policy). Multiple IETF network slices are mapped to the same Slice-Flow Aggregate as described in Section 3.3.

The network wide consistent NRP Policy definition is distributed to the devices in the network as shown in Figure 1. 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-Flow Aggregate are outside the scope of this document and will be addressed in separate documents.

5.1. NRP Policy Definition

The NRP Policy is network-wide construct that is supplied to network devices, and may include rules that control the following:
* Data plane specific policies: This includes the FAS, any firewall rules or flow-spec filters, and QoS profiles associated with the NRP Policy and any classes within it.

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

* Topology membership policies: This defines the topology filter policies that dictate node/link/function membership to a specific NRP.

There is a desire for flexibility in realizing network slices to support the services across networks consisting of implementations 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 NRP Policy will facilitate the instantiation and management of the NRP on the topological elements selected by the NRP Policy topology filter.

It is also possible to distribute the NRP 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). The extensions to enable specific protocols to carry an NRP Policy definition will be described in separate documents.

5.1.1. Network Resource Partition - Flow-Aggregate Selector

A router should be able to identify a packet belonging to a Slice-Flow Aggregate before it can apply the associated dataplane forwarding treatment or NRP-PHB. One or more fields within the packet are used as an FAS to do this.

Forwarding Address Based FAS:

It is possible to assign a different forwarding address (or MPLS forwarding label in case of MPLS network) for each Slice-Flow Aggregate on a specific node in the network. [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"'. Assigning a unique forwarding address (or MPLS forwarding label) to each Slice-Flow Aggregate allows Slice-Flow Aggregate packets destined to a node to be distinguished by the destination address (or MPLS forwarding label) that is carried in the packet.
This approach requires maintaining per Slice-Flow 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 can be reached over 'M' different Slice-Flow Aggregates, the node assigns and advertises reachability to 'N' unique forwarding addresses, or MPLS forwarding labels. Similarly, each node assigns a unique forwarding address (or MPLS forwarding label) for each of its 'K' adjacencies to enable strict steering over the adjacency for each slice. The total number of control and data plane states that need to be stored and programmed in a router’s forwarding is (N+K) * M states. Hence, as 'N', 'K', and 'M' parameters increase, this approach suffers from scalability challenges in both the control and data planes.

Global Identifier Based FAS:

An NRP Policy may include a Global Identifier FAS (G-FAS) field that is carried in each packet in order to associate it to the NRP supporting a Slice-Flow Aggregate, independent of the forwarding address or MPLS forwarding label that is bound to the destination. Routers within the NRP domain can use the forwarding address (or MPLS forwarding label) to determine the forwarding next-hop(s), and use the G-FAS field in the packet to infer the specific forwarding treatment that needs to be applied on the packet.

The G-FAS can be carried in one of multiple fields within the packet, depending on the dataplane used. For example, in MPLS networks, the G-FAS can be encoded within an MPLS label that is carried in the packet’s MPLS label stack. All packets that belong to the same Slice-Flow Aggregate may carry the same G-FAS in the MPLS label stack. It is also possible to have multiple G-FAS’s map to the same Slice-Flow Aggregate.

The G-FAS can be encoded in an MPLS label and may appear in several positions in the MPLS label stack. For example, the VPN service label may act as a G-FAS to allow VPN packets to be mapped to the Slice-Flow Aggregate. In this case, a single VPN service label acting as a G-FAS may be allocated by all Egress PEs of a VPN. Alternatively, multiple VPN service labels may act as G-FAS’s that map a single VPN to the same Slice-Flow Aggregate to allow for multiple Egress PEs to allocate different VPN service labels for a VPN. In other cases, a range of VPN service labels acting as multiple G-FAS’s may map multiple VPN traffic to a single Slice-Flow Aggregate. An example of such deployment is shown in Figure 3.
SR Adj-SID: G-FAS (VPN service label) on PE2: 1001
9012: P1-P2
9023: P2-PE2

In packet:

<table>
<thead>
<tr>
<th>IP</th>
<th>9012</th>
<th>9023</th>
<th>1001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay-</td>
<td>9023</td>
<td>1001</td>
<td>IP</td>
</tr>
<tr>
<td>Load</td>
<td>1001</td>
<td>IP</td>
<td>Pay-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pay-</td>
<td>Load</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Load</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: G-FAS or VPN label at bottom of label stack.

In some cases, the position of the G-FAS may not be at a fixed position in the MPLS label header. In this case, the G-FAS label can show up in any position in the MPLS label stack. To enable a transit router to identify the position of the G-FAS label, a special purpose label can be used to indicate the presence of a G-FAS in the MPLS label stack as shown in Figure 4.
SR Adj-SID: G-FAS: 1001
9012: P1-P2
9023: P2-PE2

\[-----\] \[-----\] \[-----\] \[-----\]
| PE1 | ----- | P1 | ----- | P2 | ----- | PE2 |
\[-----\] \[-----\] \[-----\] \[-----\]

In packet:

<table>
<thead>
<tr>
<th>IP</th>
<th>9012</th>
<th>9023</th>
<th>FAI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay-</td>
<td>9023</td>
<td>FAI</td>
<td>1001</td>
</tr>
<tr>
<td>Load</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: FAI and G-FAS label in the label stack.

When the slice is realized over an IP dataplane, the G-FAS can be encoded in the IP header (e.g. as an IPv6 option header).

5.1.2. Network Resource Partition 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 management of available bandwidth and the 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 NRPs may also be desirable. For example, a Slice-Flow Aggregate may not be using all of the NRP reservable bandwidth; this allows other NRPs in the same group to use the available bandwidth resources for other Slice-Flow Aggregates.
Congestion on shared network resources may result from sub-optimal placement of paths in different slice policies. When this occurs, preemption of some Slice-Flow Aggregate paths may be desirable to alleviate congestion. A preference-based allocation scheme enables prioritization of Slice-Flow Aggregate paths that can be preempted.

Since network characteristics and its state can change over time, the NRP topology and its network state need 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 NRP network resources.

5.1.3. Network Resource Partition 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.

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

The Slice-Flow Aggregate traffic may be identified at NRP ingress boundary nodes by carrying a FAS 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 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-Flow Aggregate, a Slice-Flow 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 NRP.

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. Network Resource Partition Topology

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

An NRP topology can overlap or share a subset of links with another NRP topology. A number of topology filtering policies can be defined as part of the NRP Policy to limit the specific topology elements that belong to the NRP. For example, a topology filtering policy can leverage Resource Affinities as defined in [RFC2702] to include or exclude certain links that the NRP is instantiated on in support of the Slice-Flow Aggregate.

The NRP 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 in [RFC4915]).

5.2. Network Resource Partition Boundary

A network slice originates at the edge nodes of a network slice provider. Traffic that is steered over the corresponding NRP supporting a Slice-Flow Aggregate may traverse NRP capable as well as NRP 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. Network Resource Partition Edge Nodes

NRP edge nodes sit at the boundary of a network slice provider network and receive traffic that requires steering over network resources specific to a NRP that supports a Slice-Flow Aggregate. These edge nodes are responsible for identifying Slice-Flow 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 NRP 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 NRP mode is employed, the NRP ingress nodes are responsible for adding a suitable FAS onto packets that belong to specific Slice-Flow Aggregate. In addition, edge nodes may mark the corresponding Diffserv CS to differentiate between different types of traffic carried over the same Slice-Flow Aggregate.

5.2.2. Network Resource Partition Interior Nodes

An NRP interior node receives slice traffic and may be able to identify the packets belonging to a specific Slice-Flow Aggregate by inspecting the FAS 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-Flow Aggregate. For example, when data plane NRP mode is applied, interior nodes can use the FAS carried within the packet to apply the corresponding NRP-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 NRP Policy, and allow differentiation of forwarding treatments for packets forwarded over the same NRP that supports the Slice-Flow Aggregate.

5.2.3. Network Resource Partition Incapable Nodes

Packets that belong to a Slice-Flow Aggregate may need to traverse nodes that are NRP 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 NRP forwarding treatment once the traffic reaches devices that are NRP-capable.

When data plane NRP mode is employed, packets carry a FAS to allow slice interior nodes to identify them. To support end-to-end network slicing, the FAS is maintained in the packets as they traverse devices within the network - including NRP capable and incapable devices.

For example, when the FAS is an MPLS label at the bottom of the MPLS label stack, packets can traverse over devices that are NRP incapable without any further considerations. On the other hand when the FASL is at the top of the MPLS label stack, packets can be bypassed (or tunneled) over the NRP incapable devices towards the next device that supports NRP as shown in Figure 5.
5.2.4. Combining Network Resource Partition Modes

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

The usual techniques to steer traffic onto paths can be applicable when steering traffic over paths established for a specific Slice-Flow Aggregate.

For example, one or more (layer-2 or layer-3) VPN services can be directly mapped to paths established for a Slice-Flow 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-Flow Aggregate path. External interfaces can be further partitioned (e.g., using VLANs) to allow mapping one or more VLANs to specific Slice-Flow 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 belongs to a specific Slice-Flow Aggregate. This option allows for applying a rich set of rules to identify specific packets to be mapped to a Slice-Flow Aggregate. However, it requires data plane network resources to be able to perform the additional checks in hardware.

6.1. Network Slice-Flow Aggregate Relationships

The following describes the generalization relationships between the IETF network slice and different parts of the solution as described in Figure 1.

- A customer may request one or more IETF Network Slices.
- Any given Attachment Circuit (AC) may support the traffic for one or more IETF Network Slices. If there is more than one IETF Network Slice using a single AC, the IETF Network Slice Service request must include enough information to allow the edge nodes to demultiplex the traffic for the different IETF Network Slices.
- By definition, multiple IETF Network Slices may be mapped to a single Slice-Flow Aggregate. However, it is possible for an Slice-Flow Aggregate to contain just a single IETF Network Slice.
o The physical network may be filtered to multiple Filter Topologies. Each such Filter Topology facilitates planning the placement of paths for the Slice-Flow Aggregate by presenting only the subset of links and nodes that meet specific criteria. Note, however, in absence of any Filter Topology, Slice-Flow Aggregate are free to operate over the full physical network.

o It is anticipated that there may be very many IETF Network Slices supported by a network operator over a single physical network. A network may support a limited number of Slice-Flow Aggregates, with each of the Slice-Flow Aggregates grouping any number of the IETF Network Slices streams.

7. Path Selection and Instantiation

7.1. Applicability of Path Selection to Slice-Flow Aggregates

In State-dependent TE [I-D.ietf-teas-rfc3272bis], the path selection adapts based on the current state of the network. The state of the network can be based on parameters flooded by the routers as described in [RFC2702]. 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). TE extensions exist today to allow IGP (e.g., [RFC3630] and [RFC5305]), and BGP-LS [RFC7752] to advertise such link state reservations.

When the network resource reservations are maintained for NRPs, the link state can carry per NRP state (e.g., reservable bandwidth). This allows path computation to take into account the specific network resources available for an NRP. In this case, we refer to the process of path placement and path provisioning as NRP aware TE (NRP-TE).

7.2. Applicability of Path Control Technologies to Slice-Flow Aggregates

The NRP modes described in this document are agnostic to the technology used to setup paths that carry Slice-Flow 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 NRP that supports a Slice-Flow Aggregate.

The feasible paths can be computed using the NRP topology and network state subject the optimization metrics and constraints.
7.2.1. RSVP-TE Based Slice-Flow Aggregate Paths

RSVP-TE [RFC3209] can be used to signal LSPs over the computed feasible paths in order to carry the Slice-Flow Aggregate traffic. The specific extensions to the RSVP-TE protocol required to enable signaling of NRP aware RSVP-TE LSPs are outside the scope of this document.

7.2.2. SR Based Slice-Flow Aggregate Paths

Segment Routing (SR) [RFC8402] can be used to setup and steer traffic over the computed Slice-Flow Aggregate feasible paths.

The SR architecture defines a number of building blocks that can be leveraged to support the realization of NRPs that support Slice-Flow Aggregates in an SR network.

Such building blocks include:

* SR Policy with or without Flexible Algorithm.

* Steering of services (e.g. VPN) traffic over SR paths

* SR Operation, Administration and Management (OAM) and Performance Management (PM)

SR allows a headend node to steer packets onto specific SR paths using a Segment Routing Policy (SR Policy). The SR policy supports various optimization objectives and constraints and can be used to steer Slice-Flow Aggregate traffic in the SR network.

The SR policy can be instantiated with or without the IGP Flexible Algorithm (Flex-Algorithm) feature. It may be possible to dedicate a single SR Flex-Algorithm to compute and instantiate SR paths for one Slice-Flow Aggregate traffic. In this case, the SR Flex-Algorithm computed paths and Flex-Algorithm SR SIDs are not shared by other Slice-Flow Aggregates traffic. However, to allow for better scale, it may be desirable for multiple Slice-Flow Aggregates traffic to share the same SR Flex-Algorithm computed paths and SIDs.


Routing protocols may need to be extended to carry additional per NRP 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 an NRP Policy may need to be automated. Multiple options are possible to facilitate automation of distribution of an NRP Policy to capable devices.

For example, a YANG data model for the NRP 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 NRP Policy YANG data model is outside the scope of this document.

9. Outstanding Issues

Note to RFC Editor: Please remove this section prior to publication.

This section records non-blocking issues that were raised during the Working Group Adoption Poll for the document. The below list of issues needs to be fully addressed before progressing the document to publication in IESG.

1. Add new Appendix section with examples for the NRP modes described in Section 4.

2. Add text to clarify the relationship between Slice-Flow Aggregates, the NRP Policy, and the NRP.

3. Remove redundant references to Diffserv behaviors.

4. Elaborate on the SFA packet treatment when no rules to associate the packet to an NRP are defined in the NRP Policy.

5. Clarify the NRP instantiation through the NRP Policy enforcement.

6. Clarify how the solution caters to the different IETF Network Slice Service Demarcation Point locations described in Section 4.2 of [I-D.ietf-teas-ietf-network-slices].

7. Clarify the relationship the underlay physical network, the filter topology and the NRP resources.

8. Expand on how isolation between NRPs can be realized depending on the deployed NRP mode.

9. Revise Section 5.2.3 to describe how nodes can discover NRP incapable downstream neighbors.
10. Expand Section 11 on additional security threats introduced with the solution.

11. Expand Section 5.2 on NRP domain boundary and multi-domain aspects.

10. IANA Considerations

This document has no IANA actions.

11. 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 NRP is indicated primarily by the FAS, and hence an adversary may be able to utilize resources allocated to a specific NRP by injecting packets carrying the same FAS 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 NRP.

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

12. Acknowledgement

The authors would like to thank Krzysztof Szarkowicz, Swamy SRK, Navaneetha Krishnan, Prabhu Raj Villadathu Karunakaran, and Mohamed Boucadair for their review of this document and for providing valuable feedback on it. The authors would also like to thank Adrian Farrel for detailed discussions that resulted in Section 3.

13. Contributors

The following individuals contributed to this document:
14. References

14.1. Normative References


14.2. Informative References
[I-D.ietf-lsr-flex-algo]

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

[I-D.ietf-teas-rfc3272bis]


Authors’ Addresses

Tarek Saad
Juniper Networks
Email: tsaad@juniper.net

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

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

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

Reza Rokui
Ciena
Email: rrokui@ciena.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com
Abstract

A Network Resource Partition (NRP) is a collection of resources identified in the underlay network to support services (like IETF Network Slices) that need logical network structures with required characteristics to be created. An NRP policy is a policy construct that enables instantiation of mechanisms in support of service specific control and data plane behaviors on select topological elements associated with the NRP. This document defines a YANG data model for the management of NRP policies on NRP 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

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

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

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

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

Copyright Notice

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

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

Table of Contents

1. Introduction .................................. 3
   1.1. Terminology ............................. 4
   1.2. Tree Structure .......................... 4
2. NRP Policy Data Model .......................... 4
   2.1. Model Usage ............................. 4
   2.2. Model Structure .......................... 4
   2.3. NRP Policies ............................. 5
      2.3.1. Resource Reservation ................. 5
      2.3.2. Flow Aggregate Selector .............. 6
      2.3.3. Per-Hop-Behavior ..................... 7
      2.3.4. Topology ............................ 7
   2.4. YANG Module ............................. 9
3. Acknowledgements ............................. 19
4. Contributors ................................. 19
5. IANA Considerations ........................... 19
6. Security Considerations ...................... 20
7. References .................................. 20
   7.1. Normative References .................... 20
   7.2. Informative References .................. 22
Appendix A. Complete Model Tree Structure ........... 22
Authors’ Addresses ............................ 25
1. Introduction

An IETF Network Slice [I-D.ietf-teas-ietf-network-slices] is a service that provides connectivity coupled with a set of specific commitments of network resources between a number of endpoints over a shared underlay network. The IETF Network Slice service is expressed in terms of one or more connectivity constructs. One or more connectivity constructs from one or more IETF Network Slices are mapped to a set of network resources called a Network Resource Partition (NRP). An NRP is a collection of resources identified in the underlay network to support the IETF Network Slice service (or any other service that needs logical network structures with required characteristics to be created). An NRP Policy [I-D.bestbar-teas-ns-packet] is a policy construct that enables instantiation of mechanisms in support of service specific control and data plane behaviors on select topological elements associated with the NRP.

An NRP policy specifies the rules for determining the topology associated with the NRP and dictates how an NRP can be realized in IP/MPLS networks using one of three modes. The NRP 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 NRP policy modes (a) and (c) require the forwarding engine on each NRP capable node to identify the traffic belonging to a specific flow aggregate and to apply the corresponding Per-Hop Behavior (PHB) that determines the forwarding treatment of the packets belonging to the flow aggregate. The identification of the flow aggregate that the packet belongs to and the corresponding forwarding treatment that needs to be applied to the packet is dictated by the NRP policy.

When catering to IETF Network Slices, this flow aggregate is referred to as the Slice-Flow Aggregate [I-D.bestbar-teas-ns-packet] and comprises of traffic streams from one or more connectivity constructs (belonging to one or more IETF network slices) mapped to a specific NRP.

The NRP policy modes (b) and (c) require the distributed/centralized resource reservation manager in the control plane to manage NRP resource reservation. The provisions for enabling NRP state aware traffic engineering (NRP-TE) [I-D.bestbar-teas-ns-packet] are dictated by the NRP policy.

This document defines a YANG data model for the management of NRP policies on NRP 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-slices] and [I-D.bestbar-teas-ns-packet].

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. NRP Policy Data Model

2.1. Model Usage

A controller that consumes the IETF Network Slice service requests determines which specific connectivity constructs from one or more slices can be grouped together. This could be based on a specific set of SLOs and SLEs, or on any administrative or operational reason.

A controller function that has visibility of the underlay network and its resources maps these connectivity constructs onto the NRP. It also constructs and distributes the network wide consistent NRP policy (using the data model defined in this document) to the relevant NRP capable nodes and controllers.

2.2. Model Structure

The high-level model structure defined by this document is as shown below:
module: ietf-nrp-policy
    augment /nw:networks:
        +++rw nrp-policies
        +++rw nrp-policy* [name]
            +++rw name  string
            +++rw nrp-id? uint32
            +++rw resource-reservation
                |  + ............
            +++rw flow-agg-selector
                |  + ............
            +++rw phb?  string
        +++rw topology
            +++rw filters
                +++rw filter* [filter-ref]
                    + ............
                +++rw resource-reservation
                    |  + ............
                +++rw flow-agg-selector
                    |  + ............
                +++rw phb?  string

The ‘networks’ container from the ‘ietf-network’ module [RFC8345] provides a placeholder for an inventory of nodes in the network. This container is augmented to carry a set of NRP policies.

2.3. NRP Policies

The ‘nrp-policies’ container carries a list of NRP policies. Each ‘nrp-policy’ entry is identified by a name and holds the set of attributes needed to instantiate the NRP. Each entry also carries an ‘nrp-id’ leaf which uniquely identifies the NRP created by the enforcement of this policy. The key elements of each nrp-policy entry are discussed in the following sub-sections.

2.3.1. Resource Reservation

The ‘resource-reservation’ container carries data nodes that are used to support NRP state aware bandwidth engineering. The data nodes in this container facilitate preference-based preemption of NRP state aware TE paths, sharing of resources amongst a group of NRPs and backup path bandwidth protection.
2.3.2. Flow Aggregate Selector

The 'flow-agg-selector' container carries data nodes that specify the rules for identifying which packets belong to the flow aggregate that this NRP caters to.
2.3.3. Per-Hop-Behavior

The 'phb' leaf carries a name of a PHB profile available on the topological element where the policy is being enforced.

++-rw phb?                    string

2.3.4. Topology

The 'topology' container consists of a list of filters where each entry references a topology filter [I-D.bestbar-teas-yang-topology-filter]. The resultant topology from the union of these filters is referred to as the NRP topology. The topological elements that satisfy the membership criteria can optionally override the default resource-reservation, flow-agg-selector and phb specific leafs.
++--rw topology
  ++--rw filters
    ++--rw filter* [filter-ref]
      ++--rw filter-ref
        |       nrp-policy-topo-filter-ref
    ++--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-nrp-id?
        |           uint32
      ++--rw (backup-bw-type)?
        |        +--:(backup-bw-value)
        |              |        ++--rw backup-bandwidth?  uint64
        |              |        +--:(backup-bw-percentage)
        |                      +--rw backup-bandwidth-percent?
        |                             rt-types:percentage
    ++--rw flow-agg-selector
      ++--rw mpls
        ++--rw (fas-type)?
          |        +--:(label)
          |          |        ++--rw (specification-type)?
          |          |            |        +--:(derived)
          |          |            |              |        ++--rw forwarding-label?
          |          |            |                        |       empty
          |          |            |                        |        +--:(explicit)
          |          |            |                        |              |        ++--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?
2.4. YANG Module

<CODE BEGINS> file "ietf-nrp-policy@2022-03-07.yang"

module ietf-nrp-policy {
    yang-version 1.1;
    prefix nrp-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-network {
        prefix nw;
        reference
            "RFC 8345: A YANG Data Model for Network Topologies";
    }

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

    import ietf-topology-filter {
        prefix topo-filt;
        reference
            "draft-bestbar-teas-yang-topology-filter: YANG Data Model";
    }

    identityref
        +--rw label-position-offset? uint8
    +-rw ipv4
        +--rw destination-prefix* inet:ipv4-prefix
    +-rw ipv6
        +--rw (fas-type)?
        |      +--:(ipv6-destination)
        |        +--rw destination-prefix* inet:ipv6-prefix
        |        +--:(ipv6-hbh-eh)
        |        |        +--rw fas-hbh-eh* uint32
    +-rw acl-ref* nrp-policy-acl-ref
    +-rw phb? string

</CODE BEGINS>
for Topology Filter";
}

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 Network
  Resource Partition Policies on Network Resource Partition
  capable nodes and controllers.

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

  Redistribution and use in source and binary forms, with or
  without modification, is permitted pursuant to, and subject to
  the license terms contained in, the Simplified BSD License set
  forth in Section 4.c of the IETF Trust’s Legal Provisions
  Relating to IETF Documents
  (https://trustee.ietf.org/license-info)."
This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.

revision 2022-03-07 {
  description "Initial revision.";
  reference "RFC XXXX: YANG Data Model for Network Resource Partition Policies.";
}

/*
 * IDENTITIES
 */

/* Identity:
 * MPLS Flow Aggregate Selector (FAS) Label Position Type.
 */

identity fas-mpls-label-position-type {
  description "Base identity for the position of the MPLS FAS label.";
}

identity fas-mpls-label-position-top {
  base fas-mpls-label-position-type;
  description "MPLS FAS label is at the top of the label stack.";
}

identity fas-mpls-label-position-bottom {
  base fas-mpls-label-position-type;
  description "MPLS FAS label is either at the bottom or at a specific offset from the bottom of the label stack.";
}

identity fas-mpls-label-position-indicator {
  base fas-mpls-label-position-type;
  description "MPLS FAS is preceded by a special purpose indicator label in the label stack.";
}

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

typedef nrp-policy-topo-filter-ref {
    type leafref {
        path "/nw:networks/topo-filt:topology-filters/
            + "topo-filt:topology-filter/topo-filt:name";
    }
    description "This type is used to reference a Topology Filter.";
}

/*----------------------------------------------------------------------------
* GROUPINGS
*----------------------------------------------------------------------------

* Grouping - MPLS FAS label location specific fields
*----------------------------------------------------------------------------

grouping nrp-pol-fas-mpls-label-location {
    description "Grouping for MPLS FAS label location specific fields.";
    leaf label-position {
        type identityref {
            base fas-mpls-label-position-type;
        }
        description "MPLS FAS label position.";
    }
    leaf label-position-offset {
        when "derived-from-or-self(../label-position,
            + "nrp-pol:fas-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 nrp-pol-flow-agg-selector {
  description
    "Grouping for Flow-Aggregate Selector (FAS).";
  container flow-agg-selector {
    description
      "Container for FAS.";
    container mpls {
      description
        "Container for MPLS FAS.";
      choice fas-type {
        description
          "Choices for MPLS FAS.";
        case label {
          choice specification-type {
            description
              "Choices for MPLS label specification.";
            case derived {
              leaf forwarding-label {
                type empty;
                description
                  "MPLS FAS Label is derived from forwarding label.";
              }
            }
            case explicit {
              leaf label {
                type rt-types:mpls-label;
                description
                  "MPLS FAS Label is explicitly specified.";
              }
            }
          }
        }
        case label-ranges {
          list label-range {
            key "index";
            unique "start-label end-label";
            description
              "Any label from the specified set of MPLS label ranges can be used as the FAS.";
          }
        }
      }
    }
  }
}
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 nrp-pol-fas-mpls-label-location;
}
}
}
container ipv4 {
  description
    "Container for IPv4 FAS.";
  leaf-list destination-prefix {
    type inet:ipv4-prefix;
    description
      "Any prefix from the specified set of IPv4 destination prefixes can be the FAS.";
  }
}
}
container ipv6 {
  description
    "Container for IPv6 FAS.";
  choice fas-type {
    description
      "Choices for IPv6 FAS.";
    case ipv6-destination {

leaf-list destination-prefix {
  type inet:ipv6-prefix;
  description "Any prefix from the specified set of IPv6
destination prefixes can be the FAS.";
}
}
case ipv6-hbh-eh {
  leaf-list fas-hbh-eh {
    type uint32;
    description "Set of FAS values carried in Hop-by-Hop
    Option of IPv6 extension header.";
  }
}
}
leaf-list acl-ref {
  type nrp-policy-acl-ref;
  description "Flow Aggregate selection is done based on the
  specified list of ACLs.";
}
}
/
* Grouping - NRP Policy Resource Reservation
*/

grouping nrp-pol-resource-reservation {
  description "Grouping for NRP policy resource reservation.";
  container resource-reservation {
    description "Container for NRP policy resource reservation.";
    leaf preference {
      type uint16;
      description "Control plane preference for the corresponding
      Network Resource Partition (NRP). 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 an NRP
    - specified as absolute value.";
}

case bw-percentage {
    leaf maximum-bandwidth-percent {
        type rt-types:percentage;
        description
        "The maximum bandwidth allocated to an NRP
        - specified as percentage of link
capacity.";
    }
}

leaf-list shared-resource-groups {
    type uint32;
    description
    "List of shared resource groups that an NRP
    shares its allocated resources with.";
}

container protection {
    description
    "Container for NRP protection reservation.";
    leaf backup-nrp-id {
        type uint32;
        description
        "The ID that identifies the NRP used for
        backup paths that protect primary paths
        setup over a specific NRP.";
    }
}

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 allocated to an NRP
            - specified as percentage of link
capacity.";
        }
    }
}
description
 "The maximum bandwidth on a network resource that
 is allocated for backup traffic - specified as
 percentage of the link capacity.";
 }
 }
 }
 }

 guint nrp-pol-phb {
 description
 "Grouping for NRP-PHB.";
 leaf phb {
 type string;
 description
 "PHB profile identifier.";
 }
 }

 guint nrp-pol-topology {
 description
 "Grouping for NRP topology.";
 container topology {
 description
 "Container for NRP topology.";
 container filters {
 description
 "Container for filters.";
 list filter {
 key "filter-ref";
 description
 "List of filters.";
 leaf filter-ref {
 type nrp-policy-topo-filter-ref;
 description
 "Reference to a specific topology filter from the
 list of global topology filters.";
 }
/*
 * Grouping - Network Resource Partition Policies
 */

grouping nrp-pol {
    description "Grouping for NRP policies.";
    container nrp-policies {
        description "Container for nrp policies.";
        list nrp-policy {
            key "name";
            unique "nrp-id";
            description "List of NRP policies.”;
            leaf name {
                type string;
                description "A string that uniquely identifies the NRP policy.";
            }
            leaf nrp-id {
                type uint32;
                description "A 32-bit ID that uniquely identifies the NRP created by the enforcement of this NRP policy.";
            }
            uses nrp-pol-resource-reservation;
            uses nrp-pol-flow-agg-selector;
            uses nrp-pol-phb;
            uses nrp-pol-topology;
        }
    }
    uses nrp-pol-resource-reservation;
    uses nrp-pol-flow-agg-selector;
    uses nrp-pol-phb;
}

/*
 */
augment "/nw:networks" {
    description
"Augment networks with network resource partition policies."
uses nrp-pol;
}
}
</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.

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-nrp-policy
prefix: sl-pol
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:

* `/networks/nrp-policies`: This subtree specifies the configurations for NRP 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 flow 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:

* `/networks/nrp-policies`: Unauthorized access to this subtree can disclose the NRP policy definitions on the network element.

7. References

7.1. Normative References

[I-D.bestbar-teas-ns-packet]
Beeram, V. P., Saad, T., Gandhi, R., and X. Liu, "YANG Data Model for Topology Filter", draft-bestbar-teas-yang-topology-filter-02 (work in progress), October 2021.


7.2. Informative References


Appendix A. Complete Model Tree Structure

module: ietf-nrp-policy
augment /rw:nw-networks:
  +--rw nrp-policies
    +--rw nrp-policy* [name]
      +--rw name                    string
      +--rw nrp-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-nrp-id?         uint32
      +--rw (backup-bw-type)?
        +--:(backup-bw-value)
          +--rw backup-bandwidth?    uint64
          +--:(backup-bw-percentage)
            +--rw backup-bandwidth-percent?  rt-types:percentage
      +--rw flow-agg-selector
        +--rw mpls
          +--rw (fas-type)?
+--:(label)
  |  +--:(derived)
  |     |  +--rw forwarding-label? empty
  |     +--:(explicit)
  |        +--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
    +--:(fas-type)?
    |    +--:(ipv6-destination)
    |         +--rw destination-prefix* inet:ipv6-prefix
    |         +--:(ipv6-hbh-eh)
    |            +--rw fas-hbh-eh* uint32
    +--rw acl-ref* nrp-policy-acl-ref
  +--rw phb? string
  +--rw topology
  +--rw filters
    +--rw filter* [filter-ref]
      +--rw filter-ref
      |       nrp-policy-topo-filter-ref
    +--rw resource-reservation
      +--rw preference? uint16
      +--:(bw-value)
      |    +--rw maximum-bandwidth? uint64
      |    +--:(bw-percentage)
      |        +--rw maximum-bandwidth-percent? rt-types:percentage
      +--rw shared-resource-groups* uint32
      +--rw protection
      |    +--rw backup-nrp-id?
      |           uint32
      +--rw (backup-bw-type)?
      |    +--:(backup-bw-value)
+-rw backup-bandwidth?
  uint64
  +-:(backup-bw-percentage)
    +-rw backup-bandwidth-percent?
      rt-types:percentage

+-rw flow-agg-selector
  +-rw mpls
    +-rw (fas-type)?
      +-:(label)
        +-rw (specification-type)?
          +-:(derived)
            +-rw forwarding-label?
              empty
          +-:(explicit)
            +-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 (fas-type)?
        +-:(ipv6-destination)
          +-rw destination-prefix*  inet:ipv6-prefix
          +-:(ipv6-hbh-eh)
            +-rw fas-hbh-eh*  uint32
        +-rw acl-ref*  nrp-policy-acl-ref
  +-rw phb?  string
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
IBM Corporation

Email: xufeng.liu.ietf@gmail.com
YANG Data Model for Network Resource Partition Policy
draft-bestbar-teas-yang-nrp-policy-01

Abstract

A Network Resource Partition (NRP) is a collection of resources identified in the underlay network to support services (like IETF Network Slices) that need logical network structures with required characteristics to be created. An NRP policy is a policy construct that enables instantiation of mechanisms in support of service specific control and data plane behaviors on select topological elements associated with the NRP. This document defines a YANG data model for the management of NRP policies on NRP 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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

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

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

Table of Contents

1.  Introduction ............................................. 3
2.1.  Model Usage ........................................ 4
2.2.  Model Structure ..................................... 4
2.3.  NRP Policies .......................................... 5
2.3.1.  Resource Reservation ............................... 5
2.3.2.  Flow Aggregate Selector ............................ 6
2.3.3.  Per-Hop-Behavior .................................. 7
2.3.4.  Topology .......................................... 7
2.4.  YANG Module .......................................... 9
3.  Acknowledgements ........................................ 20
4.  Contributors ............................................. 21
5.  IANA Considerations .................................... 21
6.  Security Considerations ................................ 21
7.  References .............................................. 22
7.1.  Normative References ................................. 22
7.2.  Informative References ............................... 23
Appendix A.  Complete Model Tree Structure .................. 24
1. Introduction

An IETF Network Slice [I-D.ietf-teas-ietf-network-slices] is a service that provides connectivity coupled with a set of specific commitments of network resources between a number of endpoints over a shared underlay network. The IETF Network Slice service is expressed in terms of one or more connectivity constructs. One or more connectivity constructs from one or more IETF Network Slices are mapped to a set of network resources called a Network Resource Partition (NRP). An NRP [I-D.ietf-teas-ietf-network-slices] is a collection of resources identified in the underlay network to support the IETF Network Slice service (or any other service that needs logical network structures with required characteristics to be created). An NRP Policy [I-D.ietf-teas-ns-ip-mpls] is a policy construct that enables instantiation of mechanisms in support of service specific control and data plane behaviors on select topological elements associated with the NRP.

An NRP policy specifies the rules for determining the topology associated with the NRP and dictates how an NRP can be realized in IP/MPLS networks using one of three modes. The NRP 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 NRP policy modes (a) and (c) require the forwarding engine on each NRP capable node to identify the traffic belonging to a specific flow aggregate and to apply the corresponding Per-Hop Behavior (PHB) that determines the forwarding treatment of the packets belonging to the flow aggregate. The identification of the flow aggregate that the packet belongs to and the corresponding forwarding treatment that needs to be applied to the packet is dictated by the NRP policy. When catering to IETF Network Slices, this flow aggregate is referred to as the Slice-Flow Aggregate [I-D.ietf-teas-ns-ip-mpls] and comprises of traffic streams from one or more connectivity constructs (belonging to one or more IETF network slices) mapped to a specific NRP.

The NRP policy modes (b) and (c) require the distributed/centralized resource reservation manager in the control plane to manage NRP resource reservation. The provisions for enabling NRP state aware traffic engineering (NRP-TE) [I-D.ietf-teas-ns-ip-mpls] are dictated by the NRP policy.

This document defines a YANG data model for the management of NRP policies on NRP 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-slices] and [I-D.ietf-teas-ns-ip-mpls].

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.  NRP Policy Data Model

2.1.  Model Usage

A controller that consumes the IETF Network Slice service requests determines which specific connectivity constructs from one or more slices can be grouped together. This could be based on a specific set of SLOs and SLEs, or on any administrative or operational reason. A controller function that has visibility of the underlay network and its resources maps these connectivity constructs onto the NRP. It also constructs and distributes the network wide consistent NRP policy (using the data model defined in this document) to the relevant NRP capable nodes and controllers.

2.2.  Model Structure

The high-level model structure defined by this document is as shown below:
module: ietf-nrp-policy
    augment /nw:networks:
        +-rw nrp-policies
        |  +-rw nrp-policy* [name]
        |     |  +-rw name                    string
        |     |  +-rw nrp-id?                 uint32
        |     |  +-rw resource-reservation
        |     |     |  +  ............
        |     |  +-rw flow-agg-selector
        |     |     |  +  ............
        |     |  +-rw phb?                    string
        |     |  +-rw topology
        |     |     |  +-rw filters
        |     |     |     |  +-rw filter* [filter-ref]
        |     |     |     |     |  +  ............
        |     |     |  +-rw resource-reservation
        |     |     |     |     |  +  ............
        |     |     |  +-rw flow-agg-selector
        |     |     |     |     |  +  ............
        |     |     |  +-rw phb?                    string
        |     |  +-ro filtered-topology
        |     |     |  +  ............

The ‘networks’ container from the ‘ietf-network’ module [RFC8345] provides a placeholder for an inventory of nodes in the network. This container is augmented to carry a set of NRP policies.

2.3. NRP Policies

The ‘nrp-policies’ container carries a list of NRP policies. Each ‘nrp-policy’ entry is identified by a name and holds the set of attributes needed to instantiate the NRP. Each entry also carries an ‘nrp-id’ leaf which uniquely identifies the NRP created by the enforcement of this policy. The key elements of each nrp-policy entry are discussed in the following sub-sections.

2.3.1. Resource Reservation

The ‘resource-reservation’ container carries data nodes that are used to support NRP state aware bandwidth engineering. The data nodes in this container facilitate preference-based preemption of NRP state aware TE paths, sharing of resources amongst a group of NRPs and backup 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-nrp-id?                    uint32
    +-rw (backup-bw-type)?
      +-:(backup-bw-value)
        |- +=rw backup-bandwidth?           uint64
      +-:(backup-bw-percentage)
        +=rw backup-bandwidth-percent?    rt-types:percentage

2.3.2. Flow Aggregate Selector

The 'flow-agg-selector' container carries data nodes that specify the rules for identifying which packets belong to the flow aggregate that this NRP caters to.
++-rw flow-agg-selector
  +--rw mpls
     +--rw (fas-type)?
        +--:(label)
           |  +--rw (specification-type)?
           |     +--:(derived)
           |     |  +--rw forwarding-label? empty
           |     +--:(explicit)
           |        +--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 (fas-type)?
        +--:(ipv6-destination)
           | +--rw destination-prefix* inet:ipv6-prefix
           +--:(ipv6-hbh-eh)
              +--rw fas-hbh-eh* uint32
              +--rw acl-ref* nrp-policy-acl-ref

2.3.3. Per-Hop-Behavior

The ‘phb’ leaf carries a name of a PHB profile available on the
topological element where the policy is being enforced.

  +--rw phb? string

2.3.4. Topology

The ‘topology’ container consists of a list of filters where each
element references a topology filter
[I-D.bestbar-teas-yang-topology-filter]. The topological elements
that satisfy the membership criteria can optionally override the
default resource-reservation, flow-agg-selector and phb specific
leaves. The ‘topology’ container also consists of a read-only
reference to the resultant filtered topology formed from the union of
the specified filters.

```
++-rw topology
  ++-rw filters
    ++-rw filter* [filter-ref]
      ++-rw filter-ref
        nrp-policy-topo-filter-ref
    ++-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-nrp-id?
      | uint32
    ++-rw (backup-bw-type)?
      | +--:(backup-bw-value)
      |     ++-rw backup-bandwidth? uint64
      |     +--:(backup-bw-percentage)
      |        ++-rw backup-bandwidth-percent? rt-types:percentage
  ++-rw flow-agg-selector
    ++-rw mpls
      ++-rw (fas-type)?
      | +--:(label)
      |    ++-rw (specification-type)?
      |    | +--:(derived)
      |    | | ++-rw forwarding-label?
      |    |        empty
      |    | +--:(explicit)
      |    | | ++-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?
```
2.4. YANG Module

<CODE BEGINS> file "ietf-nrp-policy@2022-07-11.yang"
module ietf-nrp-policy {
  yang-version 1.1;
  prefix nrp-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-network {
    prefix nw;
}
import ietf-network-topology {
    prefix nt;
    reference
        "RFC 8345: A YANG Data Model for Network Topologies";
}

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

import ietf-topology-filter {
    prefix topo-filt;
    reference
        "draft-bestbar-teas-yang-topology-filter: YANG Data Model for Topology Filter";
}

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>
This YANG module defines a data model for managing Network Resource Partition Policies on Network Resource Partition capable nodes and controllers.

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

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

This version of this YANG module is part of RFC XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.

revision 2022-07-11 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Data Model for Network Resource Partition Policies.";
}

/*
  * IDENTITIES
  */

/*
  * Identity:
  * MPLS Flow Aggregate Selector (FAS) Label Position Type.
  */

identity fas-mpls-label-position-type {
  description
    "Base identity for the position of the MPLS FAS label.";
}

identity fas-mpls-label-position-top {
  base fas-mpls-label-position-type;
  description
    "MPLS FAS label is at the top of the label stack.";
}
identity fas-mpls-label-position-bottom {
    base fas-mpls-label-position-type;
    description
        "MPLS FAS label is either at the bottom or at a specific offset from the bottom of the label stack.";
}

identity fas-mpls-label-position-indicator {
    base fas-mpls-label-position-type;
    description
        "MPLS FAS is preceded by a special purpose indicator label in the label stack.";
}

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

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

typedef nrp-policy-topo-filter-ref {
    type leafref {
        path "/nw:networks/topo-filt:topology-filters/
            + "topo-filt:topology-filter/topo-filt:name";
    }
    description
        "This type is used to reference a Topology Filter.";
}

typedef nrp-policy-topo-network-ref {
    type leafref {
        path "/nw:networks/nw:network/nw:network-id";
    }
    description
        "This type is used to reference a network.";
}

typedef nrp-policy-topo-node-ref {
    type leafref {
        path "/nw:networks/nw:network/nw:node/
            + "nw:node-id";
    }
typedef nrp-policy-topo-link-ref {
    type leafref {
        path "/nw:networks/nw:network/nt:link/" + "nt:link-id";
    }
    description
        "This type is used to reference a link.";
}

/* */
* G R O U P I N G S */
/* */
/* Grouping - MPLS FAS label location specific fields */
/* */
grouping nrp-pol-fas-mpls-label-location {
    description
        "Grouping for MPLS FAS label location specific fields.";
    leaf label-position {
        type identityref {
            base fas-mpls-label-position-type;
        }
        description
            "MPLS FAS label position.";
    }
    leaf label-position-offset {
        when "derived-from-or-self(../label-position," + "'nrp-pol:fas-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 - Flow-Aggregate Selector (FAS) */
/* */
grouping nrp-pol-flow-agg-selector {
description
"Grouping for Flow-Aggregate Selector (FAS).";
container flow-agg-selector {
    description
    "Container for FAS.";
    container mpls {
        description
        "Container for MPLS FAS.";
        choice fas-type {
            description
            "Choices for MPLS FAS.";
            case label {
                choice specification-type {
                    description
                    "Choices for MPLS label specification.";
                    case derived {
                        leaf forwarding-label {
                            type empty;
                            description
                            "MPLS FAS Label is derived from forwarding label.";
                        }
                    }
                    case explicit {
                        leaf label {
                            type rt-types:mpls-label;
                            description
                            "MPLS FAS Label is explicitly specified.";
                        }
                    }
                    uses nrp-pol-fas-mpls-label-location;
                }
            }
            case label-ranges {
                list label-range {
                    key "index";
                    unique "start-label end-label";
                    description
                    "Any label from the specified set of MPLS label ranges can be used as the FAS.";
                    leaf index {
                        type string;
                        description
                        "A string that uniquely identifies a label range.";
                    }
                    leaf start-label {
                        type rt-types:mpls-label;
                        description
                        "MPLS FAS Label is explicitly specified.";
                    }
                    uses nrp-pol-fas-mpls-label-location;
                }
            }
        }
    }
}
}
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 nrp-pol-fas-mpls-label-location;
}
}
}
container ipv4 {

description
  "Container for IPv4 FAS.";
leaf-list destination-prefix {
  type inet:ipv4-prefix;
  description
    "Any prefix from the specified set of IPv4
destination prefixes can be the FAS.";
}
}
container ipv6 {

description
  "Container for IPv6 FAS.";
choice fas-type {
  description
    "Choices for IPv6 FAS.";
  case ipv6-destination {
    leaf-list destination-prefix {
      type inet:ipv6-prefix;
      description
        "Any prefix from the specified set of IPv6
destination prefixes can be the FAS.";
    }
  }
}
}
```
case ipv6-hbh-eh {
    leaf-list fas-hbh-eh {
        type uint32;
        description
            "Set of FAS values carried in Hop-by-Hop
             Option of IPv6 extension header.";
    }
}
}
leaf-list acl-ref {
    type nrp-policy-acl-ref;
    description
        "Flow Aggregate selection is done based on the
         specified list of ACLs.";
}
}
/
* Grouping - NRP Policy Resource Reservation
*/
grouping nrp-pol-resource-reservation {
    description
        "Grouping for NRP policy resource reservation.";
    container resource-reservation {
        description
            "Container for NRP policy resource reservation.";
        leaf preference {
            type uint16;
            description
                "Control plane preference for the corresponding
                 Network Resource Partition (NRP). 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 an NRP
                         - specified as absolute value.";
                }
            }
        }
    }
}
```
case bw-percentage {
  leaf maximum-bandwidth-percent {
    type rt-types:percentage;
    description
    "The maximum bandwidth allocated to an NRP
    - specified as percentage of link
capacity.";
  }
}
}  
leaf-list shared-resource-groups {
  type uint32;
  description
  "List of shared resource groups that an NRP
  shares its allocated resources with.";
}
}  
container protection {
  description
  "Container for NRP protection reservation.";
leaf backup-nrp-id {
  type uint32;
  description
  "The ID that identifies the NRP used for
  backup paths that protect primary paths
  setup over a specific NRP.";
}
}  
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 nrp-pol-phb {
    description "Grouping for NRP-PHB."
    leaf phb {
        type string;
        description "PHB profile identifier.";
    }
}

/*
* Grouping - NRP policy - Topology
*/

grouping nrp-pol-topology {
    description "Grouping for NRP topology."
    container topology {
        description "Container for NRP topology."
        container filters {
            description "Container for filters."
            list filter {
                key "filter-ref";
                description "List of filters."
                leaf filter-ref {
                    type nrp-policy-topo-filter-ref;
                    description "Reference to a specific topology filter from the list of global topology filters."
                }
            }
        }
        uses nrp-pol-resource-reservation;
        uses nrp-pol-flow-agg-selector;
        uses nrp-pol-phb;
    }
    config false;
}

/*
* Grouping - NRP policy - PHB (NRP-PHB)
*/
description
"Container for filtered topology."
choice filtered-topo-type { description
"Choices for filtered topology.";
case network {
  list network {
    key "network-ref";
    description
"List of networks.";
    leaf network-ref {
      type nrp-policy-topo-network-ref;
      description
"Reference to a specific network.";
    }
  }
}
}
choice network-elements { case network-elements {
  list node {
    key "node-ref";
    description
"List of nodes.";
    leaf node-ref {
      type nrp-policy-topo-node-ref;
      description
"Reference to a specific node.";
    }
  }
  list link {
    key "link-ref";
    description
"List of links.";
    leaf link-ref {
      type nrp-policy-topo-link-ref;
      description
"Reference to a specific link.";
    }
  }
}
/*
 * Grouping - Network Resource Partition Policies
 */
grouping nrp-pol {
    description
        "Grouping for NRP policies.";
    container nrp-policies {
        description
            "Container for nrp policies.";
    list nrp-policy {
        key "name";
        unique "nrp-id";
        description
            "List of NRP policies.";
        leaf name {
            type string;
            description
                "A string that uniquely identifies the NRP policy.";
        }
        leaf nrp-id {
            type uint32;
            description
                "A 32-bit ID that uniquely identifies the NRP
                created by the enforcement of this NRP policy.";
        }
        uses nrp-pol-resource-reservation;
        uses nrp-pol-flow-agg-selector;
        uses nrp-pol-phb;
        uses nrp-pol-topology;
    }
} 

/*
 */
augment "/nw:networks" {
    description
        "Augment networks with network resource partition
        policies.";
    uses nrp-pol;
}

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.

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-nrp-policy
prefix: nrp-pol
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:

* "/networks/nrp-policies": This subtree specifies the configurations for NRP 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 flow 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:

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

7. References

7.1. Normative References

[I-D.bestbar-teas-yang-topology-filter]


7.2. Informative References

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

[I-D.ietf-teas-ns-ip-mpls]
Saad, T., Beeram, V. P., Dong, J., Wen, B., Ceccarelli, D., Halpern, J., Peng, S., Chen, R., Liu, X., Contreras,
Appendix A. Complete Model Tree Structure

module: ietf-nrp-policy

augment /nw:networks:
  +--rw nrp-policies
    +--rw nrp-policy* [name]
      +--rw name                    string
      +--rw nrp-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-nrp-id?         uint32
      +--rw (backup-bw-type)?
        |    +--:(backup-bw-value)
        |     |    +--rw backup-bandwidth?   uint64
        |     +--:(backup-bw-percentage)
        |          +--rw backup-bandwidth-percent?
                        rt-types:percentage
    +--rw flow-agg-selector
      +--rw mpls
        +--rw (fas-type)?
          |    +--:(label)
          |     |    +--rw (specification-type)?
          |     |     +--:(derived)
          |     |     |    +--rw forwarding-label?  empty
          |     |     +--:(explicit)
          |     |          +--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 (fas-type)?
   +--:(ipv6-destination)
     +--rw destination-prefix*  inet:ipv4-prefix
     +--:(ipv6-hbh-eh)
       +--rw fas-hbh-eh*  uint32

---rw acl-ref*  nrp-policy-acl-ref
---rw phb?  string
---rw topology

---rw filters
   +--rw filter* [filter-ref]
     +--rw filter-ref
       |  nrp-policy-topo-filter-ref
     +--rw resource-reservation
       +--rw preference?  uint16
       +--:(bw-value)
         |  +--rw maximum-bandwidth?  uint64
         +--:(bw-percentage)
           +--rw maximum-bandwidth-percent?  rt-types:percentage
     +--rw shared-resource-groups*  uint32
   +--rw protection
     +--rw backup-nrp-id?  uint32
     +--rw (backup-bw-type)?
       +--:(backup-bw-value)
         |  +--rw backup-bandwidth?  uint64
         +--:(backup-bw-percentage)
           +--rw backup-bandwidth-percent?  rt-types:percentage

---rw flow-agg-selector
   +--rw mpls
     +--rw (fas-type)?
       +--:(label)
Internet-Draft NRP Policy YANG Data Model July 2022

++-rw (specification-type)?
  +-:(derived)
  |  +-rw forwarding-label?
  |     empty
  +-:(explicit)
    +-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 (fas-type)?
        +-:(ipv6-destination)
          |  +-rw destination-prefix* inet:ipv6-prefix
          +-:(ipv6-hbh-eh)
            |  +-rw fas-hbh-eh* uint32
            |  +-rw acl-ref* nrp-policy-acl-ref
          +-rw phb? string
    +-ro filtered-topology
    +-ro (filtered-topo-type)?
      +-:(network)
        +-ro network* [network-ref]
          +-ro network-ref
            |  nrp-policy-topo-network-ref
      +-:(network-elements)
        +-ro node* [node-ref]
          |  +-ro node-ref nrp-policy-topo-node-ref
        +-ro link* [link-ref]
          |  +-ro link-ref nrp-policy-topo-link-ref

Authors’ Addresses

Updated Common YANG Data Types for Traffic Engineering
draft-busi-teas-te-types-update-01

Abstract

This document defines few additional common data types and groupings in YANG data modeling language to be imported by modules that model Traffic Engineering (TE) configuration and state capabilities.

This document updates RFC 8776 with a new revision of the module ietf-te-types.

Status of This Memo

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

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

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

This Internet-Draft will expire on 8 September 2022.
1. Introduction

After the publication of [RFC8776], the need to add a new typedef and a new grouping to ietf-te-types YANG module has arisen.

These definitions have been developed in [I-D.ietf-teas-yang-te] and [I-D.ietf-teas-yang-l3-te-topo] and are quite mature: [I-D.ietf-teas-yang-te] in particular is ready from WG Last Call.

However, these definitions have broader applicability than the I-D where they have originated, so it makes sense to move them within the ietf-te-types YANG module.
1.1. Options considered

The concern is how to be able to update the ietf-te-types YANG module published in [RFC8776] without delaying too much the progress of the mature WG documents.

Three possible options have been identified to address this concern.

One option is to keep these definitions in the YANG modules where they have initially been defined: other YANG modules can still import them. The drawback of this approach is that it defeating the value of common YANG modules like ietf-te-types since common definitions will be spread around multiple specific YANG modules.

A second option is to define them in a new common YANG module (e.g., ietf-te-types-ext). The drawback of this approach is that it will increase the number of YANG modules providing thiny updates to the ietf-te-types YANG module.

A third option is to develop a revision of the ietf-te-types YANG module within an RFC8776-bis. The drawback of this approach is that the process for developing a big RFC8776-bis just for a thiny update is too high. Moreover, it is not clear what could be done with the ietf-te-packet-types YANG module which is also defined in [RFC8776] but it does not need to be revised.

This document explores an alternative option to just update [RFC8776] with a new revision of the module ietf-te-types.

In order to focus the review process of this document only to the changes proposed by this document: - Section 2 describes only the updates to the ietf-te-types YANG module proposed by this document; - Section 3 defines only the diff between the revision of the ietf-te-types YANG module proposed in this document and the revision of the ietf-te-types YANG module published in [RFC8776].

In order to allow all the YANG toolchain to keep working by extracting the revision of the ietf-te-types YANG module proposed in this document, this revision is provided by Appendix A. This text is intended not to be subject to the review of this document.

1.2. Requirements Notation

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.
1.3. Terminology

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

1.4. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects, added to the ietf-te-types YANG module do not need to be prefixed.

The revision of the ietf-te-types YANG module uses the prefixes defined in section 1.2 of [RFC8776].

2. Overview

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

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

encoding-and-switching-type  This is a common grouping to define the LSP encoding and switching types.

3. TE Types YANG Update

This section provides the diff between the YANG module in section 3.1 of [RFC8776] and the YANG module revision in Appendix A.

Note - This diff has been generated using the following UNIX commands to compare the YANG module revisions in section 3.1 of [RFC8776] and in Appendix A:

diff ietf-te-types@2020-06-10.yang ietf-te-types.yang > model-diff.txt
sed 's/^/    /' model-diff.txt > model-diff-spaces.txt
sed 's/^    >   /    >   /' model-diff-spaces.txt > model-updates.txt

The output (model-updates.txt) is reported here:

55c55
<      Copyright (c) 2020 IETF Trust and the persons identified as
---
>      Copyright (c) 2022 IETF Trust and the persons identified as
65c65
<      This version of this YANG module is part of RFC 8776; see the
---
>      This version of this YANG module is part of RFC XXXX; see the
67a68,74
revision 2022-03-07 {
  description
  "Latest revision of TE types.";
  reference
  "RFC XXXX: Updated Common YANG Data Types for Traffic Engineering";
}

typedef bandwidth-scientific-notation {
  type string {
    pattern
    '0(\.[0-9]?)([eE](\+)?[0-9])?' \\
    + '[1-9]\([.][0-9]{0,6}\)[eE](\+)?([9][0-9]|\[1-8][0-9]|[0?][0-9])?';
  }
  units "bps";
  description
  "Bandwidth values, expressed using the scientific notation in bits per second. The encoding format is the external decimal-significant character sequences specified in IEEE 754 and ISO/IEC C99 for 32-bit decimal floating-point numbers:
  \((-1)\text{**}(S) * 10\text{**}(Exponent) \times (Significant)\),
  where Significant uses 7 digits.
  An implementation for this representation may use decimal32 or binary32. The range of the Exponent is from -95 to +96 for decimal32, and from -38 to +38 for binary32.
  As a bandwidth value, the format is restricted to be normalized, non-negative, and non-fraction:
  n.dddddde{+}dd, N.DDDDDDE{+}DD, 0e0 or 0E0,
  where 'd' and 'D' are decimal digits; 'n' and 'N' are non-zero decimal digits; 'e' and 'E' indicate a power of ten.
  Some examples are 0e0, 1e10, and 9.953e9.";
  reference
  ISO/IEC C99: Information Technology - Programming Languages - C.";
}

grouping encoding-and-switching-type {
  description
  "Common grouping to define the LSP encoding and switching types";
  leaf encoding {
4. IANA Considerations

This document updates the ietf-te-types YANG module registered by [RFC8776].

Therefore this document does not require any IANA actions.

5. Security Considerations

The security considerations defined in section 7 of [RFC8776] applies to the revision of the ietf-te-types YANG module.

This document just adds new typedefs and groupings to the YANG modules defined in [RFC8776] and therefore it does not introduce additional considerations.

6. References

6.1. Normative References


6.2. Informative References

[I-D.ietf-teas-yang-l3-te-topo]
Liu, X., Bryskin, I., Beeram, V. P., Saad, T., Shah, H.,
and O. G. D. Dios, "YANG Data Model for Layer 3 TE
Topologies", Work in Progress, Internet-Draft, draft-ietf-
teas-yang-l3-te-topo-12, 24 October 2021,
<https://www.ietf.org/archive/id/draft-ietf-teas-yang-l3-
te-topo-12.txt>.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., Bryskin, I.,
and O. G. D. Dios, "A YANG Data Model for Traffic
Engineering Tunnels, Label Switched Paths and Interfaces",
Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-
29, 7 February 2022, <https://www.ietf.org/archive/id/
draft-ietf-teas-yang-te-29.txt>.

Appendix A. TE Types YANG Module

<CODE BEGINS> file "ietf-te-types@2022-03-07.yang"
module ietf-te-types {
   yang-version 1.1;
   prefix te-types;

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

Busi, et al. Expires 8 September 2022
"RFC 8294: Common YANG Data Types for the Routing Area";
}

organization
"IETF Traffic Engineering Architecture and Signaling (TEAS)
Working Group";
contact
"WG Web:  <https://datatracker.ietf.org/wg/teas/>
WG List:  <mailto:teas@ietf.org>
Editor:  Tarek Saad
<mailto:tsaad@juniper.net>
Editor:  Rakesh Gandhi
<mailto:rgandhi@cisco.com>
Editor:  Vishnu Pavan Beeram
<mailto:vbeeram@juniper.net>
Editor:  Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>
Editor:  Igor Bryskin
<mailto:i_bryskin@yahoo.com>"

description
"This YANG module contains a collection of generally useful
YANG data type definitions specific to TE. The model fully
conforms to the Network Management Datastore Architecture
(NMDA).

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.

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

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject to
the license terms contained in, the Simplified BSD License set
forth in Section 4.c of the IETF Trust’s Legal Provisions
Relating to IETF Documents

This version of this YANG module is part of RFC XXXX; see the
RFC itself for full legal notices.";
typedef admin-group {
  type yang:hex-string {
    /* 01:02:03:04 */
    length "1..11";
  }
  description
    "Administrative group / resource class / color representation in 'hex-string' type. The most significant byte in the hex-string is the farthest to the left in the byte sequence. Leading zero bytes in the configured value may be omitted for brevity.";
  reference
    "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2
    RFC 5305: IS-IS Extensions for Traffic Engineering
    RFC 7308: Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)";
}

typedef admin-groups {
  type union {
    type admin-group;
    type extended-admin-group;
  }
  description
    "Derived types for TE administrative groups.";
}

typedef extended-admin-group {
  type yang:hex-string;
description
"Extended administrative group / resource class / color representation in 'hex-string' type.
The most significant byte in the hex-string is the farthest to the left in the byte sequence. Leading zero bytes in the configured value may be omitted for brevity."
reference
"RFC 7308: Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)"

typedef path-attribute-flags {
type union {
  type identityref {
    base session-attributes-flags;
  }
  type identityref {
    base lsp-attributes-flags;
  }
}
description
"Path attributes flags type."
}

typedef performance-metrics-normality {
type enumeration {
  enum unknown {
    value 0;
    description
    "Unknown.";
  }
  enum normal {
    value 1;
    description
    "Normal. Indicates that the anomalous bit is not set.";
  }
  enum abnormal {
    value 2;
    description
    "Abnormal. Indicates that the anomalous bit is set.";
  }
}
description
"Indicates whether a performance metric is normal (anomalous bit not set), abnormal (anomalous bit set), or unknown."
reference
"RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
RFC 7823: Performance-Based Path Selection for Explicitly
typedef srlg {
    type uint32;
    description "SRLG type.";
    reference "RFC 4203: OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)"
    RFC 5307: IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)";
}

typedef te-common-status {
    type enumeration {
        enum up {
            description "Enabled.";
        }
        enum down {
            description "Disabled.";
        }
        enum testing {
            description "In some test mode.";
        }
        enum preparing-maintenance {
            description "The resource is disabled in the control plane to prepare for a graceful shutdown for maintenance purposes.";
            reference "RFC 5817: Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks";
        }
        enum maintenance {
            description "The resource is disabled in the data plane for maintenance purposes.";
        }
        enum unknown {
            description "Status is unknown.";
        }
    }
}
typedef te-bandwidth {
    type string {
        pattern '0[xX]0(\((\.(0)?[pP]\+(0)?(\.(0)?))|' 
            + '1(\.([da-fA-F]{0,5}[02468acce]?))\)\)|' 
            + '[pP]\+(12[0-7])'| 
            + '1[01]\d|0?\d?\d?)|0[xX]\([da-fA-F]{1,8}|\d+\) |' 
            + ',(0[xX](0(\.(0)?[pP]\+(0)?(\.(0)?))|' 
            + '\([da-fA-F]{0,5}[02468acce]?)\)\)|' 
            + '[pP]\+(12[0-7])'| 
            + '1[01]\d|0?\d?\d?)|0[xX]\([da-fA-F]{1,8}|\d+\))*';
    } 
    description 
    "This is the generic bandwidth type. It is a string containing 
a list of numbers separated by commas, where each of these 
numbers can be non-negative decimal, hex integer, or 
hex float:

    (dec | hex | float)["\', (dec | hex | float)]"

For the packet-switching type, the string encoding follows
the type 'bandwidth-ieee-float32' as defined in RFC 8294 
(e.g., 0x1p10), where the units are in bytes per second.

For the Optical Transport Network (OTN) switching type,
a list of integers can be used, such as '0,2,3,1', indicating 
two ODU0s and one ODU3. ('ODU' stands for 'Optical Data 
Unit'.) For Dense Wavelength Division Multiplexing (DWDM),
a list of pairs of slot numbers and widths can be used,
such as '0,2,3,3', indicating a frequency slot 0 with 
slot width 2 and a frequency slot 3 with slot width 3.
Canonically, the string is represented as all lowercase and in 
hex, where the prefix '0x' precedes the hex number."

reference 
"RFC 8294: Common YANG Data Types for the Routing Area 
ITU-T Recommendation G.709: Interfaces for the 
optical transport network";
}

typedef te-ds-class {
    type uint8 {
        range "0..7";
    } 
    description
"The Differentiated Services Class-Type of traffic.";
reference
"RFC 4124: Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering, Section 4.3.1";
}
typedef te-global-id {
type uint32;
description
"An identifier to uniquely identify an operator, which can be either a provider or a client. The definition of this type is taken from RFCs 6370 and 5003. This attribute type is used solely to provide a globally unique context for TE topologies.";
reference
"RFC 5003: Attachment Individual Identifier (AII) Types for Aggregation
RFC 6370: MPLS Transport Profile (MPLS-TP) Identifiers";
}
typedef te-hop-type {
type enumeration {
enum loose {
description
"A loose hop in an explicit path.";
}
enum strict {
description
"A strict hop in an explicit path.";
}
}
description
"Enumerated type for specifying loose or strict paths.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 4.3.3";
}
typedef te-link-access-type {
type enumeration {
enum point-to-point {
description
"The link is point-to-point.";
}
enum multi-access {
description
"The link is multi-access, including broadcast and NBMA.";
}
}
typedef te-label-direction {
    type enumeration {
        enum forward {
            description
            "Label allocated for the forward LSP direction.";
        }
        enum reverse {
            description
            "Label allocated for the reverse LSP direction.";
        }
    }
    description
    "Enumerated type for specifying the forward or reverse label.";
}

typedef te-link-direction {
    type enumeration {
        enum incoming {
            description
            "The explicit route represents an incoming link on a node.";
        }
        enum outgoing {
            description
            "The explicit route represents an outgoing link on a node.";
        }
    }
    description
    "Enumerated type for specifying the direction of a link on a node.";
}

typedef te-metric {
    type uint32;
    description
    "TE metric.";
    reference
    "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
typedef te-node-id {
    type yang:dotted-quad;
    description
    "A type representing the identifier for a node in a TE topology.
    The identifier is represented as 4 octets in dotted-quad notation.
    This attribute MAY be mapped to the Router Address TLV described in Section 2.4.1 of RFC 3630, the TE Router ID described in Section 3 of RFC 6827, the Traffic Engineering Router ID TLV described in Section 4.3 of RFC 5305, or the TE Router ID TLV described in Section 3.2.1 of RFC 6119. The reachability of such a TE node MAY be achieved by a mechanism such as that described in Section 6.2 of RFC 6827.";
    reference
    "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2, Section 2.4.1
RFC 5305: IS-IS Extensions for Traffic Engineering, Section 4.3
RFC 6119: IPv6 Traffic Engineering in IS-IS, Section 3.2.1
RFC 6827: Automatically Switched Optical Network (ASON) Routing for OSPFv2 Protocols, Section 3";
}

typedef te-oper-status {
    type te-common-status;
    description
    "Defines a type representing the operational status of a TE resource.";
}

typedef te-admin-status {
    type te-common-status;
    description
    "Defines a type representing the administrative status of a TE resource.";
}

typedef te-path-disjointness {
    type bits {
        bit node {
            position 0;
            description
            "Node disjoint.";
        }
    }
}
typedef te-recovery-status {
  type enumeration {
    enum normal {
      description
      "Both the recovery span and the working span are fully
allocated and active, data traffic is being
transported over (or selected from) the working
span, and no trigger events are reported.";
    }
    enum recovery-started {
      description
      "The recovery action has been started but not completed.";
    }
    enum recovery-succeeded {
      description
      "The recovery action has succeeded. The working span has
reported a failure/degrade condition, and the user traffic
is being transported (or selected) on the recovery span.";
    }
    enum recovery-failed {
      description
      "The recovery action has failed.";
    }
    enum reversion-started {
      description
      "The reversion has started.";
    }
    enum reversion-succeeded {
      description
      "The reversion action has succeeded.";
    }
  }
}
enum reversion-failed {
    description
    "The reversion has failed.";
}

enum recovery-unavailable {
    description
    "The recovery is unavailable, as a result of either an
    operator’s lockout command or a failure condition
detected on the recovery span.";
}

enum recovery-admin {
    description
    "The operator has issued a command to switch the user
traffic to the recovery span.";
}

enum wait-to-restore {
    description
    "The recovery domain is recovering from a failure/degrade
condition on the working span that is being controlled by
the Wait-to-Restore (WTR) timer.";
}

description
"Defines the status of a recovery action.";

reference
"RFC 4427: Recovery (Protection and Restoration) Terminology
for Generalized Multi-Protocol Label Switching (GMPLS)
RFC 6378: MPLS Transport Profile (MPLS-TP) Linear Protection";

typedef te-template-name {
    type string {
        pattern '/?([a-zA-Z0-9\-_]+)(/[a-zA-Z0-9\-_]+)*';
    }
    description
    "A type for the name of a TE node template or TE link
    template.";
}

typedef te-topology-event-type {
    type enumeration {
        enum add {
            value 0;
            description
            "A TE node or TE link has been added.";
        }
        enum remove {

typedef te-topology-id {
  type union {
    type string {
      length "0";
      // empty string
    }
    type string {
      pattern '([a-zA-Z0-9\-_\.]+):)*'  
      + '/?([a-zA-Z0-9\-_\.]+)(/[a-zA-Z0-9\-_\.]+)*';
    }
  }
  description
  "An identifier for a topology. It is optional to have one or more
  prefixes at the beginning, separated by colons. The prefixes can be
  'network-types' as defined in the 'ietf-network' module in RFC 8345,
  to help the user better understand the topology before further
  inquiry is made.";
  reference
  "RFC 8345: A YANG Data Model for Network Topologies";
}

typedef te-tp-id {
  type union {
    type uint32;
    // Unnumbered
    type inet:ip-address;
    // IPv4 or IPv6 address
  }
  description
  "An identifier for a TE link endpoint on a node. This attribute is
  mapped to a local or remote link identifier as defined in RFCs 3630
  and 5305.";
  reference

typedef bandwidth-scientific-notation {
  type string {
    pattern
      '0\.0?([eE]\+0?)?|'
      + '[1-9]\.\[0-9\]{0,6})?([eE]\+9[0-6]|1-8[0-9]|0?[0-9])?';
  } units "bps";
  description
    "Bandwidth values, expressed using the scientific notation
    in bits per second.
    The encoding format is the external decimal-significant
    character sequences specified in IEEE 754 and ISO/IEC C99
    for 32-bit decimal floating-point numbers:
    (-1)**(S) * 10**(Exponent) * (Significant),
    where Significant uses 7 digits.
    An implementation for this representation may use decimal32
    or binary32. The range of the Exponent is from -95 to +96
    for decimal32, and from -38 to +38 for binary32.
    As a bandwidth value, the format is restricted to be
    normalized, non-negative, and non-fraction:
    n.ddddde{+}dd, N.DDDDE{+}DD, 0e0 or 0E0,
    where 'd' and 'D' are decimal digits; 'n' and 'N' are
    non-zero decimal digits; 'e' and 'E' indicate a power of ten.
    Some examples are 0e0, 1e10, and 9.953e9.";
  reference
    "IEEE Std 754-2008: IEEE Standard for Floating-Point
    Arithmetic.
    ISO/IEC C99: Information technology - Programming
    Languages - C.";
}

/* TE features */

feature p2mp-te {
  description
    "Indicates support for Point-to-Multipoint TE (P2MP-TE).";
  reference
    "RFC 4875: Extensions to Resource Reservation Protocol -
    Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
    Label Switched Paths (LSPs)";
}

feature frr-te {
description
"Indicates support for TE Fast Reroute (FRR).";
reference
"RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}

feature extended-admin-groups {
  description
  "Indicates support for TE link extended administrative groups.";
  reference
  "RFC 7308: Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)";
}

feature named-path-affinities {
  description
  "Indicates support for named path affinities.";
}

feature named-extended-admin-groups {
  description
  "Indicates support for named extended administrative groups.";
}

feature named-srlg-groups {
  description
  "Indicates support for named SRLG groups.";
}

feature named-path-constraints {
  description
  "Indicates support for named path constraints.";
}

feature path-optimization-metric {
  description
  "Indicates support for path optimization metrics.";
}

feature path-optimization-objective-function {
  description
  "Indicates support for path optimization objective functions.";
}

/*
 * Identities
 */
identity session-attributes-flags {
    description
    "Base identity for the RSVP-TE session attributes flags.";
}

identity local-protection-desired {
    base session-attributes-flags;
    description
    "Local protection is desired.";
    reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 4.7.1";
}

identity se-style-desired {
    base session-attributes-flags;
    description
    "Shared explicit style, to allow the LSP to be established and share resources with the old LSP.";
    reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity local-recording-desired {
    base session-attributes-flags;
    description
    "Label recording is desired.";
    reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 4.7.1";
}

identity bandwidth-protection-desired {
    base session-attributes-flags;
    description
    "Requests FRR bandwidth protection on LSRs, if present.";
    reference
    "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}

identity node-protection-desired {
    base session-attributes-flags;
    description
    "Requests FRR node protection on LSRs, if present.";
    reference
    "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}
identity path-reevaluation-request {
    base session-attributes-flags;
    description
        "This flag indicates that a path re-evaluation (of the
        current path in use) is requested. Note that this does
        not trigger any LSP reroutes but instead just signals a
        request to evaluate whether a preferable path exists.";
    reference
        "RFC 4736: Reoptimization of Multiprotocol Label Switching
        (MPLS) Traffic Engineering (TE) Loosely Routed Label Switched
        Path (LSP)";
}

identity soft-preemption-desired {
    base session-attributes-flags;
    description
        "Soft preemption of LSP resources is desired.";
    reference
        "RFC 5712: MPLS Traffic Engineering Soft Preemption";
}

identity lsp-attributes-flags {
    description
        "Base identity for LSP attributes flags.";
}

identity end-to-end-rerouting-desired {
    base lsp-attributes-flags;
    description
        "Indicates end-to-end rerouting behavior for an LSP
        undergoing establishment. This MAY also be used to
        specify the behavior of end-to-end LSP recovery for
        established LSPs.";
    reference
        "RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS
        RSVP-TE
        RFC 5420: Encoding of Attributes for MPLS LSP Establishment
        Using Resource Reservation Protocol Traffic Engineering
        (RSVP-TE)
        RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
        Route Object (ERO)";
}

identity boundary-rerouting-desired {
    base lsp-attributes-flags;
    description
        "Indicates boundary rerouting behavior for an LSP undergoing
        establishment. This MAY also be used to specify
segment-based LSP recovery through nested crankback for established LSPs. The boundary Area Border Router (ABR) / Autonomous System Border Router (ASBR) can decide to forward the PathErr message upstream to either an upstream boundary ABR/ASBR or the ingress LSR. Alternatively, it can try to select another egress boundary LSR.

reference
"RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"

identity segment-based-rerouting-desired {
  base lsp-attributes-flags;
  description
  "Indicates segment-based rerouting behavior for an LSP undergoing establishment. This MAY also be used to specify segment-based LSP recovery for established LSPs."
  reference
  "RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE
  RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"
}

identity lsp-integrity-required {
  base lsp-attributes-flags;
  description
  "Indicates that LSP integrity is required."
  reference
  "RFC 4875: Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)
  RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"
}

identity contiguous-lsp-desired {
  base lsp-attributes-flags;
  description
  "Indicates that a contiguous LSP is desired."
reference
"RFC 5150: Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";

identity lsp-stitching-desired {
    base lsp-attributes-flags;
    description
    "Indicates that LSP stitching is desired.";
    reference
    "RFC 5150: Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";
}

identity pre-planned-lsp-flag {
    base lsp-attributes-flags;
    description
    "Indicates that the LSP MUST be provisioned in the control plane only.";
    reference
    "RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";
}

identity non-php-behavior-flag {
    base lsp-attributes-flags;
    description
    "Indicates that non-PHP (non-Penultimate Hop Popping) behavior for the LSP is desired.";
    reference
    "RFC 6511: Non-Penultimate Hop Popping Behavior and Out-of-Band Mapping for RSVP-TE Label Switched Paths
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";
}

identity oob-mapping-flag {
    base lsp-attributes-flags;
    description
    "Indicates that signaling of the egress binding information is out of band (e.g., via the Border Gateway Protocol (BGP)).";
identity entropy-label-capability {
  base lsp-attributes-flags;
  description
    "Indicates entropy label capability.";
  reference
    "RFC 6790: The Use of Entropy Labels in MPLS Forwarding
    RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
    Route Object (ERO)"
}

identity oam-mep-entity-desired {
  base lsp-attributes-flags;
  description
    "OAM Maintenance Entity Group End Point (MEP) entities
    desired.";
  reference
    "RFC 7260: GMPLS RSVP-TE Extensions for Operations,
    Administration, and Maintenance (OAM) Configuration"
}

identity oam-mip-entity-desired {
  base lsp-attributes-flags;
  description
    "OAM Maintenance Entity Group Intermediate Points (MIP)
    entities desired.";
  reference
    "RFC 7260: GMPLS RSVP-TE Extensions for Operations,
    Administration, and Maintenance (OAM) Configuration"
}

identity srlg-collection-desired {
  base lsp-attributes-flags;
  description
    "SRLG collection desired.";
  reference
    "RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
    Route Object (ERO)
    RFC 8001: RSVP-TE Extensions for Collecting Shared Risk
    Link Group (SRLG) Information";
}
identity loopback-desired {
    base lsp-attributes-flags;
    description
      "This flag indicates that a particular node on the LSP is
       required to enter loopback mode. This can also be
       used to specify the loopback state of the node.";
    reference
      "RFC 7571: GMPLS RSVP-TE Extensions for Lock Instruct and
       Loopback";
}

identity p2mp-te-tree-eval-request {
    base lsp-attributes-flags;
    description
      "P2MP-TE tree re-evaluation request.";
    reference
      "RFC 8149: RSVP Extensions for Reoptimization of Loosely Routed
       Point-to-Multipoint Traffic Engineering Label Switched Paths
       (LSPs)";
}

identity rtm-set-desired {
    base lsp-attributes-flags;
    description
      "Residence Time Measurement (RTM) attribute flag requested.";
    reference
      "RFC 8169: Residence Time Measurement in MPLS Networks";
}

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

identity link-protection-unprotected {
    base link-protection-type;
    description
      "Unprotected link type.";
    reference
      "RFC 4872: RSVP-TE Extensions in Support of End-to-End
       Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity link-protection-extra-traffic {
    base link-protection-type;
    description
      "Extra-Traffic protected link type.";
    reference

"RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"

}  

identity link-protection-shared {
  base link-protection-type;
  description
    "Shared protected link type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}  

identity link-protection-1-for-1 {
  base link-protection-type;
  description
    "One-for-one (1:1) protected link type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}  

identity link-protection-1-plus-1 {
  base link-protection-type;
  description
    "One-plus-one (1+1) protected link type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}  

identity link-protection-enhanced {
  base link-protection-type;
  description
    "A compound link protection type derived from the underlay TE tunnel protection configuration supporting the TE link.";
}  

identity association-type {
  description
    "Base identity for the tunnel association.";
}  

identity association-type-recovery {
  base association-type;
  description
    "Association type for recovery, used to associate LSPs of the same tunnel for recovery.";
}
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
Generalized Multi-Protocol Label Switching (GMPLS) Recovery
RFC 6780: RSVP ASSOCIATION Object Extensions";
}

identity association-type-resource-sharing {
  base association-type;
  description
  "Association type for resource sharing, used to enable
  resource sharing during make-before-break.";
  reference
  "RFC 4873: GMPLS Segment Recovery
  RFC 6780: RSVP ASSOCIATION Object Extensions";
}

identity association-type-double-sided-bidir {
  base association-type;
  description
  "Association type for double-sided bidirectional LSPs,
  used to associate two LSPs of two tunnels that are
  independently configured on either endpoint.";
  reference
  "RFC 7551: RSVP-TE Extensions for Associated Bidirectional
  Label Switched Paths (LSPs)";
}

identity association-type-single-sided-bidir {
  base association-type;
  description
  "Association type for single-sided bidirectional LSPs,
  used to associate two LSPs of two tunnels, where one
  tunnel is configured on one side/endpoint and the other
  tunnel is dynamically created on the other endpoint.";
  reference
  "RFC 6780: RSVP ASSOCIATION Object Extensions
  RFC 7551: RSVP-TE Extensions for Associated Bidirectional
  Label Switched Paths (LSPs)";
}

identity objective-function-type {
  description
  "Base objective function type.";
}

identity of-minimize-cost-path {
  base objective-function-type;
  description
"Objective function for minimizing path cost.";
reference
"RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)";
}

identity of-minimize-load-path {
  base objective-function-type;
  description
    "Objective function for minimizing the load on one or more
paths.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)";
}

identity of-maximize-residual-bandwidth {
  base objective-function-type;
  description
    "Objective function for maximizing residual bandwidth.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)";
}

identity of-minimize-agg-bandwidth-consumption {
  base objective-function-type;
  description
    "Objective function for minimizing aggregate bandwidth
consumption.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)";
}

identity of-minimize-load-most-loaded-link {
  base objective-function-type;
  description
    "Objective function for minimizing the load on the link that
is carrying the highest load.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)";
}

identity of-minimize-cost-path-set {
  base objective-function-type;
  description
"Objective function for minimizing the cost on a path set."

reference
"RFC 5541: Encoding of Objective Functions in the Path
Computation Element Communication Protocol (PCEP)"

identity path-computation-method {
  description
    "Base identity for supported path computation mechanisms."
}

identity path-locally-computed {
  base path-computation-method;
  description
    "Indicates a constrained-path LSP in which the
    path is computed by the local LER."
  reference
    "RFC 3272: Overview and Principles of Internet Traffic
    Engineering, Section 5.4"
}

identity path-externally-queried {
  base path-computation-method;
  description
    "Constrained-path LSP in which the path is obtained by
    querying an external source, such as a PCE server.
    In the case that an LSP is defined to be externally queried,
    it may also have associated explicit definitions (provided
    to the external source to aid computation). The path that is
    returned by the external source may require further local
    computation on the device."
  reference
    "RFC 3272: Overview and Principles of Internet Traffic
    Engineering
    RFC 4657: Path Computation Element (PCE) Communication
    Protocol Generic Requirements"
}

identity path-explicitly-defined {
  base path-computation-method;
  description
    "Constrained-path LSP in which the path is
    explicitly specified as a collection of strict and/or loose
    hops."
  reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
    RFC 3272: Overview and Principles of Internet Traffic
    Engineering";
identity lsp-metric-type {
    description
    "Base identity for the LSP metric specification types.";
}

identity lsp-metric-relative {
    base lsp-metric-type;
    description
    "The metric specified for the LSPs to which this identity
    refers is specified as a value relative to the IGP metric
cost to the LSP’s tail end.";
    reference
    "RFC 4657: Path Computation Element (PCE) Communication
    Protocol Generic Requirements";
}

identity lsp-metric-absolute {
    base lsp-metric-type;
    description
    "The metric specified for the LSPs to which this identity
    refers is specified as an absolute value.";
    reference
    "RFC 4657: Path Computation Element (PCE) Communication
    Protocol Generic Requirements";
}

identity lsp-metric-inherited {
    base lsp-metric-type;
    description
    "The metric for the LSPs to which this identity refers is
    not specified explicitly; rather, it is directly inherited
    from the IGP cost.";
    reference
    "RFC 4657: Path Computation Element (PCE) Communication
    Protocol Generic Requirements";
}

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

identity te-tunnel-p2p {
    base te-tunnel-type;
    description
    "TE Point-to-Point (P2P) tunnel type.";
reference
  "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity te-tunnel-p2mp {
  base te-tunnel-type;
  description
  "TE P2MP tunnel type.";
  reference
  "RFC 4875: Extensions to Resource Reservation Protocol -
  Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
  Label Switched Paths (LSPs)";
}

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

identity tunnel-action-resetup {
  base tunnel-action-type;
  description
  "TE tunnel action that tears down the tunnel’s current LSP
  (if any) and attempts to re-establish a new LSP.";
}

identity tunnel-action-reoptimize {
  base tunnel-action-type;
  description
  "TE tunnel action that reoptimizes the placement of the
  tunnel LSP(s).";
}

identity tunnel-action-switchpath {
  base tunnel-action-type;
  description
  "TE tunnel action that switches the tunnel’s LSP to use the
  specified path.";
}

identity te-action-result {
  description
  "Base identity from which specific TE action results
  are derived.";
}

identity te-action-success {
base te-action-result;
description
  "TE action was successful.";
}

identity te-action-fail {
  base te-action-result;
description
  "TE action failed.";
}

identity tunnel-action-inprogress {
  base te-action-result;
description
  "TE action is in progress.";
}

identity tunnel-admin-state-type {
  description
  "Base identity for TE tunnel administrative states.";
}

identity tunnel-admin-state-up {
  base tunnel-admin-state-type;
description
  "Tunnel’s administrative state is up.";
}

identity tunnel-admin-state-down {
  base tunnel-admin-state-type;
description
  "Tunnel’s administrative state is down.";
}

identity tunnel-state-type {
  description
  "Base identity for TE tunnel states.";
}

identity tunnel-state-up {
  base tunnel-state-type;
description
  "Tunnel’s state is up.";
}

identity tunnel-state-down {
  base tunnel-state-type;
description
  "Tunnel’s state is down.";
}
"Tunnel’s state is down."
}

identity lsp-state-type {
    description
    "Base identity for TE LSP states."
}

identity lsp-path-computing {
    base lsp-state-type;
    description
    "State path computation is in progress."
}

identity lsp-path-computation-ok {
    base lsp-state-type;
    description
    "State path computation was successful."
}

identity lsp-path-computation-failed {
    base lsp-state-type;
    description
    "State path computation failed."
}

identity lsp-state-setting-up {
    base lsp-state-type;
    description
    "State is being set up."
}

identity lsp-state-setup-ok {
    base lsp-state-type;
    description
    "State setup was successful."
}

identity lsp-state-setup-failed {
    base lsp-state-type;
    description
    "State setup failed."
}

identity lsp-state-up {
    base lsp-state-type;
    description
    "State is up."
identity lsp-state-tearing-down {
    base lsp-state-type;
    description
        "State is being torn down.";
}

identity lsp-state-down {
    base lsp-state-type;
    description
        "State is down.";
}

identity path-invalidation-action-type {
    description
        "Base identity for TE path invalidation action types.";
}

identity path-invalidation-action-drop {
    base path-invalidation-action-type;
    description
        "Upon invalidation of the TE tunnel path, the tunnel remains
        valid, but any packet mapped over the tunnel is dropped.";
    reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
        Section 2.5";
}

identity path-invalidation-action-teardown {
    base path-invalidation-action-type;
    description
        "TE path invalidation action teardown.";
    reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
        Section 2.5";
}

identity lsp-restoration-type {
    description
        "Base identity from which LSP restoration types are derived.";
}

identity lsp-restoration-restore-any {
    base lsp-restoration-type;
    description
        "Any LSP affected by a failure is restored.";
}
identity lsp-restoration-restore-all {
  base lsp-restoration-type;
  description
    "Affected LSPs are restored after all LSPs of the tunnel are broken.";
}

identity restoration-scheme-type {
  description
    "Base identity for LSP restoration schemes.";
}

identity restoration-scheme-preconfigured {
  base restoration-scheme-type;
  description
    "Restoration LSP is preconfigured prior to the failure.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
     for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity restoration-scheme-precomputed {
  base restoration-scheme-type;
  description
    "Restoration LSP is precomputed prior to the failure.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
     for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity restoration-scheme-presignaled {
  base restoration-scheme-type;
  description
    "Restoration LSP is presignaled prior to the failure.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
     for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity lsp-protection-type {
  description
    "Base identity from which LSP protection types are derived.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
     Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-unprotected {
base lsp-protection-type;
description
"'Unprotected' LSP protection type."
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
  Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-reroute-extra {
  base lsp-protection-type;
description
"'(Full) Rerouting' LSP protection type."
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
  Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-reroute {
  base lsp-protection-type;
description
"'Rerouting without Extra-Traffic' LSP protection type."
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
  Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-1-for-n {
  base lsp-protection-type;
description
"'1:N Protection with Extra-Traffic' LSP protection type."
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
  Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-1-for-1 {
  base lsp-protection-type;
description
"'LSP protection '1:1 Protection Type'."
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
  Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-unidir-1-plus-1 {
  base lsp-protection-type;
description
"'1+1 Unidirectional Protection' LSP protection type."
}
identity lsp-protection-bidir-1-plus-1 {
  base lsp-protection-type;
  description
    "'1+1 Bidirectional Protection' LSP protection type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-extra-traffic {
  base lsp-protection-type;
  description
    "Extra-Traffic LSP protection type.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity lsp-protection-state {
  description
    "Base identity of protection states for reporting purposes.";
}

identity normal {
  base lsp-protection-state;
  description
    "Normal state.";
}

identity signal-fail-of-protection {
  base lsp-protection-state;
  description
    "The protection transport entity has a signal fail condition that is of higher priority than the forced switchover command.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity lockout-of-protection {
  base lsp-protection-state;
  description

"A Loss of Protection (LoP) command is active."
reference
"RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity forced-switch {
    base lsp-protection-state;
    description
    "A forced switchover command is active."
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity signal-fail {
    base lsp-protection-state;
    description
    "There is a signal fail condition on either the working path or the protection path."
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity signal-degrade {
    base lsp-protection-state;
    description
    "There is a signal degrade condition on either the working path or the protection path."
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity manual-switch {
    base lsp-protection-state;
    description
    "A manual switchover command is active."
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity wait-to-restore {
    base lsp-protection-state;
    description
    "A WTR timer is running."
}
identity do-not-revert {
    base lsp-protection-state;
    description
        "A Do Not Revert (DNR) condition is active because of
        non-revertive behavior.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity failure-of-protocol {
    base lsp-protection-state;
    description
        "LSP protection is not working because of a protocol failure
        condition.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity protection-external-commands {
    description
        "Base identity from which protection-related external commands
        used for troubleshooting purposes are derived.";
}

identity action-freeze {
    base protection-external-commands;
    description
        "A temporary configuration action initiated by an operator
        command that prevents any switchover action from being taken
        and, as such, freezes the current state.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity clear-freeze {
    base protection-external-commands;
    description
        "An action that clears the active freeze state.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}
for Generalized Multi-Protocol Label Switching (GMPLS)."
}

identity action-lockout-of-normal {
    base protection-external-commands;
    description "A temporary configuration action initiated by an operator
command to ensure that the normal traffic is not allowed
to use the protection transport entity.";
    reference "RFC 4427: Recovery (Protection and Restoration) Terminology
for Generalized Multi-Protocol Label Switching (GMPLS)."
}

identity clear-lockout-of-normal {
    base protection-external-commands;
    description "An action that clears the active lockout of the
normal state.";
    reference "RFC 4427: Recovery (Protection and Restoration) Terminology
for Generalized Multi-Protocol Label Switching (GMPLS)."
}

identity action-lockout-of-protection {
    base protection-external-commands;
    description "A temporary configuration action initiated by an operator
command to ensure that the protection transport entity is
temporarily not available to transport a traffic signal
(either normal or Extra-Traffic).";
    reference "RFC 4427: Recovery (Protection and Restoration) Terminology
for Generalized Multi-Protocol Label Switching (GMPLS)."
}

identity action-forced-switch {
    base protection-external-commands;
    description "A switchover action initiated by an operator command to switch
the Extra-Traffic signal, the normal traffic signal, or the
null signal to the protection transport entity, unless a
switchover command of equal or higher priority is in effect.";
    reference "RFC 4427: Recovery (Protection and Restoration) Terminology
for Generalized Multi-Protocol Label Switching (GMPLS)."
}
identity action-manual-switch {
  base protection-external-commands;
  description
    "A switchover action initiated by an operator command to switch
    the Extra-Traffic signal, the normal traffic signal, or
    the null signal to the protection transport entity, unless
    a fault condition exists on other transport entities or a
    switchover command of equal or higher priority is in effect.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity action-exercise {
  base protection-external-commands;
  description
    "An action that starts testing whether or not APS communication
    is operating correctly. It is of lower priority than any
    other state or command.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity clear {
  base protection-external-commands;
  description
    "An action that clears the active near-end lockout of a
    protection, forced switchover, manual switchover, WTR state,
    or exercise command.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity switching-capabilities {
  description
    "Base identity for interface switching capabilities.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity switching-psc1 {
  base switching-capabilities;
  description
    "Packet-Switch Capable-1 (PSC-1).";
  reference

"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description";
}

identity switching-evpl {
  base switching-capabilities;
  description
    "Ethernet Virtual Private Line (EVPL).";
  reference
    "RFC 6004: Generalized MPLS (GMPLS) Support for Metro Ethernet Forum and G.8011 Ethernet Service Switching";
}

identity switching-l2sc {
  base switching-capabilities;
  description
    "Layer-2 Switch Capable (L2SC).";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description";
}

identity switching-tdm {
  base switching-capabilities;
  description
    "Time-Division-Multiplex Capable (TDM).";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description";
}

identity switching-otn {
  base switching-capabilities;
  description
    "OTN-TDM capable.";
  reference
    "RFC 7138: Traffic Engineering Extensions to OSPF for GMPLS Control of Evolving G.709 Optical Transport Networks";
}

identity switching-dcsc {
  base switching-capabilities;
  description
    "Data Channel Switching Capable (DCSC).";
  reference
    "RFC 6002: Generalized MPLS (GMPLS) Data Channel Switching Capable (DCSC) and Channel Set Label Extensions";
}
identity switching-lsc {
    base switching-capabilities;
    description "Lambda-Switch Capable (LSC).";
    reference "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity switching-fsc {
    base switching-capabilities;
    description "Fiber-Switch Capable (FSC).";
    reference "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-types {
    description "Base identity for encoding types.";
    reference "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-packet {
    base lsp-encoding-types;
    description "Packet LSP encoding.";
    reference "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-ethernet {
    base lsp-encoding-types;
    description "Ethernet LSP encoding.";
    reference "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-pdh {
    base lsp-encoding-types;
    description "ANSI/ETSI PDH LSP encoding.";
}


reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";

} identity lsp-encoding-sdh {
base lsp-encoding-types;
description
"SDH ITU-T G.707 / SONET ANSI T1.105 LSP encoding.";
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";

}

identity lsp-encoding-digital-wrapper {
base lsp-encoding-types;
description
"Digital Wrapper LSP encoding.";
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";

}

identity lsp-encoding-lambda {
base lsp-encoding-types;
description
"Lambda (photonic) LSP encoding.";
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";

}

identity lsp-encoding-fiber {
base lsp-encoding-types;
description
"Fiber LSP encoding.";
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";

}

identity lsp-encoding-fiber-channel {
base lsp-encoding-types;
description
"FiberChannel LSP encoding.";
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";


identity lsp-encoding-oduk {
    base lsp-encoding-types;
    description
        "G.709 ODUk (Digital Path) LSP encoding.";
    reference
        "RFC 4328: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Extensions for G.709 Optical Transport Networks Control";
}

identity lsp-encoding-optical-channel {
    base lsp-encoding-types;
    description
        "G.709 Optical Channel LSP encoding.";
    reference
        "RFC 4328: Generalized Multi-Protocol Label Switching (GMPLS) Signaling Extensions for G.709 Optical Transport Networks Control";
}

identity lsp-encoding-line {
    base lsp-encoding-types;
    description
        "Line (e.g., 8B/10B) LSP encoding.";
    reference
        "RFC 6004: Generalized MPLS (GMPLS) Support for Metro Ethernet Forum and G.8011 Ethernet Service Switching";
}

identity path-signaling-type {
    description
        "Base identity from which specific LSP path setup types are derived.";
}

identity path-setup-static {
    base path-signaling-type;
    description
        "Static LSP provisioning path setup.";
}

identity path-setup-rsvp {
    base path-signaling-type;
    description
        "RSVP-TE signaling path setup.";
    reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity path-setup-sr {
  base path-signaling-type;
  description
    "Segment-routing path setup.";
}

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

identity path-scope-segment {
  base path-scope-type;
  description
    "Path scope segment.";
  reference
    "RFC 4873: GMPLS Segment Recovery";
}

identity path-scope-end-to-end {
  base path-scope-type;
  description
    "Path scope end to end.";
  reference
    "RFC 4873: GMPLS Segment Recovery";
}

identity route-usage-type {
  description
    "Base identity for route usage.";
}

identity route-include-object {
  base route-usage-type;
  description
    "'Include route' object.";
}

identity route-exclude-object {
  base route-usage-type;
  description
    "'Exclude route' object.";
  reference
    "RFC 4874: Exclude Routes - Extension to Resource ReserVation
Protocol-Traffic Engineering (RSVP-TE);}

identity route-exclude-srlg {
   base route-usage-type;
   description
      "Excludes SRLGs.";
   reference
      "RFC 4874: Exclude Routes - Extension to Resource ReserVation
         Protocol-Traffic Engineering (RSVP-TE)";
}

identity path-metric-type {
   description
      "Base identity for the path metric type.";
}

identity path-metric-te {
   base path-metric-type;
   description
      "TE path metric.";
   reference
      "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
         second MPLS Traffic Engineering (TE) Metric";
}

identity path-metric-igp {
   base path-metric-type;
   description
      "IGP path metric.";
   reference
      "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
         second MPLS Traffic Engineering (TE) Metric";
}

identity path-metric-hop {
   base path-metric-type;
   description
      "Hop path metric.";
}

identity path-metric-delay-average {
   base path-metric-type;
   description
      "Average unidirectional link delay.";
   reference
      "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}
identity path-metric-delay-minimum {
    base path-metric-type;
    description
        "Minimum unidirectional link delay.";
    reference
        "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}

identity path-metric-residual-bandwidth {
    base path-metric-type;
    description
        "Unidirectional Residual Bandwidth, which is defined to be
        Maximum Bandwidth (RFC 3630) minus the bandwidth currently
        allocated to LSPs.";
    reference
        "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2
        RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}

identity path-metric-optimize-includes {
    base path-metric-type;
    description
        "A metric that optimizes the number of included resources
        specified in a set.";
}

identity path-metric-optimize-excludes {
    base path-metric-type;
    description
        "A metric that optimizes to a maximum the number of excluded
        resources specified in a set.";
}

identity path-tiebreaker-type {
    description
        "Base identity for the path tiebreaker type.";
}

identity path-tiebreaker-minfill {
    base path-tiebreaker-type;
    description
        "Min-Fill LSP path placement.";
}

identity path-tiebreaker-maxfill {
    base path-tiebreaker-type;
    description
"Max-Fill LSP path placement."
}

identity path-tiebreaker-random {
  base path-tiebreaker-type;
  description
    "Random LSP path placement."
}

identity resource-affinities-type {
  description
    "Base identity for resource class affinities."
    reference
    "RFC 2702: Requirements for Traffic Engineering Over MPLS"
}

identity resource-aff-include-all {
  base resource-affinities-type;
  description
    "The set of attribute filters associated with a
tunnel, all of which must be present for a link
to be acceptable."
    reference
    "RFC 2702: Requirements for Traffic Engineering Over MPLS
RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels"
}

identity resource-aff-include-any {
  base resource-affinities-type;
  description
    "The set of attribute filters associated with a
tunnel, any of which must be present for a link
to be acceptable."
    reference
    "RFC 2702: Requirements for Traffic Engineering Over MPLS
RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels"
}

identity resource-aff-exclude-any {
  base resource-affinities-type;
  description
    "The set of attribute filters associated with a
tunnel, any of which renders a link unacceptable."
    reference
    "RFC 2702: Requirements for Traffic Engineering Over MPLS
RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}
identity te-optimization-criterion {
  description
    "Base identity for the TE optimization criteria.";
  reference
    "RFC 3272: Overview and Principles of Internet Traffic
      Engineering";
}

identity not-optimized {
  base te-optimization-criterion;
  description
    "Optimization is not applied.";
}

identity cost {
  base te-optimization-criterion;
  description
    "Optimized on cost.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
      Computation Element Communication Protocol (PCEP)";
}

identity delay {
  base te-optimization-criterion;
  description
    "Optimized on delay.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
      Computation Element Communication Protocol (PCEP)";
}

identity path-computation-srlg-type {
  description
    "Base identity for SRLG path computation.";
}

identity srlg-ignore {
  base path-computation-srlg-type;
  description
    "Ignores SRLGs in the path computation.";
}

identity srlg-strict {
  base path-computation-srlg-type;
  description
    "Includes a strict SRLG check in the path computation.";
}
identity srlg-preferred {
  base path-computation-srlg-type;
  description
    "Includes a preferred SRLG check in the path computation.";
}

identity srlg-weighted {
  base path-computation-srlg-type;
  description
    "Includes a weighted SRLG check in the path computation.";
}

/**
 * TE bandwidth groupings
**/ 

grouping te-bandwidth {
  description
    "This grouping defines the generic TE bandwidth. For some known data-plane technologies, specific modeling structures are specified. The string-encoded 'te-bandwidth' type is used for unspecified technologies. The modeling structure can be augmented later for other technologies.";
  container te-bandwidth {
    description
      "Container that specifies TE bandwidth. The choices can be augmented for specific data-plane technologies.";
    choice technology {
      default "generic";
      description
        "Data-plane technology type.";
      case generic {
        leaf generic {
          type te-bandwidth;
          description
            "Bandwidth specified in a generic format.";
        }
      }
    }
  }
}

/**
 * TE label groupings
**/ 

grouping te-label {

description
  "This grouping defines the generic TE label.
  The modeling structure can be augmented for each technology.
  For unspecified technologies, 'rt-types:generalized-label'
  is used."
container te-label {
  description
    "Container that specifies the TE label. The choices can
    be augmented for specific data-plane technologies."
  choice technology {
    default "generic";
    description
      "Data-plane technology type."
    case generic {
      leaf generic {
        type rt-types:generalized-label;
        description
          "TE label specified in a generic format."
      }
    }
  }
  leaf direction {
    type te-label-direction;
    default "forward";
    description
      "Label direction."
  }
}

grouping te-topology-identifier {
  description
    "Augmentation for a TE topology."
  container te-topology-identifier {
    description
      "TE topology identifier container."
    leaf provider-id {
      type te-global-id;
      default "0";
      description
        "An identifier to uniquely identify a provider.
        If omitted, it assumes that the topology provider ID
        value = 0 (the default)."
    }
    leaf client-id {
      type te-global-id;
      default "0";
      description
  

"An identifier to uniquely identify a client. If omitted, it assumes that the topology client ID value = 0 (the default)."

leaf topology-id {
  type te-topology-id;
  default "";
  description
  "When the datastore contains several topologies, 'topology-id' distinguishes between them. If omitted, the default (empty) string for this leaf is assumed."
}

/**
* TE performance metrics groupings
**/

grouping performance-metrics-one-way-delay-loss {
  description
  "Performance Metrics (PM) information in real time that can be applicable to links or connections. PM defined in this grouping are applicable to generic TE PM as well as packet TE PM."
  reference
  "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
  RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
  RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions"
  leaf one-way-delay {
    type uint32 { range "0..16777215";
    } 
    description
    "One-way delay or latency in microseconds."
  }
  leaf one-way-delay-normality {
    type te-types:performance-metrics-normality;
    description
    "One-way delay normality."
  }
}

grouping performance-metrics-two-way-delay-loss {
  description
  "PM information in real time that can be applicable to links or connections.
  PM defined in this grouping are applicable to generic TE PM as well as packet TE PM."
  reference
  "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
  RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
  RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions"
  leaf two-way-delay {
    type uint32 { range "0..16777215";
    } 
    description
    "Two-way delay or latency in microseconds."
  }
  leaf two-way-delay-normality {
    type te-types:performance-metrics-normality;
    description
    "Two-way delay normality."
  }
}
connections. PM defined in this grouping are applicable to generic TE PM as well as packet TE PM.

reference
- RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
- RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
- RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions

leaf two-way-delay {
  type uint32 {
    range "0..16777215";
  }
  description
  "Two-way delay or latency in microseconds.";
}

leaf two-way-delay-normality {
  type te-types:performance-metrics-normality;
  description
  "Two-way delay normality.";
}

leaf one-way-residual-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
  "Residual bandwidth that subtracts tunnel reservations from Maximum Bandwidth (or link capacity) (RFC 3630) and provides an aggregated remainder across QoS classes.";
  reference
  "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2";
}

leaf one-way-residual-bandwidth-normality {
  type te-types:performance-metrics-normality;
  default "normal";
}
leaf one-way-available-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
    "Available bandwidth that is defined to be residual
    bandwidth minus the measured bandwidth used for the
    actual forwarding of non-RSVP-TE LSP packets. For a
    bundled link, available bandwidth is defined to be the
    sum of the component link available bandwidths.";
}
leaf one-way-available-bandwidth-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "Available bandwidth normality.";
}
leaf one-way- utilized-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
    "Bandwidth utilization that represents the actual
    utilization of the link (i.e., as measured in the router).
    For a bundled link, bandwidth utilization is defined to
    be the sum of the component link bandwidth utilizations.";
}
leaf one-way-utilized-bandwidth-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "Bandwidth utilization normality.";
}
grouping one-way-performance-metrics {
  description
    "One-way PM throttle grouping.";
  leaf one-way-delay {
    type uint32 {
      range "0..16777215";
    }
    default "0";
    description
      "One-way delay or latency in microseconds.";
  }
leaf one-way-residual-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description "Residual bandwidth that subtracts tunnel reservations from Maximum Bandwidth (or link capacity) (RFC 3630) and provides an aggregated remainder across QoS classes.";
    reference "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2";
}

leaf one-way-available-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description "Available bandwidth that is defined to be residual bandwidth minus the measured bandwidth used for the actual forwarding of non-RSVP-TE LSP packets. For a bundled link, available bandwidth is defined to be the sum of the component link available bandwidths.";
}

leaf one-way-utilized-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description "Bandwidth utilization that represents the actual utilization of the link (i.e., as measured in the router). For a bundled link, bandwidth utilization is defined to be the sum of the component link bandwidth utilizations.";
}

grouping two-way-performance-metrics {
    description "Two-way PM throttle grouping.";
    leaf two-way-delay {
        type uint32 {
            range "0..16777215";
        }
        default "0";
        description "Two-way delay or latency in microseconds.";
    }
}
grouping performance-metrics-thresholds {
  description
  "Grouping for configurable thresholds for measured attributes.";
  uses one-way-performance-metrics;
  uses two-way-performance-metrics;
}

grouping performance-metrics-attributes {
  description
  "Contains PM attributes.";
  container performance-metrics-one-way {
    description
    "One-way link performance information in real time.";
    reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions"
    RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
    RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions"
    uses performance-metrics-one-way-delay-loss;
    uses performance-metrics-one-way-bandwidth;
  }
  container performance-metrics-two-way {
    description
    "Two-way link performance information in real time.";
    reference
    "RFC 6374: Packet Loss and Delay Measurement for MPLS Networks"
    uses performance-metrics-two-way-delay-loss;
  }
}

grouping performance-metrics-throttle-container {
  description
  "Controls PM throttling.";
  container throttle {
    must 'suppression-interval >= measure-interval' {
      error-message "'suppression-interval' cannot be less than " + "'measure-interval'.";
      description
      "Constraint on 'suppression-interval' and 'measure-interval'.";
    }
    description
    "Link performance information in real time.";
    reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
  }
}

Busi, et al. Expires 8 September 2022 [Page 58]
RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions

leaf one-way-delay-offset {
  type uint32 {
    range "0..16777215";
  }
  default "0";
  description
    "Offset value to be added to the measured delay value.";
}

leaf measure-interval {
  type uint32;
  default "30";
  description
    "Interval, in seconds, to measure the extended metric values.";
}

leaf advertisement-interval {
  type uint32;
  default "0";
  description
    "Interval, in seconds, to advertise the extended metric values.";
}

leaf suppression-interval {
  type uint32 {
    range "1..max";
  }
  default "120";
  description
    "Interval, in seconds, to suppress advertisement of the extended metric values.";
  reference
    "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions, Section 6";
}

container threshold-out {
  uses performance-metrics-thresholds;
  description
    "If the measured parameter falls outside an upper bound for all but the minimum-delay metric (or a lower bound for the minimum-delay metric only) and the advertised value is not already outside that bound, an 'anomalous' announcement (anomalous bit set) will be triggered.";
}

container threshold-in {
uses performance-metrics-thresholds;

description
"If the measured parameter falls inside an upper bound
for all but the minimum-delay metric (or a lower bound
for the minimum-delay metric only) and the advertised
value is not already inside that bound, a 'normal'
announcement (anomalous bit cleared) will be triggered."
}
}

container threshold-accelerated-advertisement {

description
"When the difference between the last advertised value and
the current measured value exceeds this threshold, an
'anomalous' announcement (anomalous bit set) will be
triggered."

uses performance-metrics-thresholds;
}

/**
 * TE tunnel generic groupings
 /**

grouping explicit-route-hop {

description
"The explicit route entry grouping."

choice type {

description
"The explicit route entry type."

case numbered-node-hop {

container numbered-node-hop {

leaf node-id {

type te-node-id;

mandatory true;

description
"The identifier of a node in the TE topology.";
}

leaf hop-type {

type te-hop-type;

default "strict";

description
"Strict or loose hop.";
}

description
"Numbered node route hop."

reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
Section 4.3, EXPLICIT_ROUTE in RSVP-TE
case numbered-link-hop {
  container numbered-link-hop {
    leaf link-tp-id {
      type te-tp-id;
      mandatory true;
      description
      "TE Link Termination Point (LTP) identifier.";
    }
    leaf hop-type {
      type te-hop-type;
      default "strict";
      description
      "Strict or loose hop.";
    }
    leaf direction {
      type te-link-direction;
      default "outgoing";
      description
      "Link route object direction.";
    }
    description
    "Numbered link explicit route hop.";
    reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
     Section 4.3, EXPLICIT_ROUTE in RSVP-TE
     RFC 3477: Signalling Unnumbered Links in Resource
     ReSerVation Protocol - Traffic Engineering (RSVP-TE)";
  }
}
case unnumbered-link-hop {
  container unnumbered-link-hop {
    leaf link-tp-id {
      type te-tp-id;
      mandatory true;
      description
      "TE LTP identifier. The combination of the TE link ID
       and the TE node ID is used to identify an unnumbered
       TE link.";
    }
    leaf node-id {
      type te-node-id;
      mandatory true;
      description
      "The identifier of a node in the TE topology.";
    }
  }
}
leaf hop-type {
    type te-hop-type;
    default "strict";
    description
        "Strict or loose hop.";
}
leaf direction {
    type te-link-direction;
    default "outgoing";
    description
        "Link route object direction.";
}
description
    "Unnumbered link explicit route hop."
reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
     Section 4.3, EXPLICIT_ROUTE in RSVP-TE
    RFC 3477: Signalling Unnumbered Links in Resource
     ReSerVation Protocol - Traffic Engineering (RSVP-TE)";
}
case as-number {
    container as-number-hop {
        leaf as-number {
            type inet:as-number;
            mandatory true;
            description
                "The Autonomous System (AS) number.";
        }
        leaf hop-type {
            type te-hop-type;
            default "strict";
            description
                "Strict or loose hop.";
        }
        description
            "AS explicit route hop.";
    }
}
case label {
    container label-hop {
        description
            "Label hop type.";
        uses te-label;
    }
    description
        "The label explicit route hop type.";
grouping record-route-state {
  description
    "The Record Route grouping.";
  leaf index {
    type uint32;
    description
        "Record Route hop index. The index is used to
         identify an entry in the list. The order of entries
         is defined by the user without relying on key values.";
  }
  choice type {
    description
      "The Record Route entry type.";
    case numbered-node-hop {
      container numbered-node-hop {
        description
          "Numbered node route hop container."
        leaf node-id {
          type te-node-id;
          mandatory true;
          description
              "The identifier of a node in the TE topology.";
        }
        leaf-list flags {
          type path-attribute-flags;
          description
              "Path attributes flags.";
          reference
              "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
              RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP
              Tunnels
              RFC 4561: Definition of a Record Route Object (RRO)
              Node-Id Sub-Object";
        }
      }
      description
          "Numbered node route hop.";
    }
    case numbered-link-hop {
      container numbered-link-hop {
        description
          "Numbered link route hop container.";
        leaf link-tp-id {
          type te-tp-id;
        }
      }
      description
          "Numbered link route hop.";
    }
  }
}

mandatory true;

description
"Numbered TE LTP identifier."

}
leaf-list flags {

type path-attribute-flags;

description
"Path attributes flags."

reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
RFC 4561: Definition of a Record Route Object (RRO)
Node-Id Sub-Object"

}

description
"Numbered link route hop."

}

case unnumbered-link-hop {

container unnumbered-link-hop {


leaf link-tp-id {

type te-tp-id;

mandatory true;


description
"TE LTP identifier. The combination of the TE link ID
and the TE node ID is used to identify an unnumbered TE link."

}

leaf node-id {

type te-node-id;


description
"The identifier of a node in the TE topology."

}

leaf-list flags {

type path-attribute-flags;


description
"Path attributes flags."

reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
RFC 4561: Definition of a Record Route Object (RRO)
Node-Id Sub-Object"

}

description
"Unnumbered link Record Route hop."

reference
"RFC 3477: Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)";

} description
  "Unnumbered link route hop."
}

} case label {
  container label-hop {
    description
    "Label route hop type."
    uses te-label;
    leaf-list flags {
      type path-attribute-flags;
      description
      "Path attributes flags."
      reference
      "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
      RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
      RFC 4561: Definition of a Record Route Object (RRO) Node-Id Sub-Object"
    }
    description
    "The label Record Route entry types."
  }
}

} grouping label-restriction-info {
  description
  "Label set item information."
  leaf restriction {
    type enum enumeration {
      enum inclusive {
        description
        "The label or label range is inclusive."
      }
      enum exclusive {
        description
        "The label or label range is exclusive."
      }
    }
    default "inclusive"
    description
    "Indicates whether the list item is inclusive or exclusive."
  }
  leaf index {

type uint32;

description
  "The index of the label restriction list entry."
;
}

container label-start {
    must "(not(../label-end/te-label/direction) and"
        + " not(te-label/direction))"
        + " or "
        + "(../label-end/te-label/direction = te-label/direction)"
        + " or "
        + "(not(te-label/direction) and"
        + " (.label-end/te-label/direction = 'forward'))"
        + " or "
        + "(not(../label-end/te-label/direction) and"
        + " (te-label/direction = 'forward'))" {
        error-message "'label-start' and 'label-end' must have the "
            + "same direction.";
    }

description
  "This is the starting label if a label range is specified.
   This is the label value if a single label is specified,
   in which case the 'label-end' attribute is not set.";
    uses te-label;
}

container label-end {
    must "(not(../label-start/te-label/direction) and"
        + " not(te-label/direction))"
        + " or "
        + "(../label-start/te-label/direction = te-label/direction)"
        + " or "
        + "(not(te-label/direction) and"
        + " (.label-start/te-label/direction = 'forward'))"
        + " or "
        + "(not(../label-start/te-label/direction) and"
        + " (te-label/direction = 'forward'))" {
        error-message "'label-start' and 'label-end' must have the "
            + "same direction.";
    }

description
  "This is the ending label if a label range is specified.
   This attribute is not set if a single label is specified.";
    uses te-label;
}

container label-step {
    description
    "The step increment between labels in the label range.
     The label start/end values will have to be consistent
     with the sign of label step. For example,
'label-start' < 'label-end' enforces 'label-step' > 0
'label-start' > 'label-end' enforces 'label-step' < 0.

choice technology {
    default "generic";
    description "Data-plane technology type."
    case generic {
        leaf generic {
            type int32;
            default "1";
            description "Label range step."
        }
    }
}

leaf range-bitmap {
    type yang:hex-string;
    description "When there are gaps between 'label-start' and 'label-end',
    this attribute is used to specify the positions
    of the used labels. This is represented in big endian as
    'hex-string'.
    The most significant byte in the hex-string is the farthest
to the left in the byte sequence. Leading zero bytes in the
configured value may be omitted for brevity.
Each bit position in the 'range-bitmap' 'hex-string' maps
to a label in the range derived from 'label-start'.

For example, assuming that 'label-start' = 16000 and
'range-bitmap' = 0x01000001, then:

- bit position (0) is set, and the corresponding mapped
  label from the range is 16000 + (0 * 'label-step') or
  16000 for default 'label-step' = 1.
- bit position (24) is set, and the corresponding mapped
  label from the range is 16000 + (24 * 'label-step') or
  16024 for default 'label-step' = 1."
}

grouping label-set-info {
    description "Grouping for the list of label restrictions specifying what
    labels may or may not be used."
    container label-restrictions {
        description "The label restrictions container.";
    }
}
list label-restriction {
  key "index";
  description
    "The absence of the label restrictions container implies
    that all labels are acceptable; otherwise, only restricted
    labels are available.";
  reference
    "RFC 7579: General Network Element Constraint Encoding
    for GMPLS-Controlled Networks";
  uses label-restriction-info;
}
}
}

grouping optimization-metric-entry {
  description
    "Optimization metrics configuration grouping.";
  leaf metric-type {
    type identityref {
      base path-metric-type;
    }
    description
      "Identifies the 'metric-type' that the path computation
      process uses for optimization.";
  }
  leaf weight {
    type uint8;
    default "1";
    description
      "TE path metric normalization weight.";
  }
  container explicit-route-exclude-objects {
    when "./metric-type = "
      + "'te-types:path-metric-optimize-excludes'";
    description
      "Container for the 'exclude route' object list.";
    uses path-route-exclude-objects;
  }
  container explicit-route-include-objects {
    when "./metric-type = "
      + "'te-types:path-metric-optimize-includes'";
    description
      "Container for the 'include route' object list.";
    uses path-route-include-objects;
  }
}

grouping common-constraints {
description
"Common constraints grouping that can be set on
a constraint set or directly on the tunnel."
uses te-bandwidth {
  description
  "A requested bandwidth to use for path computation.";
}
leaf link-protection {
  type identityref {
    base link-protection-type;
  }
  default "te-types:link-protection-unprotected";
  description
  "Link protection type required for the links included
  in the computed path.";
  reference
  "RFC 4202: Routing Extensions in Support of
  Generalized Multi-Protocol Label Switching (GMPLS)";
}
leaf setup-priority {
  type uint8 {
    range "0..7";
    default "7";
  }
  description
  "TE LSP requested setup priority.";
  reference
  "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}
leaf hold-priority {
  type uint8 {
    range "0..7";
    default "7";
  }
  description
  "TE LSP requested hold priority.";
  reference
  "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}
leaf signaling-type {
  type identityref {
    base path-signaling-type;
  }
  default "te-types:path-setup-rsvp";
  description
  "TE tunnel path signaling type.";
}
grouping tunnel-constraints {
    description
    "Tunnel constraints grouping that can be set on a constraint set or directly on the tunnel.";
    uses te-topology-identifier;
    uses common-constraints;
}

grouping path-constraints-route-objects {
    description
    "List of route entries to be included or excluded when performing the path computation.";
    container explicit-route-objects-always {
        description
        "Container for the 'exclude route' object list.";
        list route-object-exclude-always {
            key "index";
            ordered-by user;
            description
            "List of route objects to always exclude from the path computation.";
            leaf index {
                type uint32;
                description
                "Explicit Route Object index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values.";
            }
            uses explicit-route-hop;
        }
        list route-object-include-exclude {
            key "index";
            ordered-by user;
            description
            "List of route objects to include or exclude in the path computation.";
            leaf explicit-route-usage {
                type identityref {
                    base route-usage-type;
                }
                default "te-types:route-include-object";
                description
                "Indicates whether to include or exclude the route object. The default is to include it.";
            }
            leaf index {
                type uint32;
                description
                "Explicit Route Object index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values.";
            }
        }
    }
}

Busi, et al. Expires 8 September 2022 [Page 70]
"Route object include-exclude index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values."

uses explicit-route-hop {
  augment "type" {
    case srlg {
      container srlg {
        description "SRLG container.";
        leaf srlg {
          type uint32;
          description "SRLG value.";
        }
      }
    }
    description "An SRLG value to be included or excluded.";
    description "Augmentation for a generic explicit route for SRLG exclusion.";
  }
}

grouping path-route-include-objects {
  description "List of route objects to be included when performing the path computation.";
  list route-object-include-object {
    key "index";
    ordered-by user;
    description "List of Explicit Route Objects to be included in the path computation.";
    leaf index {
      type uint32;
      description "Route object entry index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values.";
    }
    uses explicit-route-hop;
  }
}
grouping path-route-exclude-objects {
    description
        "List of route objects to be excluded when performing
         the path computation.";
    list route-object-exclude-object {
        key "index";
        ordered-by user;
        description
            "List of Explicit Route Objects to be excluded in the
             path computation.";
        leaf index {
            type uint32;
            description
                "Route object entry index. The index is used to
                 identify an entry in the list. The order of entries
                 is defined by the user without relying on key values.";
        }
        uses explicit-route-hop {
            augment "type" {
                case srlg {
                    container srlg {
                        description
                            "SRLG container.";
                        leaf srlg {
                            type uint32;
                            description
                                "SRLG value.";
                    }
                }
                description
                    "An SRLG value to be included or excluded.";
            }
            description
                "Augmentation for a generic explicit route for SRLG
                 exclusion.";
        }
    }
}

grouping generic-path-metric-bounds {
    description
        "TE path metric bounds grouping.";
    container path-metric-bounds {
        description
            "TE path metric bounds container.";
    list path-metric-bound {
        key "metric-type";
description
  "List of TE path metric bounds.";
leaf metric-type {
  type identityref {
    base path-metric-type;
  }
description
  "Identifies an entry in the list of 'metric-type' items bound for the TE path.";
}
leaf upper-bound {
  type uint64;
default "0";
description
  "Upper bound on the end-to-end TE path metric. A zero indicates an unbounded upper limit for the specific 'metric-type'.";
}
}
}
grouping generic-path-optimization {
  description
  "TE generic path optimization grouping.";
container optimizations {
  description
    "The objective function container that includes attributes to impose when computing a TE path.";
choice algorithm {
  description
    "Optimizations algorithm.";
case metric {
    if-feature "path-optimization-metric";
    /* Optimize by metric */
    list optimization-metric {
      key "metric-type";
description
        "TE path metric type.";
uses optimization-metric-entry;
    }
    /* Tiebreakers */
    container tiebreakers {
      description
        "Container for the list of tiebreakers.";
list tiebreaker {
      key "tiebreaker-type";
description
        "";
    }
  }
}

"The list of tiebreaker criteria to apply on an equally favored set of paths, in order to pick the best.";
leaf tiebreaker-type {
  type identityref {
    base path-metric-type;
  }
  description
   "Identifies an entry in the list of tiebreakers.";
}

case objective-function {
  if-feature "path-optimization-objective-function";
  /* Objective functions */
  container objective-function {
    description
     "The objective function container that includes attributes to impose when computing a TE path.";
    leaf objective-function-type {
      type identityref {
        base objective-function-type;
      }
      default "te-types:of-minimize-cost-path";
      description
       "Objective function entry.";
    }
  }
}

 grouping generic-path-affinities {
  description
   "Path affinities grouping.";
  container path-affinities-values {
    description
     "Path affinities represented as values.";
    list path-affinities-value {
      key "usage";
      description
       "List of named affinity constraints.";
      leaf usage {
        type identityref {
          base resource-affinities-type;
        }
      }
    }
  }
}
description
   "Identifies an entry in the list of value affinity constraints."
}
leaf value {
    type admin-groups;
    default "";
    description
    "The affinity value. The default is empty.";
}
}
container path-affinity-names {
    description
    "Path affinities represented as names.";
    list path-affinity-name {
        key "usage";
        description
        "List of named affinity constraints.";
        leaf usage {
            type identityref {
                base resource-affinities-type;
            }
            description
            "Identifies an entry in the list of named affinity constraints.";
        }
        list affinity-name {
            key "name";
            leaf name {
                type string;
                description
                "Identifies a named affinity entry.";
            }
            description
            "List of named affinities.";
        }
    }
}
}

grouping generic-path-srlgs {
    description
    "Path SRLG grouping.";
    container path-srlgs-lists {
        description
        "Path SRLG properties container.";
        list path-srlgs-list {
            
        }
    }
}
key "usage";
description
  "List of SRLG values to be included or excluded.";
leaf usage {
  type identityref {
    base route-usage-type;
  }
description
  "Identifies an entry in a list of SRLGs to either
   include or exclude.";
}
leaf-list values {
  type srlg;
description
  "List of SRLG values.";
}
}
}
}

container path-srlgs-names {
  description
  "Container for the list of named SRLGs.";
list path-srlgs-name {
  key "usage";
description
  "List of named SRLGs to be included or excluded.";
leaf usage {
  type identityref {
    base route-usage-type;
  }
description
  "Identifies an entry in a list of named SRLGs to either
   include or exclude.";
}
leaf-list names {
  type string;
description
  "List of named SRLGs.";
}
}
}

grouping generic-path-disjointness {
  description
  "Path disjointness grouping.";
leaf disjointness {
  type te-path-disjointness;
description
  "Path disjointness.";
}
}

Busi, et al. Expires 8 September 2022 [Page 76]
"The type of resource disjointness. When configured for a primary path, the disjointness level applies to all secondary LSPs. When configured for a secondary path, the disjointness level overrides the level configured for the primary path.;

}

}

grouping common-path-constraints-attributes {
  description
    "Common path constraints configuration grouping.";
  uses common-constraints;
  uses generic-path-metric-bounds;
  uses generic-path-affinities;
  uses generic-path-srlgs;
}

grouping generic-path-constraints {
  description
    "Global named path constraints configuration grouping.";
  container path-constraints {
    description
      "TE named path constraints container.";
    uses common-path-constraints-attributes;
    uses generic-path-disjointness;
  }
}

grouping generic-path-properties {
  description
    "TE generic path properties grouping.";
  container path-properties {
    config false;
    description
      "The TE path properties.";
    list path-metric {
      key "metric-type";
      description
        "TE path metric type.";
      leaf metric-type {
        type identityref {
          base path-metric-type;
        }
        description
          "TE path metric type.";
      }
      leaf accumulative-value {
        type uint64;
      }
    }
  }
description
  "TE path metric accumulative value."
}
}
uses generic-path-affinities;
uses generic-path-srlgs;
container path-route-objects {
  description
  "Container for the list of route objects either returned by
  the computation engine or actually used by an LSP."
  list path-route-object {
    key "index";
    ordered-by user;
    description
    "List of route objects either returned by the computation
    engine or actually used by an LSP."
    leaf index {
      type uint32;
      description
      "Route object entry index. The index is used to
      identify an entry in the list. The order of entries
      is defined by the user without relying on key
      values."
    }
    uses explicit-route-hop;
  }
}
}
}

grouping encoding-and-switching-type {
  description
  "Common grouping to define the LSP encoding and
  switching types"
  leaf encoding {
    type identityref {
      base te-types:lsp-encoding-types;
    }
    description
    "LSP encoding type."
    reference
    "RFC3945";
  }
  leaf switching-type {
    type identityref {
      base te-types:switching-capabilities;
    }
    description
"LSP switching type.";
reference
"RFC3945";
}

Acknowledgements

This document was prepared using kramdown.

Authors’ Addresses

Italo Busi
Huawei
Email: italo.busi@huawei.com

Aihua Guo
Futurewei Technologies
Email: aihuaguo.ietf@gmail.com

Xufeng Liu
IBM Corporation
Email: xufeng.liu.ietf@gmail.com

Tarek Saad
Juniper Networks
Email: tsaad@juniper.net

Rakesh Gandhi
Cisco Systems, Inc.
Email: rgandhi@cisco.com

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

Igor Bryskin
Individual
Email: i_bryskin@yahoo.com
Abstract

This document defines few additional common data types and groupings in YANG data modeling language to be imported by modules that model Traffic Engineering (TE) configuration and state capabilities.

This document updates RFC 8776 with a new revision of the module ietf-te-types.

Status of This Memo

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

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

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

This Internet-Draft will expire on 6 October 2022.
1. Introduction

After the publication of [RFC8776], the need to add a new typedef and a new grouping to ietf-te-types YANG module has arisen.

These definitions have been developed in [I-D.ietf-teas-yang-te] and [I-D.ietf-teas-yang-l3-te-topo] and are quite mature: [I-D.ietf-teas-yang-te] in particular is ready from WG Last Call.

However, these definitions have broader applicability than the I-D where they have originated, so it makes sense to move them within the ietf-te-types YANG module.
1.1. Requirements Notation

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. Terminology

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

1.3. Prefixes in Data Node Names

In this document, names of data nodes and other data model objects, added to the ietf-te-types YANG module do not need to be prefixed.

The revision of the ietf-te-types YANG module uses the prefixes defined in section 1.2 of [RFC8776].

2. Overview

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

- bandwidth-scientific-notation This types represents the bandwidth in bit-per-second, using the scientific notation (e.g., 10e3).
- encoding-and-switching-type This is a common grouping to define the LSP encoding and switching types.

3. TE Types YANG Module Revision

This section provides the updated revision of the "ietf-te-types" YANG module.

NOTE: Only the typedef bandwidth-scientific-notation and the grouping encoding-and-switching-type have been added in this module revision. Please focus your review on this part.

RFC Editor: remove the note above and this note
module ietf-te-types {
  yang-version 1.1;
  prefix te-types;

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

  organization
    "IETF Traffic Engineering Architecture and Signaling (TEAS)
     Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/teas/>
     WG List:  <mailto:teas@ietf.org>
     Editor:  Tarek Saad
              <mailto:tsaad@juniper.net>
     Editor:  Rakesh Gandhi
              <mailto:rgandhi@cisco.com>
     Editor:  Vishnu Pavan Beeram
              <mailto:vbeeram@juniper.net>
     Editor:  Xufeng Liu
              <mailto:xufeng.liu.ietf@gmail.com>
     Editor:  Igor Bryskin
              <mailto:i_bryskin@yahoo.com>";
  description
    "This YANG module contains a collection of generally useful
     YANG data type definitions specific to TE. The model fully
     conforms to the Network Management Datastore Architecture
     (NMDA).";
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.

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

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

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

revision 2022-03-25 {
  description
    "Added:
    - typedef bandwidth-scientific-notation;
    - grouping encoding-and-switching-type."
  reference
    "RFC XXXX: Updated Common YANG Data Types for Traffic Engineering"
}

// RFC Editor: replace XXXX with actual RFC number, update date information and remove this note
revision 2020-06-10 {
  description
    "Latest revision of TE types."
  reference
    "RFC 8776: Common YANG Data Types for Traffic Engineering"
}

/**
 * Typedefs
 */

typedef admin-group {
  type yang:hex-string {
    /* 01:02:03:04 */
    length "1..11";
  }
  description
    "Administrative group / resource class / color representation

in 'hex-string' type.
The most significant byte in the hex-string is the farthest
to the left in the byte sequence. Leading zero bytes in the
configured value may be omitted for brevity.

reference
"RFC 3630: Traffic Engineering (TE) Extensions to OSPF
Version 2
RFC 5305: IS-IS Extensions for Traffic Engineering
RFC 7308: Extended Administrative Groups in MPLS Traffic
Engineering (MPLS-TE)"
}

typedef admin-groups {
type union {
  type admin-group;
  type extended-admin-group;
}
description
  "Derived types for TE administrative groups."
}

typedef extended-admin-group {
type yang:hex-string;
description
  "Extended administrative group / resource class / color
  representation in 'hex-string' type.
The most significant byte in the hex-string is the farthest
to the left in the byte sequence. Leading zero bytes in the
configured value may be omitted for brevity."
reference
"RFC 7308: Extended Administrative Groups in MPLS Traffic
Engineering (MPLS-TE)"
}

typedef path-attribute-flags {
type union {
  type identityref {
    base session-attributes-flags;
  }
  type identityref {
    base lsp-attributes-flags;
  }
}
description
  "Path attributes flags type."
}

typedef performance-metrics-normality {

type enumeration {
    enum unknown {
        value 0;
        description
        "Unknown.";
    }
    enum normal {
        value 1;
        description
        "Normal. Indicates that the anomalous bit is not set.";
    }
    enum abnormal {
        value 2;
        description
        "Abnormal. Indicates that the anomalous bit is set.";
    }
}

description
"Indicates whether a performance metric is normal (anomalous
bit not set), abnormal (anomalous bit set), or unknown.";
reference
"RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
RFC 7823: Performance-Based Path Selection for Explicitly
Routed Label Switched Paths (LSPs) Using TE Metric
Extensions
RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
}

typedef srlg {
    type uint32;
    description
    "SRLG type.";
    reference
    "RFC 4203: OSPF Extensions in Support of Generalized
Multi-Protocol Label Switching (GMPLS)
RFC 5307: IS-IS Extensions in Support of Generalized
Multi-Protocol Label Switching (GMPLS)";
}

typedef te-common-status {
    type enumeration {
        enum up {
            description
            "Enabled.";
        }
        enum down {
            description
            "Disabled.";
        }
    }
}
} enum testing {
  description
  "In some test mode.";
}
enum preparing-maintenance {
  description
  "The resource is disabled in the control plane to prepare
  for a graceful shutdown for maintenance purposes.";
  reference
  "RFC 5817: Graceful Shutdown in MPLS and Generalized MPLS
  Traffic Engineering Networks";
}
enum maintenance {
  description
  "The resource is disabled in the data plane for maintenance
  purposes.";
}
enum unknown {
  description
  "Status is unknown.";
}

description
  "Defines a type representing the common states of a TE
  resource.";
}

typedef te-bandwidth {
  type string {
    pattern '0[xX]0((\.0)?[pP](\+)?0?|\.0?))'|'
      + '1(\.([da-fA-F]{0,5}[02468aAcCeE]?))'|'
      + '[pP](\+)?([20-7])'|'
      + '1[01]\d|0?\d?\d|0[xX][da-fA-F]{1,8}|\d+|'
      + '([0][xX][0((\.)?([pP])(\+)?0?|\.0?))]|'
      + '1(\.([da-fA-F]{0,5}[02468aAcCeE]?))'|'
      + '[pP](\+)?([20-7])'|'
      + '1[01]\d|0?\d?\d|0[xX][da-fA-F]{1,8}|\d+)*';
  }
  description
  "This is the generic bandwidth type. It is a string containing
  a list of numbers separated by commas, where each of these
  numbers can be non-negative decimal, hex integer, or
  hex float:

  (dec | hex | float)["(",(dec | hex | float))]"
}

For the packet-switching type, the string encoding follows
the type 'bandwidth-ieee-float32' as defined in RFC 8294 (e.g., 0x1p10), where the units are in bytes per second.

For the Optical Transport Network (OTN) switching type, a list of integers can be used, such as '0,2,3,1', indicating two ODU0s and one ODU3. (‘ODU’ stands for ‘Optical Data Unit’.) For Dense Wavelength Division Multiplexing (DWDM), a list of pairs of slot numbers and widths can be used, such as '0,2,3,3', indicating a frequency slot 0 with slot width 2 and a frequency slot 3 with slot width 3. Canonically, the string is represented as all lowercase and in hex, where the prefix ‘0x’ precedes the hex number.);

reference
"RFC 8294: Common YANG Data Types for the Routing Area
ITU-T Recommendation G.709: Interfaces for the optical transport network";
}

typedef te-ds-class {
  type uint8 {
    range "0..7";
  }
  description
    "The Differentiated Services Class-Type of traffic.";
  reference
    "RFC 4124: Protocol Extensions for Support of Diffserv-aware
     MPLS Traffic Engineering, Section 4.3.1";
}

typedef te-global-id {
  type uint32;
  description
    "An identifier to uniquely identify an operator, which can be
     either a provider or a client.
     The definition of this type is taken from RFCs 6370 and 5003.
     This attribute type is used solely to provide a globally
     unique context for TE topologies.";
  reference
    "RFC 5003: Attachment Individual Identifier (AII) Types for
     Aggregation
     RFC 6370: MPLS Transport Profile (MPLS-TP) Identifiers";
}

typedef te-hop-type {
  type enumeration {
    enum loose {
      description
        "A loose hop in an explicit path.";
    }

enum strict {
    description
        "A strict hop in an explicit path.";
}

description
"Enumerated type for specifying loose or strict paths.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 4.3.3";

typedef te-link-access-type {
    type enumeration {
        enum point-to-point {
            description
                "The link is point-to-point.";
        }
        enum multi-access {
            description
                "The link is multi-access, including broadcast and NBMA.";
        }
    }
    description
        "Defines a type representing the access type of a TE link.";
    reference
        "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2";
}

typedef te-label-direction {
    type enumeration {
        enum forward {
            description
                "Label allocated for the forward LSP direction.";
        }
        enum reverse {
            description
                "Label allocated for the reverse LSP direction.";
        }
    }
    description
        "Enumerated type for specifying the forward or reverse label.";
}

typedef te-link-direction {
type enumeration {
    enum incoming {
        description
            "The explicit route represents an incoming link on
            a node.";
    }
    enum outgoing {
        description
            "The explicit route represents an outgoing link on
            a node.";
    }
}
description
    "Enumerated type for specifying the direction of a link on
    a node.";
}
typedef te-metric {
    type uint32;
    description
        "TE metric.";
    reference
        "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
        second MPLS Traffic Engineering (TE) Metric";
}
typedef te-node-id {
    type yang:dotted-quad;
    description
        "A type representing the identifier for a node in a TE
topology. The identifier is represented as 4 octets in dotted-quad
notation. This attribute MAY be mapped to the Router Address TLV
described in Section 2.4.1 of RFC 3630, the TE Router ID
described in Section 3 of RFC 6827, the Traffic Engineering
Router ID TLV described in Section 4.3 of RFC 5305, or the
TE Router ID TLV described in Section 3.2.1 of RFC 6119.
The reachability of such a TE node MAY be achieved by a
mechanism such as that described in Section 6.2 of RFC 6827.";
    reference
        "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
Version 2, Section 2.4.1
RFC 5305: IS-IS Extensions for Traffic Engineering,
Section 4.3
RFC 6119: IPv6 Traffic Engineering in IS-IS, Section 3.2.1
RFC 6827: Automatically Switched Optical Network (ASON)
Routing for OSPFv2 Protocols, Section 3";
typedef te-oper-status {
  type te-common-status;
  description
    "Defines a type representing the operational status of
     a TE resource.";
}

typedef te-admin-status {
  type te-common-status;
  description
    "Defines a type representing the administrative status of
     a TE resource.";
}

typedef te-path-disjointness {
  type bits {
    bit node { position 0;
      description
        "Node disjoint.";
    }
    bit link { position 1;
      description
        "Link disjoint.";
    }
    bit srlg { position 2;
      description
        "SRLG (Shared Risk Link Group) disjoint.";
    }
  }
  description
    "Type of the resource disjointness for a TE tunnel path.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
     Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

typedef te-recovery-status {
  type enumeration {
    enum normal {
      description
        "Both the recovery span and the working span are fully allocated and active, data traffic is being transported over (or selected from) the working
 The recovery action has been started but not completed.

} enum recovery-succeeded {
  description
  "The recovery action has succeeded. The working span has reported a failure/degrade condition, and the user traffic is being transported (or selected) on the recovery span.

} enum recovery-failed {
  description
  "The recovery action has failed.

} enum reversion-started {
  description
  "The reversion has started.

} enum reversion-succeeded {
  description
  "The reversion action has succeeded.

} enum reversion-failed {
  description
  "The reversion has failed.

} enum recovery-unavailable {
  description
  "The recovery is unavailable, as a result of either an operator’s lockout command or a failure condition detected on the recovery span.

} enum recovery-admin {
  description
  "The operator has issued a command to switch the user traffic to the recovery span.

} enum wait-to-restore {
  description
  "The recovery domain is recovering from a failure/degrade condition on the working span that is being controlled by the Wait-to-Restore (WTR) timer.

} description
  "Defines the status of a recovery action."
typedef te-template-name {
  type string {
    pattern '/?([a-zA-Z0-9\-_.]+)(/[a-zA-Z0-9\-_.]+)*';
  }
  description
    "A type for the name of a TE node template or TE link template.";
}

typedef te-topology-event-type {
  type enumeration {
    enum add {
      value 0;
      description
        "A TE node or TE link has been added.";
    }
    enum remove {
      value 1;
      description
        "A TE node or TE link has been removed.";
    }
    enum update {
      value 2;
      description
        "A TE node or TE link has been updated.";
    }
  }
  description
    "TE event type for notifications.";
}

typedef te-topology-id {
  type union {
    type string {
      length "0";
      // empty string
    }
    type string {
      pattern '([^a-zA-Z0-9\-_.]+):*'
        + '/?([^a-zA-Z0-9\-_.]+)(/[a-zA-Z0-9\-_.]+)*';
    }
  }
}
description
"An identifier for a topology.
It is optional to have one or more prefixes at the beginning,
separated by colons. The prefixes can be 'network-types' as
defined in the 'ietf-network' module in RFC 8345, to help the
user better understand the topology before further inquiry
is made.";
reference
"RFC 8345: A YANG Data Model for Network Topologies";
}
typedef te-tp-id {
type union {
type uint32;
    // Unnumbered
    type inet:ip-address;
    // IPv4 or IPv6 address
}
description
"An identifier for a TE link endpoint on a node.
This attribute is mapped to a local or remote link identifier
as defined in RFCs 3630 and 5305.";
reference
"RFC 3630: Traffic Engineering (TE) Extensions to OSPF
Version 2
RFC 5305: IS-IS Extensions for Traffic Engineering";

// NOTE: The typedef bandwidth-scientific-notation below has been
// added in this module revision
// RFC Editor: remove the note above and this note

typedef bandwidth-scientific-notation {
type string {
pattern
'0(\.[0]?([eE](\+)?0)?|'
+ '([1-9]\.[0-9]{0,6})([eE](\+)?(9[0-6]|[1-8][0-9]|0?[0-9])?)?';
}
units "bps";
description
"Bandwidth values, expressed using the scientific notation
in bits per second.
The encoding format is the external decimal-significant
character sequences specified in IEEE 754 and ISO/IEC C99
for 32-bit decimal floating-point numbers:
(-1)**(S) * 10**(Exponent) * (Significant),
where Significant uses 7 digits.
An implementation for this representation may use decimal32
or binary32. The range of the Exponent is from -95 to +96
for decimal32, and from -38 to +38 for binary32.
As a bandwidth value, the format is restricted to be
normalized, non-negative, and non-fraction:
n.dddddd{+}dd, N.DDDDDDE{+}DD, 0e0 or 0E0,
where 'd' and 'D' are decimal digits; 'n' and 'N' are
non-zeror decimal digits; 'e' and 'E' indicate a power of ten.
Some examples are 0e0, 1e10, and 9.953e9."
reference
"IEEE Std 754-2008: IEEE Standard for Floating-Point
Arithmetic.
ISO/IEC C99: Information technology - Programming
Languages - C.";
}

/* TE features */

feature p2mp-te {
  description
    "Indicates support for Point-to-Multipoint TE (P2MP-TE).";
  reference
    "RFC 4875: Extensions to Resource Reservation Protocol -
    Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE
    Label Switched Paths (LSPs)";
}

feature frr-te {
  description
    "Indicates support for TE Fast Reroute (FRR).";
  reference
    "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}

feature extended-admin-groups {
  description
    "Indicates support for TE link extended administrative
groups.";
  reference
    "RFC 7308: Extended Administrative Groups in MPLS Traffic
    Engineering (MPLS-TE)";
}

feature named-path-affinities {
  description
    "Indicates support for named path affinities.";
}

feature named-extended-admin-groups {
description
  "Indicates support for named extended administrative groups.";
}

feature named-srlg-groups {
  description
    "Indicates support for named SRLG groups.";
}

feature named-path-constraints {
  description
    "Indicates support for named path constraints.";
}

feature path-optimization-metric {
  description
    "Indicates support for path optimization metrics.";
}

feature path-optimization-objective-function {
  description
    "Indicates support for path optimization objective functions.";
}

identity session-attributes-flags {
  description
    "Base identity for the RSVP-TE session attributes flags.";
}

identity local-protection-desired {
  base session-attributes-flags;
  description
    "Local protection is desired.";
  reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
    Section 4.7.1";
}

identity se-style-desired {
  base session-attributes-flags;
  description
    "Shared explicit style, to allow the LSP to be established
    and share resources with the old LSP.";
  reference

"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity local-recording-desired {
  base session-attributes-flags;
  description
    "Label recording is desired.";
  reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels,
    Section 4.7.1";
}

identity bandwidth-protection-desired {
  base session-attributes-flags;
  description
    "Requests FRR bandwidth protection on LSRs, if present.";
  reference
    "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}

identity node-protection-desired {
  base session-attributes-flags;
  description
    "Requests FRR node protection on LSRs, if present.";
  reference
    "RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels";
}

identity path-reevaluation-request {
  base session-attributes-flags;
  description
    "This flag indicates that a path re-evaluation (of the
    current path in use) is requested. Note that this does
    not trigger any LSP reroutes but instead just signals a
    request to evaluate whether a preferable path exists.";
  reference
    "RFC 4736: Reoptimization of Multiprotocol Label Switching
    (MPLS) Traffic Engineering (TE) Loosely Routed Label Switched
    Path (LSP)";
}

identity soft-preemption-desired {
  base session-attributes-flags;
  description
    "Soft preemption of LSP resources is desired.";
  reference
    "RFC 5712: MPLS Traffic Engineering Soft Preemption";
}
identity lsp-attributes-flags {
    description "Base identity for LSP attributes flags.";
}

identity end-to-end-rerouting-desired {
    base lsp-attributes-flags;
    description "Indicates end-to-end rerouting behavior for an LSP undergoing establishment. This MAY also be used to specify the behavior of end-to-end LSP recovery for established LSPs."
    reference "RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";
}

identity boundary-rerouting-desired {
    base lsp-attributes-flags;
    description "Indicates boundary rerouting behavior for an LSP undergoing establishment. The boundary Area Border Router (ABR) / Autonomous System Border Router (ASBR) can decide to forward the PathErr message upstream to either an upstream boundary ABR/ASBR or the ingress LSR. Alternatively, it can try to select another egress boundary LSR."
    reference "RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)";
}

identity segment-based-rerouting-desired {
    base lsp-attributes-flags;
    description "Indicates segment-based rerouting behavior for an LSP undergoing establishment. This MAY also be used to specify
segment-based LSP recovery for established LSPs.

reference
"RFC 4920: Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"

identity lsp-integrity-required {
  base lsp-attributes-flags;
  description
  "Indicates that LSP integrity is required.";
  reference
  "RFC 4875: Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"

identity contiguous-lsp-desired {
  base lsp-attributes-flags;
  description
  "Indicates that a contiguous LSP is desired.";
  reference
  "RFC 5151: Inter-Domain MPLS and GMPLS Traffic Engineering -- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"

identity lsp-stitching-desired {
  base lsp-attributes-flags;
  description
  "Indicates that LSP stitching is desired.";
  reference
  "RFC 5150: Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)
RFC 7570: Label Switched Path (LSP) Attribute in the Explicit Route Object (ERO)"

identity pre-planned-lsp-flag {
  base lsp-attributes-flags;
description
"Indicates that the LSP MUST be provisioned in the
c control plane only.";
reference
"RFC 6001: Generalized MPLS (GMPLS) Protocol Extensions for
 Multi-Layer and Multi-Region Networks (MLN/MRN)
 RFC 7570: Label Switcched Path (LSP) Attribute in the Explicit
 Route Object (ERO)";
}

identity non-php-behavior-flag {
 base lsp-attributes-flags;
 description
 "Indicates that non-PHP (non-Penultimate Hop Popping) behavior
 for the LSP is desired.";
 reference
 "RFC 6511: Non-Penultimate Hop Popping Behavior and Out-of-Band
 Mapping for RSVP-TE Label Switched Paths
 RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
 Route Object (ERO)";
}

identity oob-mapping-flag {
 base lsp-attributes-flags;
 description
 "Indicates that signaling of the egress binding information is
 out of band (e.g., via the Border Gateway Protocol (BGP)).";
 reference
 "RFC 6511: Non-Penultimate Hop Popping Behavior and Out-of-Band
 Mapping for RSVP-TE Label Switched Paths
 RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
 Route Object (ERO)";
}

identity entropy-label-capability {
 base lsp-attributes-flags;
 description
 "Indicates entropy label capability.";
 reference
 "RFC 6790: The Use of Entropy Labels in MPLS Forwarding
 RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
 Route Object (ERO)";
}

identity oam-mep-entity-desired {
 base lsp-attributes-flags;
 description
 "OAM Maintenance Entity Group End Point (MEP) entities

desired.;
reference
"RFC 7260: GMPLS RSVP-TE Extensions for Operations,
Administration, and Maintenance (OAM) Configuration";
}

identity oam-mip-entity-desired {
base lsp-attributes-flags;
description
"OAM Maintenance Entity Group Intermediate Points (MIP)
entities desired.";
reference
"RFC 7260: GMPLS RSVP-TE Extensions for Operations,
Administration, and Maintenance (OAM) Configuration";
}

identity srlg-collection-desired {
base lsp-attributes-flags;
description
"SRLG collection desired.";
reference
"RFC 7570: Label Switched Path (LSP) Attribute in the Explicit
Route Object (ERO)
RFC 8001: RSVP-TE Extensions for Collecting Shared Risk
Link Group (SRLG) Information";
}

identity loopback-desired {
base lsp-attributes-flags;
description
"This flag indicates that a particular node on the LSP is
required to enter loopback mode. This can also be
used to specify the loopback state of the node.";
reference
"RFC 7571: GMPLS RSVP-TE Extensions for Lock Instruct and
Loopback";
}

identity p2mp-te-tree-eval-request {
base lsp-attributes-flags;
description
"P2MP-TE tree re-evaluation request.";
reference
"RFC 8149: RSVP Extensions for Reoptimization of Loosely Routed
Point-to-Multipoint Traffic Engineering Label Switched Paths
(LSPs)";
}
identity rtm-set-desired {
    base lsp-attributes-flags;
    description
        "Residence Time Measurement (RTM) attribute flag requested.";
    reference
        "RFC 8169: Residence Time Measurement in MPLS Networks";
}

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

identity link-protection-unprotected {
    base link-protection-type;
    description
        "Unprotected link type.";
    reference
        "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity link-protection-extra-traffic {
    base link-protection-type;
    description
        "Extra-Traffic protected link type.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity link-protection-shared {
    base link-protection-type;
    description
        "Shared protected link type.";
    reference
        "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity link-protection-1-for-1 {
    base link-protection-type;
    description
        "One-for-one (1:1) protected link type.";
    reference
        "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}
identity link-protection-1-plus-1 {
  base link-protection-type;
  description
    "One-plus-one (1+1) protected link type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity link-protection-enhanced {
  base link-protection-type;
  description
    "A compound link protection type derived from the underlay TE tunnel protection configuration supporting the TE link.";
}

identity association-type {
  description
    "Base identity for the tunnel association.";
}

identity association-type-recovery {
  base association-type;
  description
    "Association type for recovery, used to associate LSPs of the same tunnel for recovery.";
  reference
}

identity association-type-resource-sharing {
  base association-type;
  description
    "Association type for resource sharing, used to enable resource sharing during make-before-break.";
  reference
    "RFC 4873: GMPLS Segment Recovery RFC 6780: RSVP ASSOCIATION Object Extensions";
}

identity association-type-double-sided-bidir {
  base association-type;
  description
    "Association type for double-sided bidirectional LSPs, used to associate two LSPs of two tunnels that are independently configured on either endpoint.";
identity association-type-single-sided-bidir {
  base association-type;
  description
    "Association type for single-sided bidirectional LSPs,
    used to associate two LSPs of two tunnels, where one
    tunnel is configured on one side/endpoint and the other
    tunnel is dynamically created on the other endpoint.";
  reference
    "RFC 6780: RSVP ASSOCIATION Object Extensions
    RFC 7551: RSVP-TE Extensions for Associated Bidirectional
    Label Switched Paths (LSPs)";
}

identity objective-function-type {
  description
    "Base objective function type.";
}

identity of-minimize-cost-path {
  base objective-function-type;
  description
    "Objective function for minimizing path cost.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP)";
}

identity of-minimize-load-path {
  base objective-function-type;
  description
    "Objective function for minimizing the load on one or more
    paths.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP)";
}

identity of-maximize-residual-bandwidth {
  base objective-function-type;
  description
    "Objective function for maximizing residual bandwidth.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
    Computation Element Communication Protocol (PCEP)";
}
Computation Element Communication Protocol (PCEP)

identity of-minimize-agg-bandwidth-consumption {
  base objective-function-type;
  description
    "Objective function for minimizing aggregate bandwidth consumption.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)"
}

identity of-minimize-load-most-loaded-link {
  base objective-function-type;
  description
    "Objective function for minimizing the load on the link that is carrying the highest load.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)"
}

identity of-minimize-cost-path-set {
  base objective-function-type;
  description
    "Objective function for minimizing the cost on a path set.";
  reference
    "RFC 5541: Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP)"
}

identity path-computation-method {
  description
    "Base identity for supported path computation mechanisms.";
}

identity path-locally-computed {
  base path-computation-method;
  description
    "Indicates a constrained-path LSP in which the path is computed by the local LER.";
  reference
    "RFC 3272: Overview and Principles of Internet Traffic Engineering, Section 5.4"
}

identity path-externally-queried {
base path-computation-method;

description
"Constrained-path LSP in which the path is obtained by querying an external source, such as a PCE server. In the case that an LSP is defined to be externally queried, it may also have associated explicit definitions (provided to the external source to aid computation). The path that is returned by the external source may require further local computation on the device."

reference
"RFC 3272: Overview and Principles of Internet Traffic Engineering
RFC 4657: Path Computation Element (PCE) Communication Protocol Generic Requirements";

} identity path-explicitly-defined {

base path-computation-method;

description
"Constrained-path LSP in which the path is explicitly specified as a collection of strict and/or loose hops."

reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 3272: Overview and Principles of Internet Traffic Engineering";

}

identity lsp-metric-type {

description
"Base identity for the LSP metric specification types.";

}

identity lsp-metric-relative {

base lsp-metric-type;

description
"The metric specified for the LSPs to which this identity refers is specified as a value relative to the IGP metric cost to the LSP’s tail end."

reference
"RFC 4657: Path Computation Element (PCE) Communication Protocol Generic Requirements";

}

identity lsp-metric-absolute {

base lsp-metric-type;

description
"The metric specified for the LSPs to which this identity
refers is specified as an absolute value.";
reference
"RFC 4657: Path Computation Element (PCE) Communication Protocol Generic Requirements";
}

identity lsp-metric-inherited {
    base lsp-metric-type;
    description
    "The metric for the LSPs to which this identity refers is not specified explicitly; rather, it is directly inherited from the IGP cost.";
    reference
    "RFC 4657: Path Computation Element (PCE) Communication Protocol Generic Requirements";
}

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

identity te-tunnel-p2p {
    base te-tunnel-type;
    description
    "TE Point-to-Point (P2P) tunnel type.";
    reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity te-tunnel-p2mp {
    base te-tunnel-type;
    description
    "TE P2MP tunnel type.";
    reference
    "RFC 4875: Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)";
}

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

identity tunnel-action-resetup {
    base tunnel-action-type;
description
"TE tunnel action that tears down the tunnel’s current LSP (if any) and attempts to re-establish a new LSP."
}

identity tunnel-action-reoptimize {
    base tunnel-action-type;
    description
    "TE tunnel action that reoptimizes the placement of the tunnel LSP(s)."
}

identity tunnel-action-switchpath {
    base tunnel-action-type;
    description
    "TE tunnel action that switches the tunnel’s LSP to use the specified path."
}

identity te-action-result {
    description
    "Base identity from which specific TE action results are derived."
}

identity te-action-success {
    base te-action-result;
    description
    "TE action was successful."
}

identity te-action-fail {
    base te-action-result;
    description
    "TE action failed."
}

identity tunnel-action-inprogress {
    base te-action-result;
    description
    "TE action is in progress."
}

identity tunnel-admin-state-type {
    description
    "Base identity for TE tunnel administrative states."
}
identity tunnel-admin-state-up {
    base tunnel-admin-state-type;
    description
        "Tunnel’s administrative state is up.";
}

identity tunnel-admin-state-down {
    base tunnel-admin-state-type;
    description
        "Tunnel’s administrative state is down.";
}

identity tunnel-state-type {
    description
        "Base identity for TE tunnel states.";
}

identity tunnel-state-up {
    base tunnel-state-type;
    description
        "Tunnel’s state is up.";
}

identity tunnel-state-down {
    base tunnel-state-type;
    description
        "Tunnel’s state is down.";
}

identity lsp-state-type {
    description
        "Base identity for TE LSP states.";
}

identity lsp-path-computing {
    base lsp-state-type;
    description
        "State path computation is in progress.";
}

identity lsp-path-computation-ok {
    base lsp-state-type;
    description
        "State path computation was successful.";
}

identity lsp-path-computation-failed {
    base lsp-state-type;
}
description
    "State path computation failed.";
}

identity lsp-state-setting-up {
    base lsp-state-type;
    description
        "State is being set up.";
}

identity lsp-state-setup-ok {
    base lsp-state-type;
    description
        "State setup was successful.";
}

identity lsp-state-setup-failed {
    base lsp-state-type;
    description
        "State setup failed.";
}

identity lsp-state-up {
    base lsp-state-type;
    description
        "State is up.";
}

identity lsp-state-tearing-down {
    base lsp-state-type;
    description
        "State is being torn down.";
}

identity lsp-state-down {
    base lsp-state-type;
    description
        "State is down.";
}

identity path-invalidation-action-type {
    description
        "Base identity for TE path invalidation action types.";
}

identity path-invalidation-action-drop {
    base path-invalidation-action-type;
    description
"Upon invalidation of the TE tunnel path, the tunnel remains valid, but any packet mapped over the tunnel is dropped.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 2.5";

identity path-invalidation-action-teardown {
  base path-invalidation-action-type;
  description
    "TE path invalidation action teardown.";
  reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 2.5";
}

identity lsp-restoration-type {
  description
    "Base identity from which LSP restoration types are derived.";
}

identity lsp-restoration-restore-any {
  base lsp-restoration-type;
  description
    "Any LSP affected by a failure is restored.";
}

identity lsp-restoration-restore-all {
  base lsp-restoration-type;
  description
    "Affected LSPs are restored after all LSPs of the tunnel are broken.";
}

identity restoration-scheme-type {
  description
    "Base identity for LSP restoration schemes.";
}

identity restoration-scheme-preconfigured {
  base restoration-scheme-type;
  description
    "Restoration LSP is preconfigured prior to the failure.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}
identity restoration-scheme-precomputed {
  base restoration-scheme-type;
  description "Restoration LSP is precomputed prior to the failure.";
  reference "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity restoration-scheme-presignaled {
  base restoration-scheme-type;
  description "Restoration LSP is presignaled prior to the failure.";
  reference "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity lsp-protection-type {
  description "Base identity from which LSP protection types are derived.";
}

identity lsp-protection-unprotected {
  base lsp-protection-type;
  description "'Unprotected' LSP protection type.";
}

identity lsp-protection-reroute-extra {
  base lsp-protection-type;
  description "'(Full) Rerouting' LSP protection type.";
}

identity lsp-protection-reroute {
  base lsp-protection-type;
  description "'Rerouting without Extra-Traffic' LSP protection type.";
}
reference
"RFC 4872: RSVP-TE Extensions in Support of End-to-End
Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-1-for-n {
  base lsp-protection-type;
  description
    "'1:N Protection with Extra-Traffic' LSP protection type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
    Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-1-for-1 {
  base lsp-protection-type;
  description
    "LSP protection '1:1 Protection Type'.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
    Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-unidir-1-plus-1 {
  base lsp-protection-type;
  description
    "'1+1 Unidirectional Protection' LSP protection type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
    Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-bidir-1-plus-1 {
  base lsp-protection-type;
  description
    "'1+1 Bidirectional Protection' LSP protection type.";
  reference
    "RFC 4872: RSVP-TE Extensions in Support of End-to-End
    Generalized Multi-Protocol Label Switching (GMPLS) Recovery";
}

identity lsp-protection-extra-traffic {
  base lsp-protection-type;
  description
    "Extra-Traffic LSP protection type.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
identity lsp-protection-state {
    description
    "Base identity of protection states for reporting purposes.";
}

identity normal {
    base lsp-protection-state;
    description
    "Normal state.";
}

identity signal-fail-of-protection {
    base lsp-protection-state;
    description
    "The protection transport entity has a signal fail condition
    that is of higher priority than the forced switchover
    command.";
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity lockout-of-protection {
    base lsp-protection-state;
    description
    "A Loss of Protection (LoP) command is active.";
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity forced-switch {
    base lsp-protection-state;
    description
    "A forced switchover command is active.";
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity signal-fail {
    base lsp-protection-state;
    description
    "There is a signal fail condition on either the working path
    or the protection path.";
    reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
    for Generalized Multi-Protocol Label Switching (GMPLS)";
}
"RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";

identity signal-degrade {
  base lsp-protection-state;
  description
    "There is a signal degrade condition on either the working path or the protection path.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity manual-switch {
  base lsp-protection-state;
  description
    "A manual switchover command is active.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity wait-to-restore {
  base lsp-protection-state;
  description
    "A WTR timer is running.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity do-not-revert {
  base lsp-protection-state;
  description
    "A Do Not Revert (DNR) condition is active because of non-revertive behavior.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity failure-of-protocol {
  base lsp-protection-state;
  description
    "LSP protection is not working because of a protocol failure condition.";
  reference
"RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity protection-external-commands {
  description
  "Base identity from which protection-related external commands used for troubleshooting purposes are derived."
}

identity action-freeze {
  base protection-external-commands;
  description
  "A temporary configuration action initiated by an operator command that prevents any switchover action from being taken and, as such, freezes the current state."
  reference
  "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity clear-freeze {
  base protection-external-commands;
  description
  "An action that clears the active freeze state."
  reference
  "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity action-lockout-of-normal {
  base protection-external-commands;
  description
  "A temporary configuration action initiated by an operator command to ensure that the normal traffic is not allowed to use the protection transport entity."
  reference
  "RFC 4427: Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)"
}

identity clear-lockout-of-normal {
  base protection-external-commands;
  description
  "An action that clears the active lockout of the normal state."
  reference
  "RFC 4427: Recovery (Protection and Restoration) Terminology
identity action-lockout-of-protection {
    base protection-external-commands;
    description
        "A temporary configuration action initiated by an operator
        command to ensure that the protection transport entity is
        temporarily not available to transport a traffic signal
        (either normal or Extra-Traffic).";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity action-forced-switch {
    base protection-external-commands;
    description
        "A switchover action initiated by an operator command to switch
        the Extra-Traffic signal, the normal traffic signal, or the
        null signal to the protection transport entity, unless a
        switchover command of equal or higher priority is in effect.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity action-manual-switch {
    base protection-external-commands;
    description
        "A switchover action initiated by an operator command to switch
        the Extra-Traffic signal, the normal traffic signal, or
        the null signal to the protection transport entity, unless a
        fault condition exists on other transport entities or a
        switchover command of equal or higher priority is in effect.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity action-exercise {
    base protection-external-commands;
    description
        "An action that starts testing whether or not APS communication
        is operating correctly. It is of lower priority than any
        other state or command.";
    reference
        "RFC 4427: Recovery (Protection and Restoration) Terminology
        for Generalized Multi-Protocol Label Switching (GMPLS)";
}
for Generalized Multi-Protocol Label Switching (GMPLS);"
}

identity clear {
  base protection-external-commands;
  description
    "An action that clears the active near-end lockout of a
     protection, forced switchover, manual switchover, WTR state,
     or exercise command.";
  reference
    "RFC 4427: Recovery (Protection and Restoration) Terminology
     for Generalized Multi-Protocol Label Switching (GMPLS)";
}

identity switching-capabilities {
  description
    "Base identity for interface switching capabilities.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
     Signaling Functional Description";
}

identity switching-psc1 {
  base switching-capabilities;
  description
    "Packet-Switch Capable-1 (PSC-1).";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
     Signaling Functional Description";
}

identity switching-evpl {
  base switching-capabilities;
  description
    "Ethernet Virtual Private Line (EVPL).";
  reference
    "RFC 6004: Generalized MPLS (GMPLS) Support for Metro Ethernet
     Forum and G.8011 Ethernet Service Switching";
}

identity switching-l2sc {
  base switching-capabilities;
  description
    "Layer-2 Switch Capable (L2SC).";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
     Signaling Functional Description";
}
identity switching-tdm {
    base switching-capabilities;
    description
        "Time-Division-Multiplex Capable (TDM).";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
         Signaling Functional Description";
}

identity switching-otn {
    base switching-capabilities;
    description
        "OTN-TDM capable.";
    reference
        "RFC 7138: Traffic Engineering Extensions to OSPF for GMPLS
         Control of Evolving G.709 Optical Transport Networks";
}

identity switching-dcsc {
    base switching-capabilities;
    description
        "Data Channel Switching Capable (DCSC).";
    reference
        "RFC 6002: Generalized MPLS (GMPLS) Data Channel
         Switching Capable (DCSC) and Channel Set Label Extensions";
}

identity switching-lsc {
    base switching-capabilities;
    description
        "Lambda-Switch Capable (LSC).";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
         Signaling Functional Description";
}

identity switching-fsc {
    base switching-capabilities;
    description
        "Fiber-Switch Capable (FSC).";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
         Signaling Functional Description";
}

identity lsp-encoding-types {
    description
        "Base identity for encoding types.";
}
reference
"RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
Signaling Functional Description";
}

identity lsp-encoding-packet {
  base lsp-encoding-types;
  description
    "Packet LSP encoding.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-ethernet {
  base lsp-encoding-types;
  description
    "Ethernet LSP encoding.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-pdh {
  base lsp-encoding-types;
  description
    "ANSI/ETSI PDH LSP encoding.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-sdh {
  base lsp-encoding-types;
  description
    "SDH ITU-T G.707 / SONET ANSI T1.105 LSP encoding.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}

identity lsp-encoding-digital-wrapper {
  base lsp-encoding-types;
  description
    "Digital Wrapper LSP encoding.";
  reference
    "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
    Signaling Functional Description";
}
identity lsp-encoding-lambda {
    base lsp-encoding-types;
    description
        "Lambda (photonic) LSP encoding.";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
        Signaling Functional Description";
}

identity lsp-encoding-fiber {
    base lsp-encoding-types;
    description
        "Fiber LSP encoding.";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
        Signaling Functional Description";
}

identity lsp-encoding-fiber-channel {
    base lsp-encoding-types;
    description
        "FiberChannel LSP encoding.";
    reference
        "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
        Signaling Functional Description";
}

identity lsp-encoding-oduk {
    base lsp-encoding-types;
    description
        "G.709 ODUk (Digital Path) LSP encoding.";
    reference
        "RFC 4328: Generalized Multi-Protocol Label Switching (GMPLS)
        Signaling Extensions for G.709 Optical Transport Networks
        Control";
}

identity lsp-encoding-optical-channel {
    base lsp-encoding-types;
    description
        "G.709 Optical Channel LSP encoding.";
    reference
        "RFC 4328: Generalized Multi-Protocol Label Switching (GMPLS)
        Signaling Extensions for G.709 Optical Transport Networks
        Control";
}
identity lsp-encoding-line {
    base lsp-encoding-types;
    description
        "Line (e.g., 8B/10B) LSP encoding.";
    reference
        "RFC 6004: Generalized MPLS (GMPLS) Support for Metro
         Ethernet Forum and G.8011 Ethernet Service Switching";
}

identity path-signaling-type {
    description
        "Base identity from which specific LSP path setup types
         are derived.";
}

identity path-setup-static {
    base path-signaling-type;
    description
        "Static LSP provisioning path setup.";
}

identity path-setup-rsvp {
    base path-signaling-type;
    description
        "RSVP-TE signaling path setup.";
    reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity path-setup-sr {
    base path-signaling-type;
    description
        "Segment-routing path setup.";
}

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

identity path-scope-segment {
    base path-scope-type;
    description
        "Path scope segment.";
    reference
        "RFC 4873: GMPLS Segment Recovery";
}
identity path-scope-end-to-end {
  base path-scope-type;
  description
    "Path scope end to end.";
  reference
    "RFC 4873: GMPLS Segment Recovery";
}

identity route-usage-type {
  description
    "Base identity for route usage.";
}

identity route-include-object {
  base route-usage-type;
  description
    "Include route' object.";
}

identity route-exclude-object {
  base route-usage-type;
  description
    "Exclude route' object.";
  reference
}

identity route-exclude-srlg {
  base route-usage-type;
  description
    "Excludes SRLGs.";
  reference
}

identity path-metric-type {
  description
    "Base identity for the path metric type.";
}

identity path-metric-te {
  base path-metric-type;
  description
    "TE path metric.";
  reference
    "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a
second MPLS Traffic Engineering (TE) Metric";
}

identity path-metric-igp {
  base path-metric-type;
  description
    "IGP path metric."
  reference
    "RFC 3785: Use of Interior Gateway Protocol (IGP) Metric as a second MPLS Traffic Engineering (TE) Metric";
}

identity path-metric-hop {
  base path-metric-type;
  description
    "Hop path metric."
}

identity path-metric-delay-average {
  base path-metric-type;
  description
    "Average unidirectional link delay."
  reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}

identity path-metric-delay-minimum {
  base path-metric-type;
  description
    "Minimum unidirectional link delay."
  reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}

identity path-metric-residual-bandwidth {
  base path-metric-type;
  description
    "Unidirectional Residual Bandwidth, which is defined to be Maximum Bandwidth (RFC 3630) minus the bandwidth currently allocated to LSPs."
  reference
    "RFC 3630: Traffic Engineering (TE) Extensions to OSPF Version 2
    RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions";
}

identity path-metric-optimize-includes {
  base path-metric-type;
description
   "A metric that optimizes the number of included resources specified in a set.";
}

identity path-metric-optimize-excludes {
    base path-metric-type;
    description
       "A metric that optimizes to a maximum the number of excluded resources specified in a set.";
}

identity path-tiebreaker-type {
    description
       "Base identity for the path tiebreaker type.";
}

identity path-tiebreaker-minfill {
    base path-tiebreaker-type;
    description
       "Min-Fill LSP path placement.";
}

identity path-tiebreaker-maxfill {
    base path-tiebreaker-type;
    description
       "Max-Fill LSP path placement.";
}

identity path-tiebreaker-random {
    base path-tiebreaker-type;
    description
       "Random LSP path placement.";
}

identity resource-affinities-type {
    description
       "Base identity for resource class affinities.";
    reference
       "RFC 2702: Requirements for Traffic Engineering Over MPLS";
}

identity resource-aff-include-all {
    base resource-affinities-type;
    description
       "The set of attribute filters associated with a tunnel, all of which must be present for a link to be acceptable.";
}
reference
"RFC 2702: Requirements for Traffic Engineering Over MPLS
RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity resource-aff-include-any {
  base resource-affinities-type;
description
  "The set of attribute filters associated with a
  tunnel, any of which must be present for a link
  to be acceptable.";
reference
  "RFC 2702: Requirements for Traffic Engineering Over MPLS
  RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity resource-aff-exclude-any {
  base resource-affinities-type;
description
  "The set of attribute filters associated with a
  tunnel, any of which renders a link unacceptable.";
reference
  "RFC 2702: Requirements for Traffic Engineering Over MPLS
  RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}

identity te-optimization-criterion {
  description
    "Base identity for the TE optimization criteria.";
reference
  "RFC 3272: Overview and Principles of Internet Traffic
  Engineering";
}

identity not-optimized {
  base te-optimization-criterion;
description
  "Optimization is not applied.";
}

identity cost {
  base te-optimization-criterion;
description
  "Optimized on cost.";
reference
  "RFC 5541: Encoding of Objective Functions in the Path
  Computation Element Communication Protocol (PCEP)";
}
identity delay {
  base te-optimization-criterion;
  description
    "Optimized on delay."
  reference
    "RFC 5541: Encoding of Objective Functions in the Path
     Computation Element Communication Protocol (PCEP)"
}

identity path-computation-srlg-type {
  description
    "Base identity for SRLG path computation."
}

identity srlg-ignore {
  base path-computation-srlg-type;
  description
    "Ignores SRLGs in the path computation."
}

identity srlg-strict {
  base path-computation-srlg-type;
  description
    "Includes a strict SRLG check in the path computation."
}

identity srlg-preferred {
  base path-computation-srlg-type;
  description
    "Includes a preferred SRLG check in the path computation."
}

identity srlg-weighted {
  base path-computation-srlg-type;
  description
    "Includes a weighted SRLG check in the path computation."
}

/**
 * TE bandwidth groupings
 **/

grouping te-bandwidth {
  description
    "This grouping defines the generic TE bandwidth.
     For some known data-plane technologies, specific modeling
     structures are specified. The string-encoded ‘te-bandwidth’
     type is used for unspecified technologies."
The modeling structure can be augmented later for other technologies.

container te-bandwidth {
  description
  "Container that specifies TE bandwidth. The choices can be augmented for specific data-plane technologies.";
  choice technology {
    default "generic";
    description
    "Data-plane technology type.";
    case generic {
      leaf generic {
        type te-bandwidth;
        description
        "Bandwidth specified in a generic format.";
      }
    }
  }
}

/**
 * TE label groupings
 **/

grouping te-label {
  description
  "This grouping defines the generic TE label. The modeling structure can be augmented for each technology. For unspecified technologies, 'rt-types:generalized-label' is used.";
  container te-label {
    description
    "Container that specifies the TE label. The choices can be augmented for specific data-plane technologies.";
    choice technology {
      default "generic";
      description
      "Data-plane technology type.";
      case generic {
        leaf generic {
          type rt-types:generalized-label;
          description
          "TE label specified in a generic format.";
        }
      }
    }
  }
type te-label-direction;
default "forward";
description
  "Label direction."
}
}

grouping te-topology-identifier {
  description
   "Augmentation for a TE topology.";
  container te-topology-identifier {
    description
      "TE topology identifier container.";
    leaf provider-id {
      type te-global-id;
      default "0";
      description
        "An identifier to uniquely identify a provider.
         If omitted, it assumes that the topology provider ID
         value = 0 (the default).";
    }
    leaf client-id {
      type te-global-id;
      default "0";
      description
        "An identifier to uniquely identify a client.
         If omitted, it assumes that the topology client ID
         value = 0 (the default).";
    }
    leaf topology-id {
      type te-topology-id;
      default "";
      description
        "When the datastore contains several topologies,
         'topology-id' distinguishes between them. If omitted,
         the default (empty) string for this leaf is assumed.";
    }
  }
}

/**
 * TE performance metrics groupings
 **/

grouping performance-metrics-one-way-delay-loss {
  description
    "Performance Metrics (PM) information in real time that can
be applicable to links or connections. PM defined in this

grouping are applicable to generic TE PM as well as packet TE
PM.";

reference
"RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
RFC 7823: Performance-Based Path Selection for Explicitly
Routed Label Switched Paths (LSPs) Using TE Metric
Extensions
RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";

leaf one-way-delay {
  type uint32 {
    range "0..16777215";
  }
  description
    "One-way delay or latency in microseconds.";
}
leaf one-way-delay-normality {
  type te-types:performance-metrics-normality;
  description
    "One-way delay normality.";
}
}

grouping performance-metrics-two-way-delay-loss {
  description
    "PM information in real time that can be applicable to links or
    connections. PM defined in this grouping are applicable to
generic TE PM as well as packet TE PM.";

reference
"RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions
RFC 7823: Performance-Based Path Selection for Explicitly
Routed Label Switched Paths (LSPs) Using TE Metric
Extensions
RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";

leaf two-way-delay {
  type uint32 {
    range "0..16777215";
  }
  description
    "Two-way delay or latency in microseconds.";
}
leaf two-way-delay-normality {
  type te-types:performance-metrics-normality;
  description
    "Two-way delay normality.";
}
}
grouping performance-metrics-one-way-bandwidth {
  description
    "PM information in real time that can be applicable to links. 
    PM defined in this grouping are applicable to generic TE PM 
    as well as packet TE PM.";
  reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions 
    RFC 7823: Performance-Based Path Selection for Explicitly 
    Routed Label Switched Paths (LSPs) Using TE Metric 
    Extensions 
    RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
  leaf one-way-residual-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description
      "Residual bandwidth that subtracts tunnel reservations from 
      Maximum Bandwidth (or link capacity) (RFC 3630) and 
      provides an aggregated remainder across QoS classes.";
    reference
      "RFC 3630: Traffic Engineering (TE) Extensions to OSPF 
      Version 2";
  }
  leaf one-way-residual-bandwidth-normality {
    type te-types:performance-metrics-normality;
    default "normal";
    description
      "Residual bandwidth normality.";
  }
  leaf one-way-available-bandwidth {
    type rt-types:bandwidth-ieee-float32;
    units "bytes per second";
    default "0x0p0";
    description
      "Available bandwidth that is defined to be residual 
      bandwidth minus the measured bandwidth used for the 
      actual forwarding of non-RSVP-TE LSP packets. For a 
      bundled link, available bandwidth is defined to be the 
      sum of the component link available bandwidths.";
  }
  leaf one-way-available-bandwidth-normality {
    type te-types:performance-metrics-normality;
    default "normal";
    description
      "Available bandwidth normality.";
  }
  leaf one-way-utilized-bandwidth {
    type rt-types:bandwidth-ieee-float32;


leaf one-way-utilized-bandwidth-normality {
  type te-types:performance-metrics-normality;
  default "normal";
  description
    "Bandwidth utilization normality.";
}


leaf one-way-delay {
  type uint32 {
    range "0..16777215";
  }
  default "0";
  description
    "One-way delay or latency in microseconds.";
}
leaf one-way-residual-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
    "Residual bandwidth that subtracts tunnel reservations from
     Maximum Bandwidth (or link capacity) (RFC 3630) and
     provides an aggregated remainder across QoS classes.";
  reference
    "RFC 3630: Traffic Engineering (TE) Extensions to OSPF
     Version 2";
}
leaf one-way-available-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
    "Available bandwidth that is defined to be residual
     bandwidth minus the measured bandwidth used for the
     actual forwarding of non-RSVP-TE LSP packets. For a
     bundled link, available bandwidth is defined to be the
grouping two-way-performance-metrics {
  description
  "Two-way PM throttle grouping.";
  leaf two-way-delay {
    type uint32 {
      range "0..16777215";
    }
    default "0";
    description
    "Two-way delay or latency in microseconds.";
  }
}

grouping performance-metrics-thresholds {
  description
  "Grouping for configurable thresholds for measured attributes.";
  uses one-way-performance-metrics;
  uses two-way-performance-metrics;
}

grouping performance-metrics-attributes {
  description
  "Contains PM attributes.";
  container performance-metrics-one-way {
    description
    "One-way link performance information in real time.";
    reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions"
    RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions
    RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
    uses performance-metrics-one-way-delay-loss;
}


uses performance-metrics-one-way-bandwidth;
}
container performance-metrics-two-way {
  description
    "Two-way link performance information in real time.";
  reference
    "RFC 6374: Packet Loss and Delay Measurement for MPLS Networks";
  uses performance-metrics-two-way-delay-loss;
}

grouping performance-metrics-throttle-container {
  description
    "Controls PM throttling.";
  container throttle {
    must 'suppression-interval >= measure-interval' {
      error-message "'suppression-interval' cannot be less than "
        + "'measure-interval'.";
      description
        "Constraint on 'suppression-interval' and 'measure-interval'.";
    }
  }
  description
    "Link performance information in real time.";
  reference
    "RFC 7471: OSPF Traffic Engineering (TE) Metric Extensions"
    "RFC 7823: Performance-Based Path Selection for Explicitly Routed Label Switched Paths (LSPs) Using TE Metric Extensions"
    "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions";
  leaf one-way-delay-offset {
    type uint32 {
      range "0..16777215";
    }
    default "0";
    description
      "Offset value to be added to the measured delay value.";
  }
  leaf measure-interval {
    type uint32;
    default "30";
    description
      "Interval, in seconds, to measure the extended metric values.";
  }
  leaf advertisement-interval {
    type uint32;
default "0";
description
 "Interval, in seconds, to advertise the extended metric values.";
}

leaf suppression-interval {
  type uint32 {
    range "1..max";
  }
  default "120";
  description
  "Interval, in seconds, to suppress advertisement of the extended metric values.";
  reference
  "RFC 8570: IS-IS Traffic Engineering (TE) Metric Extensions, Section 6";
}

container threshold-out {
  uses performance-metrics-thresholds;
  description
  "If the measured parameter falls outside an upper bound for all but the minimum-delay metric (or a lower bound for the minimum-delay metric only) and the advertised value is not already outside that bound, an 'anomalous' announcement (anomalous bit set) will be triggered.";
}

container threshold-in {
  uses performance-metrics-thresholds;
  description
  "If the measured parameter falls inside an upper bound for all but the minimum-delay metric (or a lower bound for the minimum-delay metric only) and the advertised value is not already inside that bound, a 'normal' announcement (anomalous bit cleared) will be triggered.";
}

container threshold-accelerated-advertisement {
  description
  "When the difference between the last advertised value and the current measured value exceeds this threshold, an 'anomalous' announcement (anomalous bit set) will be triggered.";
  uses performance-metrics-thresholds;
}
}
}
grouping explicit-route-hop {
    description "The explicit route entry grouping.";
    choice type {
        description "The explicit route entry type.";
        case numbered-node-hop {
            container numbered-node-hop {
                leaf node-id {
                    type te-node-id;
                    mandatory true;
                    description "The identifier of a node in the TE topology.";
                }
                leaf hop-type {
                    type te-hop-type;
                    default "strict";
                    description "Strict or loose hop.";
                }
            }
            description "Numbered node route hop.";
        }
        case numbered-link-hop {
            container numbered-link-hop {
                leaf link-tp-id {
                    type te-tp-id;
                    mandatory true;
                    description "TE Link Termination Point (LTP) identifier.";
                }
                leaf hop-type {
                    type te-hop-type;
                    default "strict";
                    description "Strict or loose hop.";
                }
                leaf direction {
                    type te-link-direction;
                    default "outgoing";
                }
            }
        }
    }
}
"Link route object direction.";
)

description
"Unnumbered link explicit route hop.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels, Section 4.3, EXPLICIT_ROUTE in RSVP-TE
RFC 3477: Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)";
case as-number {
    container as-number-hop {
        leaf as-number {
            type inet:as-number;
            mandatory true;
            description
                "The Autonomous System (AS) number.";
        }
        leaf hop-type {
            type te-hop-type;
            default "strict";
            description
                "Strict or loose hop.";
            description
                "AS explicit route hop.";
        }
    }
}

case label {
    container label-hop {
        description
            "Label hop type."
        uses te-label;
        description
            "The label explicit route hop type.";
    }
}

grouping record-route-state {
    description
        "The Record Route grouping.";
    leaf index {
        type uint32;
        description
            "Record Route hop index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values.";
    }
    choice type {
        description
            "The Record Route entry type.";
        case numbered-node-hop {
            container numbered-node-hop {
                description
                    "Numbered node route hop container.";
            }
        }
    }
}
leaf node-id {
  type te-node-id;
  mandatory true;
  description
    "The identifier of a node in the TE topology.";
}
leaf-list flags {
  type path-attribute-flags;
  description
    "Path attributes flags.";
  reference
    "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
    RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
    RFC 4561: Definition of a Record Route Object (RRO)
    Node-Id Sub-Object";
}
}

description
  "Numbered node route hop.";
}
case numbered-link-hop {
  container numbered-link-hop {
    description
      "Numbered link route hop container.";
    leaf link-tp-id {
      type te-tp-id;
      mandatory true;
      description
        "Numbered TE LTP identifier.";
    }
    leaf-list flags {
      type path-attribute-flags;
      description
        "Path attributes flags.";
      reference
        "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
        RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
        RFC 4561: Definition of a Record Route Object (RRO)
        Node-Id Sub-Object";
    }
    }
  }
  description
    "Numbered link route hop.";
}
case unnumbered-link-hop {
  container unnumbered-link-hop {

leaf link-tp-id {
  type te-tp-id;
  mandatory true;
  description "TE LTP identifier. The combination of the TE link ID and the TE node ID is used to identify an unnumbered TE link.";
}
leaf node-id {
  type te-node-id;
  description "The identifier of a node in the TE topology.";
}
leaf-list flags {
  type path-attribute-flags;
  description "Path attributes flags.";
  reference "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
RFC 4561: Definition of a Record Route Object (RRO)
Node-Id Sub-Object";
}
description "Unnumbered link Record Route hop.";
reference "RFC 3477: Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)";
description "Unnumbered link route hop.";
}
case label {
  container label-hop {
    description "Label route hop type.";
    uses te-label;
    leaf-list flags {
      type path-attribute-flags;
      description "Path attributes flags.";
      reference "RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels
RFC 4090: Fast Reroute Extensions to RSVP-TE for LSP Tunnels
RFC 4561: Definition of a Record Route Object (RRO)
Node-Id Sub-Object";
  }
grouping label-restriction-info {
  description
    "Label set item information.";
  leaf restriction {
    type enumeration {
      enum inclusive {
        description
          "The label or label range is inclusive.";
      }
      enum exclusive {
        description
          "The label or label range is exclusive.";
      }
    }
    default "inclusive";
    description
      "Indicates whether the list item is inclusive or exclusive.";
  }
  leaf index {
    type uint32;
    description
      "The index of the label restriction list entry.";
  }
  container label-start {
    must "!(../label-end/te-label/direction) and"
      + " !not(te-label/direction)
      + " or 
      + "!(../label-end/te-label/direction = te-label/direction)
      + " or 
      + "!(not(te-label/direction) and"
      + " !!(../label-end/te-label/direction = 'forward'))
      + " or 
      + "!(not(!../label-end/te-label/direction) and"
      + " !!(te-label/direction = 'forward'))" {
      error-message "'label-start' and 'label-end' must have the "
        + "same direction.";
    }
    description
      "This is the starting label if a label range is specified.
        This is the label value if a single label is specified,
in which case the 'label-end' attribute is not set.

container label-end {
  uses te-label;
}

container label-step {
  uses te-label;
}

leaf range-bitmap {
  type yang:hex-string;
  description
      "When there are gaps between 'label-start' and 'label-end',
       this attribute is used to specify the positions
       of the gaps."
};

of the used labels. This is represented in big endian as 'hex-string'.
The most significant byte in the hex-string is the farthest to the left in the byte sequence. Leading zero bytes in the configured value may be omitted for brevity.
Each bit position in the 'range-bitmap' 'hex-string' maps to a label in the range derived from 'label-start'.

For example, assuming that 'label-start' = 16000 and 'range-bitmap' = 0x01000001, then:

- bit position (0) is set, and the corresponding mapped label from the range is 16000 + (0 * 'label-step') or 16000 for default 'label-step' = 1.
- bit position (24) is set, and the corresponding mapped label from the range is 16000 + (24 * 'label-step') or 16024 for default 'label-step' = 1.

```yang
grouping label-set-info {
  description
      "Grouping for the list of label restrictions specifying what labels may or may not be used.";
  container label-restrictions {
    description
      "The label restrictions container.";
    list label-restriction {
      key "index";
      description
        "The absence of the label restrictions container implies that all labels are acceptable; otherwise, only restricted labels are available.";
      reference
        "RFC 7579: General Network Element Constraint Encoding for GMPLS-Controlled Networks";
      uses label-restriction-info;
    }
  }
}
```
leaf weight {
  type uint8;
  default "1";
  description
    "TE path metric normalization weight.";
}

container explicit-route-exclude-objects {
  when ".../metric-type = "
    + "'te-types:path-metric-optimize-excludes'";
  description
    "Container for the 'exclude route' object list.";
  uses path-route-exclude-objects;
}

container explicit-route-include-objects {
  when ".../metric-type = "
    + "'te-types:path-metric-optimize-includes'";
  description
    "Container for the 'include route' object list.";
  uses path-route-include-objects;
}

grouping common-constraints {
  description
    "Common constraints grouping that can be set on
    a constraint set or directly on the tunnel.";
  uses te-bandwidth {
    description
      "A requested bandwidth to use for path computation.";
  }

  leaf link-protection {
    type identityref {
      base link-protection-type;
    }
    default "te-types:link-protection-unprotected";
    description
      "Link protection type required for the links included
      in the computed path.";
    reference
      "RFC 4202: Routing Extensions in Support of
      Generalized Multi-Protocol Label Switching (GMPLS)";
  }

  leaf setup-priority {
    type uint8 {

range "0..7";
}
default "7";
description
"TE LSP requested setup priority.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}
leaf hold-priority {
  type uint8 {
    range "0..7";
  }
default "7";
description
"TE LSP requested hold priority.";
reference
"RFC 3209: RSVP-TE: Extensions to RSVP for LSP Tunnels";
}
leaf signaling-type {
  type identityref {
    base path-signaling-type;
  }
default "te-types:path-setup-rsvp";
description
"TE tunnel path signaling type.";
}

grouping tunnel-constraints {
  description
  "Tunnel constraints grouping that can be set on
   a constraint set or directly on the tunnel.";
  uses te-topology-identifier;
  uses common-constraints;
}

grouping path-constraints-route-objects {
  description
  "List of route entries to be included or excluded when
   performing the path computation.";
  container explicit-route-objects-always {
    description
    "Container for the 'exclude route' object list.";
    list route-object-exclude-always {
      key "index";
      ordered-by user;
      description
      "List of route objects to always exclude from the path
computation.

leaf index {
  type uint32;
  description
  "Explicit Route Object index. The index is used to
  identify an entry in the list. The order of entries
  is defined by the user without relying on key values."
}
uses explicit-route-hop;

list route-object-include-exclude {
  key "index";
  ordered-by user;
  description
  "List of route objects to include or exclude in the path
  computation.";
  leaf explicit-route-usage {
    type identityref {
      base route-usage-type;
    }
    default "te-types:route-include-object";
    description
    "Indicates whether to include or exclude the
    route object. The default is to include it."
  }
  leaf index {
    type uint32;
    description
    "Route object include-exclude index. The index is used
    to identify an entry in the list. The order of entries
    is defined by the user without relying on key values."
  }
uses explicit-route-hop {
  augment "type" {
    case srlg {
      container srlg {
        description
        "SRLG container.";
        leaf srlg {
          type uint32;
          description
          "SRLG value.";
        }
      }
      description
      "An SRLG value to be included or excluded.";
    }
    description
    "An SRLG value to be included or excluded.";
  }
}
"Augmentation for a generic explicit route for SRLG exclusion.";
case srlg {
  container srlg {
    description
    "SRLG container.";
    leaf srlg {
      type uint32;
      description
      "SRLG value.";
    }
  }
}

description
"An SRLG value to be included or excluded.";

description
"Augmentation for a generic explicit route for SRLG exclusion.";
}
}
}

grouping generic-path-metric-bounds {
  description
  "TE path metric bounds grouping.";
  container path-metric-bounds {
    description
    "TE path metric bounds container.";
    list path-metric-bound {
      key "metric-type";
      description
      "List of TE path metric bounds.";
      leaf metric-type {
        type identityref {
          base path-metric-type;
        }
        description
        "Identifies an entry in the list of 'metric-type' items bound for the TE path.";
      }
      leaf upper-bound {
        type uint64;
        default "0";
        description
        "Upper bound on the end-to-end TE path metric. A zero indicates an unbounded upper limit for the specific 'metric-type'.";
      }
    }
  }
}
grouping generic-path-optimization {
  description "TE generic path optimization grouping.";
  container optimizations {
    description "The objective function container that includes attributes to impose when computing a TE path.";
    choice algorithm {
      description "Optimizations algorithm.";
      case metric {
        if-feature "path-optimization-metric";
        /* Optimize by metric */
        list optimization-metric {
          key "metric-type";
          description "TE path metric type.";
          uses optimization-metric-entry;
        }
        /* Tiebreakers */
        container tiebreakers {
          description "Container for the list of tiebreakers.";
          list tiebreaker {
            key "tiebreaker-type";
            description "The list of tiebreaker criteria to apply on an equally favored set of paths, in order to pick the best.";
            leaf tiebreaker-type {
              type identityref {
                base path-metric-type;
              }
              description "Identifies an entry in the list of tiebreakers.";
            }
          }
        }
      }
      case objective-function {
        if-feature "path-optimization-objective-function";
        /* Objective functions */
        container objective-function {
          description "The objective function container that includes
attributes to impose when computing a TE path.
leaf objective-function-type {
  type identityref {
    base objective-function-type;
  }
  default "te-types:of-minimize-cost-path";
  description   "Objective function entry."
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
grouping generic-path-affinities {
  description   "Path affinities grouping.
  container path-affinities-values {
    description   "Path affinities represented as values."
    list path-affinities-value {
      key "usage";
      description   "List of named affinity constraints.
      leaf usage {
        type identityref {
          base resource-affinities-type;
        }
        description   "Identifies an entry in the list of value affinity constraints."
      }
      leaf value {
        type admin-groups;
        default "";
        description   "The affinity value. The default is empty."
      }
    }
  }
  container path-affinity-names {
    description   "Path affinities represented as names."
    list path-affinity-name {
      key "usage";
      description   "List of named affinity constraints."
    }
  }
}
leaf usage {
  type identityref {
    base resource-affinities-type;
  }
  description
    "Identifies an entry in the list of named affinity constraints.";
}

list affinity-name {
  key "name";
  leaf name {
    type string;
    description
      "Identifies a named affinity entry.";
  }
  description
    "List of named affinities.";
}
}

grouping generic-path-srlgs {
  description
    "Path SRLG grouping.";
  container path-srlgs-lists {
    description
      "Path SRLG properties container.";
    list path-srlgs-list {
      key "usage";
      description
        "List of SRLG values to be included or excluded.";
      leaf usage {
        type identityref {
          base route-usage-type;
        }
        description
          "Identifies an entry in a list of SRLGs to either include or exclude.";
      }
      leaf-list values {
        type srlg;
        description
          "List of SRLG values.";
      }
    }
  }
  container path-srlgs-names {

description
"Container for the list of named SRLGs."
list path-srlgs-name {
  key "usage";
  description
  "List of named SRLGs to be included or excluded.";
  leaf usage {
    type identityref {
      base route-usage-type;
    }
    description
    "Identifies an entry in a list of named SRLGs to either
     include or exclude.";
  }
  leaf-list names {
    type string;
    description
    "List of named SRLGs.";
  }
}
}

grouping generic-path-disjointness {
  description
  "Path disjointness grouping.";
  leaf disjointness {
    type te-path-disjointness;
    description
    "The type of resource disjointness.
     When configured for a primary path, the disjointness level
     applies to all secondary LSPs. When configured for a
     secondary path, the disjointness level overrides the level
     configured for the primary path.";
  }
}

grouping common-path-constraints-attributes {
  description
  "Common path constraints configuration grouping.";
  uses common-constraints;
  uses generic-path-metric-bounds;
  uses generic-path-affinities;
  uses generic-path-srlgs;
}

grouping generic-path-constraints {
  description

"Global named path constraints configuration grouping.";
container path-constraints {
    description
    "TE named path constraints container.";
    uses common-path-constraints-attributes;
    uses generic-path-disjointness;
}

grouping generic-path-properties {
    description
    "TE generic path properties grouping.";
    container path-properties {
        config false;
        description
        "The TE path properties.";
        list path-metric {
            key "metric-type";
            description
            "TE path metric type.";
            leaf metric-type {
                type identityref {
                    base path-metric-type;
                }
                description
                "TE path metric type.";
            }
            leaf accumulative-value {
                type uint64;
                description
                "TE path metric accumulative value.";
            }
        }
        uses generic-path-affinities;
        uses generic-path-srlgs;
    }
    container path-route-objects {
        description
        "Container for the list of route objects either returned by
        the computation engine or actually used by an LSP.";
        list path-route-object {
            key "index";
            ordered-by user;
            description
            "List of route objects either returned by the computation
            engine or actually used by an LSP.";
            leaf index {
                type uint32;
                description
            }
        }
    }
}
"Route object entry index. The index is used to identify an entry in the list. The order of entries is defined by the user without relying on key values."
}
uses explicit-route-hop;
}
}

// NOTE: The grouping encoding-and-switching-type below has been added in this module revision
// RFC Editor: remove the note above and this note

grouping encoding-and-switching-type {
  description "Common grouping to define the LSP encoding and switching types";
  leaf encoding {
    type identityref {
      base te-types:lsp-encoding-types;
    }
    description "LSP encoding type.";
    reference "RFC3945";
  }
  leaf switching-type {
    type identityref {
      base te-types:switching-capabilities;
    }
    description "LSP switching type.";
    reference "RFC3945";
  }
}
</CODE ENDS>

Figure 1: TE Types YANG module

4.  IANA Considerations

   This document updates the ietf-te-types YANG module registered by [RFC8776].
Therefore this document does not require any IANA actions.

5. Security Considerations

The security considerations defined in section 7 of [RFC8776] applies to the revision of the ietf-te-types YANG module.

This document just adds new typedefs and groupings to the YANG modules defined in [RFC8776] and therefore it does not introduce additional considerations.

6. References

6.1. Normative References


6.2. Informative References

Acknowledgements

The authors would like to thank Robert Wilton, Lou Berger, Mahesh Jethanandani and Jeff Haas for their valuable input to the discussion about the process to follow to provide tiny updates to a YANG module already published as an RFC.

This document was prepared using kramdown.

Authors’ Addresses

Italo Busi
Huawei
Email: italo.busi@huawei.com

Aihua Guo
Futurewei Technologies
Email: aihuaguo.ietf@gmail.com

Xufeng Liu
IBM Corporation
Email: xufeng.liu.ietf@gmail.com

Tarek Saad
Juniper Networks
Email: tsaad@juniper.net

Rakesh Gandhi
Cisco Systems, Inc.
Email: rgandhi@cisco.com

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net
Igor Bryskin
Individual
Email: i_bryskin@yahoo.com
Connecting 3GPP slices through IETF Network Slice services
draft-contreras-teas-3gpp-ietf-slice-mapping-00

Abstract

3GPP is introducing the concept of slicing as a primary way of service delivery. Slicing at 3GPP implies the differentiation of services in terms of performance expectations as well as the connection of different network entities also potentially differentiated per slice. With that aim, 3GPP is defining a number of logical constructs with the intent of being served with specific characteristics, determined by different QoS profiles. This document describes the connectivity of 3GPP slices through IETF Network Slice services taking into account that specific service level objectives, and identifies gaps existing nowadays on both 3GPP and IETF specifications for an straightforward mapping of parameters between both environments.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 8, 2022.
1. Introduction

Editor’s Note: the terminology in this draft will be aligned with the final terminology defined in [I-D.ietf-teas-ietf-network-slices].

Network slicing intends to provide network capabilities tailored to specific service expectations. 3GPP has been a precursor of the slicing concept conceiving the 5G architecture that natively supports slicing.

3GPP network slices require then tailored connectivity services to interconnect 3GPP network entities with the expected behavior and footprint. For doing so, it is expected that 3GPP higher management system will require IETF Network Slice services to an IETF Network Slice Controller, as defined in [I-D.ietf-teas-ietf-network-slices].
The NBI model in [I-D.ietf-teas-ietf-network-slice-nbi-yang] is working on the definition of a technology agnostic model with the aim of permitting the flexible provision of IETF Network Slices. Being this a work in progress it is useful and convenient to exercise the mapping of the 3GPP constructs defined for interconnecting 3GPP slice parts with the IETF model for that purpose, then identifying gaps that could be reported back into the corresponding specification fora.

2. Conventions used in this document

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

3. Network slicing artifacts at 3GPP

The network slice concept was present in 3GPP specifications from the first 5G release (Rel-15). As captured in [TS23.501], a network slice represents a logical network providing specific network capabilities and network characteristics.

To make slicing a reality, every technical domain is split into one or more logical network partitions, each referred to as a network slice subnet. The definition of multiple slice subnets on a single domain allows this segment to provide differentiated behaviors, in terms of functionality and/or performance. The stitching of slice subnets across the RAN, CN and TN results in the definition of network slices.

From a management viewpoint, the concept of network slice subnet represents an independently manageable yet composable portion of a network slice. The rules for the definition of network slice subnets and their composition into network slices are detailed in the 5G Network Resource Model (NRM) [TS28.541], specifically in the Network Slice NRM fragment. This fragment captures the information model of 5G network slicing, which specifies the relationships between different slicing related managed entities, each represented as a separate Information Object Class (IOC). An IOC captures the semantics and attributes of a manageable entity; in other words, it defines the class based on which instances (objects) from this entity can be created. In the model, we have four different IOCs:

- NetworkSlice IOC, representing a network slice. This IOC is associated with one or more ServiceProfiles, each representing the requirements of a particular service. The 1:N relationship of NetworkSlice IOC with the ServiceProfile is because one network
slice can host multiple services, as long as they do not impose conflicting requirements.

- NetworkSliceSubnet IOC, associated with a network slice subnet. This IOC is associated with one or more SliceProfiles.

- ManagedFunction IOC, which represents a 5G network function.

- EP_Transport IOC, which represents an interface associated with transport network level information, e.g., transport address, reachability information, and QoS profiles.

For the transport related part of a network slice, the key focus is on EP_Transport IOC. Instances of this IOC serve to instantiate 3GPP interfaces which are needed to support Network Slicing and to define Network Slice transport resources within the 5G NRM. In a nutshell, the EP_Transport IOC permits to define additional logical interfaces for each slice instance of the 3GPP user plane.

According to [TS28.541], the EP_Transport construct on 3GPP side has the following attributes:

- ipAddress (mandatory): specifies the IP address assigned to the logical transport interface. It is used for transport routing. Assigned uniquely per slice. As per [TS28.541] IP address is defined as an IPv4 address or an IPv6 address. The concern is that for the coherent networking, IP address should be assigned to the interface with a network mask, to form an IPv4 or IPv6 prefix.

- logicInterfaceInfo (mandatory): a set of parameters, which includes logicInterfaceType and logicInterfaceId. It specifies the type and identifier of a logical interface. It could be a VLAN ID, MPLS Tag or Segment ID. This is assigned uniquely per slice.

- nextHopInfo (optional): identifies the ingress transport node. Each node can be identified by any combination of IP address of next-hop router of transport network, system name, port name and IP management addresses of transport nodes.

- qosProfile (optional): specifies the set of QoS parameters which are logically provisioned on both sides on a logical transport interface. This is assigned uniquely per slice.

- epApplicationRef (mandatory): specifies the list of application endpoints associated with the logical transport interface. May be assigned multiple per slice. Used to maintain association with corresponding 3GPP logical interface (NgU (N3), F1_U), to which
EP_Transport is related to. Notice that one EP_Transport (representing a logical transport interface) can be associated with more than one multiple EP_Application (representing an application endpoint of a 3GPP managed function), but also the other way around. While the first case captures the typical situation, the second case can be used for the sake of resilience or load balance in the transport network.

From the Transport Network domain side, these parameters define CE transport interface configuration and shall be taken as an input to the transport service model to create coherent Network Slice transport service.

According to the [TS28.541] attributes in the EP_Transport, the IETF Network Slice may be defined by the following combination of the parameters:

<table>
<thead>
<tr>
<th>EP_Transport attribute name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipAddress</td>
</tr>
<tr>
<td>logicInterfaceId</td>
</tr>
<tr>
<td>nextHopInfo</td>
</tr>
<tr>
<td>qosProfile</td>
</tr>
</tbody>
</table>

+---------------+----------------+----------------+----------------+
|   ipAddress   |logicInterfaceId|   nextHopInfo  | qosProfile     |
|---------------+----------------+----------------+----------------|
|   Different   |  Same for all   |   Different     |  Same for all  |
|   per slice   |    slices      |   per slice    |    slices      |
| Same for all   |           Different             |  Same for all  |
|   slices     |           per slice             |    slices      |
| Different     |  Same for all   |   Different     |  Same for all  |
|   per slice   |    slices      |   per slice    |    slices      |
| Same for all   |           Different             |  Same for all  |
|   slices     |           per slice             |    slices      |
| Different     |  Same for all   |   Different     |  Same for all  |
|   per slice   |    slices      |   per slice    |    slices      |

Figure 1: Table_name

From the perspective of IETF Network Slice realization, some of these options could be realized in a straightforward manner while other could require of advanced features (e.g., PBR, SRv6, FlexE, etc).
4. IETF network slice service

The IETF Network Slice (NS) service is defined in [I-D.ietf-teas-ietf-network-slices] as a set of connections between a number of CEs, with that connections having specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network, with the traffic of one customer being separated from another. The concept of IETF network slice is conceived as technology agnostic.

The IETF NS service is specified in terms of the set of endpoints (from CE perspective) connected to the slice, the type of connectivity among them, and a set of SLOs and SLEs for each connectivity construct.

In [I-D.ietf-teas-ietf-network-slice-nbi-yang] the endpoints are described by an identifier, with some metrics associated to the connections among them as well as certain policies (e.g., rate limits for incoming and outgoing traffic).

5. Mapping of 3GPP slice and IETF network slice endpoints

At the time of provisioning a 3GPP slice, it is required to provide slice connectivity constructs by means of IETF network slices. Then it is necessary to bind two different endpoints, as depicted in Figure 2:

- Mapping of EP_Transport (as defined by [TS28.541]) to the endpoint at the CE side of the IETF network slice. This is necessary because the IETF Network Slice Controller (NSC) will receive as input for the IETF network slice service the set of endpoints at CE side to be interconnected.

- Mapping of the endpoints at both CE and PE side. The endpoint at PE side should be elicited by some means by the NSC, in order to establish and set up the connectivity construct intended for the customer slice request, according to the SLOs and SLEs received from the higher level system.
5.1. Mapping EP_transport to IETF NS CE endpoints

The 3GPP Management system provides the EP_Transport IOC to extend the slice awareness to the transport network. The EP_Transport IOC contains parameters as IP address, additional identifiers (i.e., vlan tag, MPLS label, etc), and associated QoS profile. This IOC is related to the endpoints of the 3GPP managed functions (EP_Application IOC).

The information captured in the EP_Transport IOC (3GPP concern) should be translated into the CE related parameters (IETF concern). There will be cases where such translation is straightforward, as for instance, when the 3GPP managed functions run on monolithic, purpose-specific network elements, in the way that the IP address attribute from the EP_Transport IOC is the IP address of an interface of the network element. In this case, the information on EP_Transport IOC can be directly passed to the IETF NSC through the NBI, even though some additional information could be yet required, not being defined yet on 3GPP specifications (e.g., the mask applicable to the IP address field on EP_Transport).
However, there could be other cases where such a relationship is not straightforward. This could be the case of virtualized 3GPP managed functions that could be instantiated on a general-purpose network element. In these other cases it is necessary to define additional means for eliciting the endpoint at the CE side corresponding to the endpoint of the 3GPP-related function.

With solely EP_Transport characterization in 3GPP, we could expect the NS CE endpoint being identified by a combination of IP address and some additional information such as vlan tag or SRv6 label that could discriminate against a certain logical interface. The next hop router information is related to the next hop view from the perspective of the 3GPP entity part of the slice, then providing hints for determining the slice endpoint at the other side of the slice service. Finally, the QoS profile helps to determine configurations needed at the PE side to respect the SLOs in the connection between CEs slice endpoints.

5.2. Mapping IETF NS CE to PE endpoints

As described in [I-D.ietf-teas-ietf-network-slices], there are different potential endpoint positions for an IETF NS.

![Diagram of IETF Network Slice endpoints](image)

**Figure 3: IETF Network Slice endpoints**
The information that is passed to the IETF NSC in terms of endpoints is the information relative to the CE position, which is the one known by the slice customer. From that information, the NSC needs to infer the corresponding endpoint position at PE side, in order to setup the desired connectivity constructs with the SLOs indicated in the request.

Being slice request technology-agnostic, the identification of the slice endpoints at the PE side should leverage on generic information passed through the NBI to the IETF NSC.

6. Discussion on the realization of 3GPP slices through IETF Network Slice services

The way in which 3GPP is characterizing the slice endpoint (i.e., EP_Transport) is based on Layer 3 information (e.g., the IP Address). However the information provided seems not to be sufficient for instructing the IETF Network Slice Controller for the realization of the IETF NEtwork Slice. For instance, some basic information such as the mask associated to the IP address of the EP_Transport is not specified, as well as other kind of parameters like the connection MTU or the connectivity type (unicast, multicast, etc). More sophisticated information could be required as well, like the level of isolation or protection necessary for the intended slice.

In the case in which the 3GPP managed function runs on a purpose-specific network element, the IP address specified in the EP_Transport IOC serves as reference to identify the CE endpoint, assuming the endpoint of the CE has been configured with that IP address. With that information (together with the logical interface ID) should be sufficient for the IETF NSC to identify the counterpart endpoint at the PE side, and configuring it accordingly (e.g., with a compatible IP address) for setting up the slice end-to-end. Similarly, the next hop information in EP_Transport can help validate the end-to-end slice between PE endpoints.

In the case in which the 3GPP managed function is instantiated as a virtualized network function, the direct association between the IP address of EP_Transport and the actual endpoint mapped at the CE is not so clear. It could be the case, for instance when the virtualized network function is instantiated at the internal of a data center, that the CE facing the PE is far from the point where the function is deployed, being that connectivity extended through the internals of the data center (or by some internal configuration of a virtual switch in a server). In these situations additional information is needed for accomplishing the end-to-end connection.
At the same time, [TS28.541] IOC contains useful parameters to be used in IETF Network Slice creation mechanism and enriching IETF Network Slice model. The following parameters may be suggested as candidates to the correlation of the IETF Network Slice parameters and IETF Network Slice model enrichments:

- For the latency, dLThptPerSliceSubnet, uLThptPerSliceSubnet, reliability and delayTolerance attributes, the following NRM apply (with reference to the section in that specification):
  - CNSliceSubnetProfile (section 6.3.22 in [TS28.541])
  - RANSliceSubnetProfile (section 6.3.23 in [TS28.541])
  - TopSliceSubnetProfile (section 6.3.24 in [TS28.541])

- For the qosProfile attribute, the NRM which applies is EP_Transport (detailed in section 6.3.17 in [TS28.541])

7. Security Considerations

   To be done.

8. IANA Considerations

   This draft does not include any IANA considerations

9. Acknowledgements

   The work of Luis M. Contreras has been partially funded by the European Commission under Horizon 2020 project Int5Gent (grant agreement 957403).

10. References

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

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

[I-D.liu-teas-transport-network-slice-yang]


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/
Abstract

This document analyses the needs of potential customers of network slices realized with IETF techniques in several use cases, identifies the functionalities for the North Bound Interface (NBI) of an IETF Network Slice Controller to satisfy such requests.

Status of This Memo

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

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

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

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

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents.

Contreras, et al. Expires September 8, 2022
Table of Contents

1. Introduction .......................................................... 3
2. Conventions used in this document and terminology .......... 3
3. Northbound Interface for IETF Network Slices ............... 4
4. IETF Network Slice Use Cases ....................................... 5
   4.1. 5G Services ....................................................... 5
       4.1.1. 3GPP network slice .................................... 6
       4.1.1.1. Topology of the TN-NSS ............................ 6
       4.1.1.2. Traffic segregation and mapping to S-NSSAI list 7
       4.1.1.3. Reachability information ............................ 10
       4.1.1.4. QoS profiling ........................................ 10
       4.1.2. Private 5G networks .................................... 11
       4.1.2.1. Structure Patterns of Private 5G system ........ 11
       4.1.2.2. Use Cases Assumed in Private 5G ................. 11
       4.1.2.3. Attributes Required in Private 5G ............... 12
       4.1.3. Generic network Slice Template ....................... 12
       4.1.4. Categorization of GST attributes ..................... 13
       4.1.4.1. Attributes with direct impact on the IETF network slice definition .............................. 14
       4.1.4.2. Attributes with indirect impact on the IETF network slice definition ............................. 14
       4.1.4.3. Attributes with no impact on the IETF network slice definition ................................. 15
       4.1.5. Provisioning procedures ................................. 16
4.2. NFV-based services .............................................. 16
   4.2.1. Connectivity attributes .................................. 16
   4.2.2. Provisioning procedures ................................... 17
4.3. Network sharing .................................................. 18
   4.3.1. Connectivity attributes .................................. 18
   4.3.2. Provisioning procedures ................................... 19
4.4. SD-WAN ............................................................. 19
   4.4.1. SD-WAN Structure .......................................... 19
   4.4.2. Connectivity Attributes ................................... 21
   4.4.3. SD-WAN Endpoint Attributes ............................... 22
   4.4.4. SD-WAN UNI Attributes ................................... 23
4.5. Radio functional splits ......................................... 23
   4.5.1. Attributes and procedures ................................ 24
4.6. Additional use cases ............................................ 24
5. Summary of attributes and procedures ......................... 25
1. Introduction

A number of new technologies, such as 5G, NFV and SDN are not only evolving the network from a pure technological perspective but also are changing the concept in which new services are offered to the customers [I-D.homma-slice-provision-models] by introducing the concept of network slicing.

The transport network is an essential component in the end-to-end delivery of services and, consequently, it is necessary to understand what could be the way in which the transport network is consumed as a slice. For a definition of IETF network slice refer to [I-D.ietf-teas-ietf-network-slices].

In this document it is assumed that there exists a (logically) centralized component in the transport network, namely IETF Network Slice Controller (NSC) with the responsibilities on the control and management of the IETF network slices invoked for a given service, as requested by IETF network slice customers.

This document analyses different use cases deriving the needs of potential IETF network slice customers in order to identify the functionality required on the North Bound Interface (NBI) of the NSC to be exposed towards such IETF network slice customers. Solutions to construct the requested IETF network slices are out of scope of this document.

2. Conventions used in this document and terminology

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 [RFC2119].

The terminology in this draft will be aligned in forthcoming versions with the final terminology selected for describing the notion of IETF network slice when applied to IETF technologies, as defined in [I-D.ietf-teas-ietf-network-slices].
The term "transport network" in the context of this draft refers in broad sense to WAN, MBH, IP backbone and other network segments implemented by IETF technologies.

3. Northbound Interface for IETF Network Slices

In a general manner, the transport network supports different kinds of services. These services consume capabilities provided by the transport network for deploying end-to-end services, interconnecting network functions or applications spread across the network and providing connectivity toward the final users of these services.

Under the slicing approach, a IETF network slice customer requests to a IETF network slice controller a slice with certain characteristics and parametrization. Such request it is assumed here to be done through a NBI exposed by the NSC to the customer, as reflected in Figure 1.

```
+--------------------+
   |                    |
   |    IETF Network    |
   |   Slice Customer   |
   |                    |
   +--------------------+
       |                |
       |   A            |
       |                |
       |    IETF Network|
       |    Slice       |
       |    Controller  |
       |                |
       |                |
       |                |
       |              V|
       |                |
       |    IETF Network|
       |    Slice       |
       |    Controller  |
       |                |
       +--------------------+
```

Figure 1: IETF network slice NBI concept

The functionality supported by the NBI depends on the requirements that the slice customer has to satisfy. It is then important to understand the needs of the slice customers as well as the way of expressing them.
4. IETF Network Slice Use Cases

Different use cases for slice customers can be identified, as described in the following sections.

4.1. 5G Services

5G services natively rely on the concept of network slicing. 5G is expected to allow vertical customers to request slices in such a manner that the allocated resources and capabilities in the network appear as dedicated for them.

In network slicing scenarios, a vertical customer requests a network operator to allocate a network slice instance (NSI) satisfying a particular set of service requirements. The content/format of these requirements are highly dependent on the networking expertise and use cases of the customer under consideration. To deal with this heterogeneity, it is fundamental for the network operator to define a unified ability to interpret service requirements from different vertical customers, and to represent them in a common language, with the purposes of facilitating their translation/mapping into specific slicing-aware network configuration actions. In this regard, model-based network slice descriptors built on the principles of reproducibility, reusability and customizability can be defined for this end.

As a starting point for such a definition, GSMA developed the idea of having a universal blueprint that, being offered by network operators, can be used by any vertical customer to order the deployment of an NSI based on a specific set of service requirements. The result of this work has been the definition of a baseline network slice descriptor called Generic network Slice Template (GST). The GST contains multiple attributes that can be used to characterize a network slice. A Network Slice Type (NEST) describes the characteristics of a network slice by means of filling GST attributes with values based on specific service requirements. Basically, a NEST is a filled-in version of a GST. Different NESTs allow describing different types of network slices. For slices based on standardized service types, e.g. eMBB, uRLLC and mIoT, the network operator may have a set of readymade, standardized NESTs (S-NESTs). For slices based on specific industry use cases, the network operator can define additional NESTs.

Service requirements from a given vertical customer are mapped to a NEST, which provides a self-contained description of the network slice to be provisioned for that vertical customer. According to this reasoning, the NEST can be used by the network operator as input to the NSI preparation phase, which is defined in [TS28.530]. 3GPP is
working on the translation of the GST/NEST attributes into NSI related requirements, which are defined in the "ServiceProfile" data type from the Network Slice Information Object Class (IOC) in [TS28.541]. These requirements are used by the 3GPP Management System to allocate the NSI across all network domains, including transport network. The IETF network slice defines the part of that NSI that is deployed across the transport network.

Despite the translation is an on-going work in 3GPP it seems convenient to start looking at the GST attributes to understand what kind of parameters could be required for the IETF network slice NBI.

4.1.1. 3GPP network slice

A 3GPP network slice represents a logical network that provides specific capabilities and network characteristics, supporting the service requirements of one or more network slice customers. The service requirements of each network slice customer are captured into a separate "ServiceProfile" artifact within the network slice class (see Network Slicing NRM fragment in TS 28.541).

A 3GPP network slice spans from 5G NR access nodes to the UPF that terminates the PDU session, i.e. PSA UPF. In this in-slice data path, there are TN segments (e.g. backhaul) that are out of scope of 3GPP management domain. For the provisioning and operation of these TN segments, usually referred to as transport Network Slice Subnets (TN-NSS), the 3GPP management system relies on an external TN management system, which hosts (among other components) the IETF NSC. To proceed with this delegation, the 3GPP management system needs to make available to the TN management system the information described in the following sub-sections.

4.1.1.1. Topology of the TN-NSS

The TN management system needs to know the transport termination/end points to determine the transport resources, either physical or virtual nodes. 3GPP management system systems need to provide the transport endpoints of 3GPP managed functions that are part of the RAN-NSS (e.g., gNB-CU-UP, gNB-CU-CP) and CN-NSS (e.g., UPF, AMF), and if applicable further information such as the next-hop router IP address configured in a RAN-NSS or CN-NSS. The TN management system should be able to correlate this with the transport network topology and derive the site or border routers connecting to 3GPP managed functions.
4.1.1.2. Traffic segregation and mapping to S-NSSAI list

As network functions can be shared by many network slices, it will be necessary to segregate the traffic belonging to specific slices on transport interfaces.

One option for traffic segregation is to assign application endpoints to specific sets of S-NSSAI values. The transport network can map packets to connectivity services based on local remote or remote endpoints, provided that the allocation of S-NSSAI to endpoints is known and exposed, and provided that the application endpoints are visible on the transport layer. The application endpoints visible in a RAN-NSS and CN-NSS are already mapped to a specific set of S-NSSAI. Figure 2 illustrates an example of this solution, whereby a 3GPP network slice with S-NSSAI=1 is mapped to specific application endpoints (e.g., N3 tunnel endpoint 1) by the access network node. In this example, the TN management system decides to map application endpoints 1 and 2 to the same transport connectivity service A. This mapping is implemented by the site router connecting to the access network node. On the core network slice, a similar mapping is done by the border router. Demultiplexing the packet streams belonging to different transport interfaces is based on regular routing and reachability of endpoint IP addresses.
Despite the simplicity of the above-referred approach, notice that it is not a universal solution as the application endpoint addresses are not always visible to the TN, for example when they are encrypted by IPSec tunnels. In such a case, the application endpoints are not visible to the site router, and thus cannot be used for transport connectivity mappings. To deal with these situations, an alternative solution is to use the concept of logical transport interfaces. A logical transport interface is a virtual interface separate from...
application endpoints; it can be for example a specific IP address / VLAN combination that corresponds to an IPSec termination point, an identifier (e.g., MPLS label, segment ID) that the TN recognizes, or it can be just a logical interface defined on top of a physical transport interface. As long as the interface identity can be derived from packet headers, the TN nodes can perform the mapping to transport connectivity services. In this regard, it is useful to indicate to the TN which traffic types are carried over an interface (e.g., N3 user plane packets, N2 control plane packets, etc.).
Figure 3 illustrates an example on the use of this solution. As seen, logical transport need to be exposed from 3GPP management system to TN management system, so that the latter can create transport network topology and determine the TN resources to support the 3GPP slice.

Logical transport interface, exposed by RAN NSSMF

Logical transport interface, exposed by CN NSSMF

Access network          Cell Site          Border Router          Core network node(s)          _Router (CSR)_          (BR)          node(s)

S-NSSAI=1: {NSS-AN 1, Logical Transport Interface 1, Transport Connectivity A, Logical Transport Interface 1, NSS-CN 1)
S-NSSAI=2: {NSS-AN 2, Logical Transport Interface 2, Transport Connectivity A, Logical Transport Interface 2, NSS-CN 2)
S-NSSAI=4: {NSS-AN 4, Logical Transport Interface 3, Transport Connectivity B, Logical Transport Interface 3, NSS-CN 4)

Figure 3: Logical Transport Interfaces

For traffic segregation, though solutions might be valid, 3GPP prefers the second solution: on the use of concept of transport logical interface. The reason is that it does not impose 1:1 mapping between application endpoint and transport interface (allowing for
better redundancy) and that it always works, no matter if encryption. To support this solution, the 3GPP has recently extended the Network Slice NRM fragment, including a new Information Object Class called
EP_Transport. This class provides a complete characterization of the logical transport interface, including transport level information (i.e., IP address, reachability information, QoS profile) and the set of application endpoints aggregated to this interface. For further information on reachability information and QoS profile, see next subsections. For further details on fields of EP_Transport, see Network Slice NRM fragment in TS 28.541.

4.1.1.3. Reachability information

Each physical or logical transport interface will carry the traffic associated with some 3GPP application endpoints that may be using IP addresses separate from the transport interface. These IP addresses must be reachable within the TN-NSS, and hence they need to be advertised to populate forwarding tables. A 3GPP network function can advertise such reachability information by running a dynamic routing protocol towards the next hop router. If that is not possible, it can create association between the reachability data with the logical transport interface and expose it towards the 3GPP and TN management system. This information can be derived from the IP addresses available for application and transport endpoints.

4.1.1.4. QoS profiling

Each TN-NSS may be associated a "TNSliceSubnetProfile", which hosts the SLO requirements (e.g., guaranteed throughput, bounded latency, maximum jitter) that the TN-NSS must support. "TNSliceSubnetProfile" is a 3GPP artifact that result from the decomposition of e2e service requirements ("ServiceProfile" artifact ) into domain-specific service requirements ("RANSliceSubnetProfile", "CNSliceSubnetProfile" and "TNSliceSubnetProfile") applicable to RAN-NSS, CN-NSS and TN-NSS respectively. Unlike "RANSliceSubnetProfile" and "CNSliceSubnetProfile", there is not agreement yet on the specific parameters to be captured by the "TNSliceSubnetProfile". Further work in this regard in the upcoming 3GPP SA5 meetings.

Upon receiving the "TNSliceSubnetProfile" from the 3GPP management system, the TN management system translates the SLO requirements therein into a QoS profile, which includes applicability and use of DSCPs and other QoS related properties onto the TN-NSS realization. To enable this, each logical interface may have an associated QoS profile. The QoS profile is just a reference to the detailed profile parameters which are logically provisioned on both sides of a logical transport interface.
4.1.2. Private 5G networks

Private 5G is one of variations of 5G service provision. Private 5G allows unlicensed as well as licensed companies to establish and operate 5G networks, with frequency band assigned for private 5G, in their own companies.

Private 5G can be customized flexibly rather than public 5G, and thus it enables us to provide networks specialized for their use cases. Private 5G is also called non-public 5G, and its deployment scenarios and service attributes are described in (ref. [TS23.501]).

4.1.2.1. Structure Patterns of Private 5G system

In Private 5G, a Service Provider does not necessarily have its own resources (e.g., radio bases, transit network and server resources for 5G CP functions) and can flexibly customize and deploy by selecting and combining various resources.

Private 5G has several structure patterns:

- Pattern 1: a service provider has all resources including radio bases, transit networks, and server resources for 5G CP functions.
- Pattern 2: a service provider has radio bases and server resources for 5G CP functions, and lends transit networks from other network operators.
- Pattern 3: a service provider has only radio bases and lends transit networks and server resources for 5G CP functions from other network operators and data center companies.

In pattern 2 and 3, it is assumed that a service provider uses network slices provided by other companies.

4.1.2.2. Use Cases Assumed in Private 5G

Private 5G provides a wireless communication environment which has specific features depending on applications or usage, within limited areas. From such aspects, within 5G use cases (ref. [TS22.261]), the following communication types and use cases could be especially expected to be provided with private 5G.

- High-bandwidth and reliable communication:
  - VR streaming
- Low latency and jitter:
* Smart factory
* Remote automated robot operation (e.g., robot concierge/assistant, robot waiter, drone)
  o High-bandwidth on up-link and low latency and jitter
* Remote surgery
* Uploading of high-definition video

4.1.2.3. Attributes Required in Private 5G

Private 5G has some distinguished requirements to network slice as below.

o QoS customization:
  * assured bandwidth
  * assured latency and jitter
  * customization of UL/DL rate on throughput (e.g., for video upstreaming consumes much UL bandwidth)

o Multi-homing (for high reliability, preparing multiple paths traverse different physical routes)

o Performance monitoring (e.g., for connectivity status and service availability of devices)

o Traffic flow separation/segregation (e.g., segregation of user plane and other communications physically and/or logically)

4.1.3. Generic network Slice Template

The structure of the GST is defined in [GSMA]. The template defines a total of 35 attributes. For each of them, the following information is provided:

o Attribute definition, which provides a formal definition of what the attribute represents.

o Attribute parameters, including:
  * Value, e.g. integer, float.
  * Measurement unit, e.g. milliseconds, Gbps
* Example, which provides examples of values the parameter can take in different use cases.

* Tag, which allow describing the type of parameter, according to its semantics. An attribute can be tagged as a characterization attribute or a scalability attribute. If it is characterization attribute, it can be further tagged as a performance-related attribute, a functionality-related attribute or an operation-related attribute.

* Exposure, which allow describing how this attribute interact with the slice customer, either as an API or a KPI.

  o Attribute presence, either mandatory, conditional or optional.

Attributes from GST can be used by the network operator (slice controller) and a vertical customer (slice customer) to agree SLA.

GST attributes are generic in the sense that they can be used to characterize different types of network slices. Once those attributes become filled with specific values, it becomes a NEST which can be ordered by slice customers.

4.1.4. Categorization of GST attributes

Not all the GST attributes as defined in [GSMA] have impact in the transport network since some of them are specific to either the radio or the mobile core part.

In the analysis performed in this document, the attributes have been categorized as:

  o Directly impactive attributes, which are those that have direct impact on the definition of the IETF network slice, i.e., attributes that can be directly translated into requirements required to be satisfied by a IETF network slice.

  o Indirectly impactive attributes, which are those that impact in an indirect manner on the definition of the IETF network slice, i.e., attributes that indirectly impose some requirements to a IETF network slice.

  o Non-impactive attributes, that are those which do not have impact on the IETF network slice at all.

The following sections describe the attributes falling into the three categories.
4.1.4.1. Attributes with direct impact on the IETF network slice definition

The following attributes impose requirements in the IETF network slice

- 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)

4.1.4.2. Attributes with indirect impact on the IETF network slice definition

The following attributes indirectly impose requirements in the IETF network slice to support the end-to-end service.

- Area of service (i.e., the area where terminals can access a particular network slice)
- Delay tolerance (i.e., if the service can be delivered when the system has sufficient resources)
- Downlink (maximum) throughput per UE
- Network functions owned by Network Slice Customer
- Maximum number of (concurrent) PDU sessions
- Performance prediction (i.e., capability to predict the network and service status)
- Root cause investigation
- Session and Service Continuity support
- Simultaneous use of the network slice
- Supported device velocity
- UE density
- Uplink (maximum) throughput per UE
- User management openness (i.e., capability to manage users’ network services and corresponding requirements)
- Latency from (last) UPF to Application Server

4.1.4.3. Attributes with no impact on the IETF network slice definition

The following attributes do not impact the IETF network slice.

- Location based message delivery (not related to the geographical spread of the network slice itself but with the localized distribution of information)
- MMTel support, i.e. support of and Multimedia Telephony Service (MMTel) as well as IP Multimedia Subsystem (IMS) support.
- NB-IoT Support, i.e., support of NB-IoT in the RAN in the network slice.
- Maximum number of (simultaneous) UEs
- Positioning support
- Radio spectrum
- Synchronicity (among devices)
- V2X communication mode
- Network Slice Specific Authentication and Authorization (NSSAA)
4.1.5. Provisioning procedures

3GPP identifies in [TS28.541] a number of procedures for the provisioning of a network slice in general. It can be assumed that similar procedures may also apply to a transport slice, facilitating a consistent management and control of end-to-end slices.

The envisioned procedures are the following:

- **Slice instance allocation**: this procedure permits to create a new slice instance (or reuse an existing one).
- **Slice instance de-allocation**: this procedure decommissions a previously instantiated slice.
- **Slice instance modification**: this procedure permits the change in the characteristics of an existing slice instance.
- **Get slice instance status**: this procedure helps to retrieve run-time information on the status of a deployed slice instance.
- **Retrieval of slice capabilities**: this procedure assists on getting information about the capabilities (e.g. maximum latency supported).

All these procedures fit in the operation of transport network slices.

4.2. NFV-based services

NFV technology allows the flexible and dynamic instantiation of virtualized network functions (and their composition into network services) on top of a distributed, cloud-enabled compute infrastructure. This infrastructure can span across different points of presence in a carrier network. By leveraging on transport network slicing, connectivity services established across geographically remote points of presence can be enriched by providing additional QoS guarantees with respect present state-of-the-art mechanisms, as conventional L2/L3 VPNs.

4.2.1. Connectivity attributes

The connectivity services are expressed through a number of attributes as listed:

- **Incoming and outgoing bandwidth**: bandwidth required for the connectivity services (in Mbps).
o Qos metrics: set of metrics (e.g., cost, latency and delay variation) applicable to a specific connectivity service

o Directionality: indication if the traffic is unidirectional or bidirectional.

o MTU: value of the largest PDU to be transmitted in the connectivity service.

o Protection scheme: indication of the kind of protection to be performed (e.g., 1;1, 1+1, etc.)

o Connectivity mode: indication of the service is point-to-point of point-to-multipoint

All those attributes will assist on the characterization of the connectivity slice to be deployed, and thus, are relevant for the definition of a IETF network slice supporting such connectivity.

4.2.2. Provisioning procedures

ETSI NFV defines the role of WAN Infrastructure Manager (WIM) as the component in charge of managing and controlling the connectivity external to the PoPs. In [IFA032] a number of interfaces are identified to be exposed by the WIM for supporting the multi-site connectivity, thus representing the capabilities expected for a transport network slice, as well, in case of satisfying such connectivity needs by means of the slice concept.

The interfaces considered are the following:

o Multi-Site Connectivity Service (MSCS) Management: this interface permits the creation, termination, update and query of MSCSs, including reservation. It also enables subscription for notifications and information retrieval associated to the connectivity service.

o Capacity Management: this interface allows querying about the capacity (e.g. bandwidth), topology, and network edge points of the connectivity service, as well as about information of consumed and available capacity on the underlying network resources.

o Fault Management: this interface serves for the provision of alarms related to the MSCSs.

o Performance Management: this interface assists on the retrieval of performance information (measurement results collection and notifications) related to MSCSs.
4.3. Network sharing

Network sharing is one of the means network operators exploit for increasing efficiencies. There are different scenarios of network sharing, being especially popular in the deployment of mobile networks, typically referred to as Radio Access Network (RAN) sharing. From an operational perspective, in RAN sharing we have two roles: master operator, being the actor (e.g. infrastructure provider, network operator) to which the deployment and daily operation of shared RAN elements are entrusted to; and the participant operators, who are the mobile operators who share the RAN facilities provided by the master operator. Note that in this context the master and participant operator can be seen as provider and customer, respectively.

While there exist different modes of RAN sharing [TS23.251], including passive RAN sharing (infrastructure site sharing) and active RAN sharing (e.g. Multi-Operator Core Networks or MOCN), most of the cases require the establishment of separated connections in order to separate the traffic per participant operator. Such connections typically extend from the cell site to some pre-defined and agreed interconnection points, from which the traffic is routed and delivered to individual participant operators.

The above-referred connections can have specific attributes. Aspects like guaranteed bandwidth (in line with the expected load from the aggregated cells), redundancy, bounded latency (per kind of traffic), or secure delivery of the information should be considered.

The master operator is the one in charge of provisioning the connections and collecting management data (e.g. performance measurements, telemetry, fault alarms, trace data) for individual participant operators. The use of network slicing could make the network sharing approach more flexible by allowing the other operators control and manage the established connections [MEF].

The implications of the RAN sharing scenario here described can be extended to either fixed networks or even to mobile networks leveraging on radio functional split (i.e., including fronthaul and midhaul network segments).

4.3.1. Connectivity attributes

The connections for RAN sharing typically consider attributes like:

- Maximum and Guaranteed Bit Rate (MBR and GBR respectively).
- Bounded latency (e.g., for user plane, control plane, etc)
o Packet loss rate.
o IP addressing (consistent among the operators sharing the infrastructure).
o L2/L3 reachability.
o Recovery time (on the event of failures).
o Secure connection (e.g., encryption support).

4.3.2. Provisioning procedures

The expected provisioning procedures are:
o Connection provisioning between site and interconnection point. Those connections could evolve in time in terms of capacity depending on the capacity growth of each particular site.
o Collection of management data, including performance measurements, fault alarms and trace data.

4.4. SD-WAN

SD-WAN is a solution to provide a virtual overlay network for connecting between customer's sites, (virtual) private cloud, or public cloud/Internet. SD-WAN operates over one or more underlay networks, and enables to offer more differentiated service delivery capabilities. SD-WAN can be esteemed as a type of network slices or can be established over underlay networks provided as network slices. The definitions, specification, service attributes, and framework of SD-WAN is defined in Metro Ethernet Forum ([MEF-70]).

SD-WAN forwards traffic based on application flows, and the policies include rules and constraints on the forwarding of the application flows. In SD-WAN, it may be required from the customer to adjust the behaviors based on its needs in near real time. The service provider is required to monitor the performance of the service and modify the forwarding policies based on the real-time telemetry from the underlying network components.

4.4.1. SD-WAN Structure

SD-WAN has three logical constructs:
o SD-WAN virtual connection
o SD-WAN virtual connection endpoint
Several additional components may be visible to the customer. These include:

- Customer network
- Service provider network
- Underlay connectivity
- Tunnel virtual connection

The following figure shows the overview of SD-WAN structure. In this case, the customer sites are connected with underlay connectivity#1 and they are also connected to remote private cloud with underlay connectivity#2. An SD-WAN endpoint is usually located in each customer network site as a CPE or a customer edge, and it allocates application flow to appropriate underlay connectivity.

```
Legend

/. . : Underlay connectivity#1
==== : Underlay Connectivity#2
EP   : SD-WAN Endpoint
```

**Figure 4: Overview of SD-WAN Structure**

SD-WAN may be provided as a network slice, or it is realized on several network slices provided as underlay connectivities. In the former case, a network slice PE will be mapped to CE in SD-WAN. In
the later case, PEs of the provider of underlay connectivities will behave as network slice PEs.

4.4.2. Connectivity Attributes

SD-WAN defined in MEF-70 has several attributes on its connectivity as below:

- **SD-WAN Identifier**: the value is a string that is used by the customer and service provider to uniquely identify an SD-WAN connectivity.

- **Endpoint list**: the value is a list contains endpoint identifiers and their connected endpoints.

- **Service Uptime Objective**: the value is the proportion of time that the connectivity service is working during a given time period.

- **Reserved Prefixes**: the values are IP prefixes reserved by the service provider for use for SD-WAN within its own network or for distribution to the customer via DHCP or SLAAC.

- **List for Policies**: the value is a list of policies applied to application flows and application flow groups at endpoints. An SD-WAN policy list contains policy name and list of policy criteria. Support of the criteria listed below would be required:
  
  * Encryption: indicates whether or not the application flow requires encryption
  
  * Public-Private: indicates whether the application flow can traverse public or private underlay connectivity services (or both).
  
  * Internet-Breakout: indicates whether the application flow should be forwarded to an Internet destination.
  
  * Billing-Method: indicate the application flow can be sent over an underlay connectivity service that has usage-based or flat-rate billing.
  
  * Backup: indicates whether this application flow can use a TVC designated as aEUR&#157;backupaEUR&#157;.
  
  * Bandwidth: specifies a rate limit on the application flow.
List of Application Flow Groups: the value is a list of application flow groups that application flows can be members of. An application flow group list contains application flow group name and application flow group policy.

List of Application Flows: the value is a list of the application flows that are recognized by the SD-WAN. An application flow list contains application flow name, list of application flow criteria, and application flow group name. The criteria is listed below:

* Ethertype
* C-VLAN ID list
* IPv4 source address
* IPv4 destination address
* IPv4 source or destination address
* IPv4 protocol list
* IPv6 source address
* IPv6 destination address
* IPv6 source or destination address
* IPv6 next header list
* TCP/UDP source port list
* TCP/UDP destination port list
* Application identifier
* any

4.4.3. SD-WAN Endpoint Attributes

SD-WAN contains some endpoints as boundary nodes between underlay connections and customers sites. [MEF-70] defines some attributes for SD-WAN endpoints as below:

Endpoint Identifier: the value is for identification of SD-WAN endpoint for management purposes.
4.4.4. SD-WAN UNI Attributes

SD-WAN UNI is a reference point that represents the demarcation between the responsibility of the customer and the responsibility of the provider. Some attributes for UNI is defined in [MEF-70] as below:

- **SD-WAN UNI Identifier**: the value is for identification of the UNI for management purposes.
- **SD-WAN UNI L2 Interface**: the value describes the underlay L2 interface for the UNI.
- **SD-WAN UNI Maximum L2 Frame Size**: the value specifies the maximum length L2 frame that is accepted by the provider.
- **SD-WAN UNI IPv4 connection addressing**: the value describes IPv4 connection address mechanisms (e.g., Static or DHCP).
- **SD-WAN UNI IPv6 connection addressing**: the value describes IPv6 connection address mechanisms (e.g., DHCP, SLAAC, Static or Link-Local-only).

4.5. Radio functional splits

The disaggregation of the software stack in radio base stations allows the centralization of some of the radio processing functions. O-RAN is promoting the interoperability of implementations of radio functional splits, defining an architecture where three main entities can be considered: the Radio Unit (RU), with some basic processing, the Distributed Unit (DU) with the rest of real-time processing capabilities, and the Centralized Unit (CU) with the non-real-time processing of the software stack. The network segment between RU and DU is known as fronthaul (FH), while the segment between DU and CU is referred as midhaul (MH). Figure 5 shows this situation.
The fronthaul leverages on eCPRI protocol which can be transported directly on Ethernet frames or encapsulated in IP/UDP (for the user plane). The midhaul can be transported in a similar way as the backhaul.

With current specifications, individual service flows being carried by FH cannot be distinguished, so no possibility of differentiating connectivity slices at that point. Similar thing happens for MH. The only possible differentiation per flow can happen in downstream direction from CU to DU, but this basically can only help for policing traffic at that point (i.e., slice is yet the same).

Advanced scenarios such as RU sharing could allow traffic differentiation per mobile operator based on e.g. vlans, being each of those vlans mapped to a different slice.

4.5.1. Attributes and procedures

The attributes of IETF network slices for the conveniently supported the radio functional split are based on main characteristics of FH/MH: Latency, BW, and packet loss, as specified in [O-RAN]. Geographical location could have an impact due to latency restrictions for FH.

Regarding slice management procedures, it can be assumed a similar lifecycle as in 3GPP slices.

4.6. Additional use cases

This is a placeholder for describing additional use cases (e.g., data center interconnection, etc). To be completed.
5. Summary of attributes and procedures

After analysing the different use cases, a number of attributes and procedures can be identified to provide IETF Network Slice services. Following sections summarize the findings per SLO, SLE and procedures.

Editor Note: this summary is yet under review.

5.1. Summary of SLOs

The following SLOs can be considered common to the majority of use cases.

- Bandwidth (or throughput), as an indication of the amount of traffic allowed to the delivered. It can be expressed unidirectional or bidirectional.
- Latency, as an indication of the maximum delay expected in a connection.
- Jitter (or delay variation), as an indication of the maximum variation on the delay expected in a connection.
- Packet loss, as an indication of the bounded limit of packet losses allowed in a connection.
- To be completed

5.2. Summary of SLEs

To be completed.

5.3. Summary of procedures

The following procedures allow to cover the analysed use cases.

- IETF Network Slice provision, including allocation and de-allocation of the slice.
- IETF Network Slice modification (or update) of an existing allocated slice.
- Retrieval (or query) of IETF Network Slice status and capabilities of an existing allocated slice.
- IETF Network Slice reservation, allowing a late instantiation of the slice.
6. Security Considerations

This draft does not include any security considerations.

7. IANA Considerations

This draft does not include any IANA considerations

8. References

8.1. Normative References


8.2. Informative References


[I-D.homma-slice-provision-models]

[I-D.ietf-teas-ietf-network-slice-definition]

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

"IFA032 Interface and Information Model Specification for Multi-Site Connectivity Services V3.2.1.", ETSI GS NFV-IFA 032 V3.2.1, April 2019.


"SD-WAN Service Attributes and Services", MEF-70, July 2019.

"O-RAN Xhaul Transport Requirements 1.0", O-RAN.WG9.XTRP-REQ-v01.00, November 2020.


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/
Shunsuke Homma
NTT
Japan
Email: shunsuke.homma.ietf@gmail.com

Jose A. Ordonez-Lucena
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid  28050
Spain
Email: joseantonio.ordonezlucena@telefonica.com

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

Hidetaka Nishihara
NTT
Email: hidetaka.nishihara1104@gmail.com
Considerations about Hierarchical IETF Network Slices
draft-dong-teas-hierarchical-ietf-network-slice-01

Abstract

Network slicing is targeted at existing or emerging customers or services which may request for network connectivity services with a specific set of Service Level Objectives (SLOs) and Service Level Expectations (SLEs). In some network scenarios, there can be requirements for the deployment of hierarchical network slices. The general framework of IETF network slice supports hierarchical network slicing, while the technologies for realizing hierarchical IETF network slice need to be considered.

This document describes the typical scenarios of hierarchical IETF network slices, and provides the considerations and requirements on the technologies in different network planes to realize hierarchical IETF network slices.

Status of This Memo

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

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

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

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

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

Network slicing is targeted at existing or emerging customers or services which may request for network connectivity services with a specific set of Service Level Objectives (SLOs) and Service Level Expectations (SLEs). The concept and general framework of IETF network slice are described in [I-D.ietf-teas-ietf-network-slices]. [I-D.ietf-teas-enhanced-vpn] describes the framework and technologies which can be used for IETF network slice realization by utilizing Virtual Private Network (VPN) and Traffic Engineering (TE) mechanisms with enhancements that specific services require over traditional VPNs.
[I-D.ietf-teas-ietf-network-slices] mentions that IETF Network Slices may be combined hierarchically, which means that a network slice may itself be further sliced. The technologies for realizing hierarchical IETF network slice need to be considered.

This document describes the typical scenarios in which the deployment of hierarchical IETF network slices may be needed. This document also provides the considerations and requirements on the technologies in different network planes to realize hierarchical IETF network slices.

2. Scenarios of Hierarchical IETF Network Slices

In this section, several possible network scenarios of hierarchical IETF network slicing are introduced.

2.1. Per-Customer Network Slices in an Industrial Network Slice

One of the typical network slice deployment is in the multi-industrial network case, in which a physical network is used to deliver services of multiple vertical industries. Separate IETF network slices are provided for different industries, such as healthcare, education, manufacturing, governmental affairs, etc. Then within the network slice of a specific industry, there may be need to create separate network slices for some or all of the customers within this industry.

For example, within the education network slice, some of the universities may require for a separate network slice to connect with a set of the branch campuses. Another examples is within the health-care network slice, some of the hospitals may require for a separate network slice for the connectivity and services between a set of the branch hospitals.
2.2. Per-application Network Slices in a Customer Network Slice

Another network slice deployment case is to provide dedicated IETF network slices for some important customers as the first-level network slices. While the customers may require to further split their network slices into different sub-network slices for different applications.

For example, a network slice for a hospital may be further divided to carry different type of medical services, such as remote patient monitoring, remote ultrasound diagnose, medical image transmission etc.
IETF network slice can also be delivered as a wholesale service to other network operators. In this case a network operator can be the customer of a network slice, and it may also need to deliver IETF network slice services to its customers. This is similar to the Carrier’s Carrier VPN service mode, while some additional requirements on the SLOs and SLEs may be required by the second-level network slice customer.
3. Considerations about Hierarchical Network Slice Realization

To support the realization of hierarchical network slices, there will be specific requirements on the technologies used in each network plane. In this section, the requirements of hierarchical network slicing on the forwarding plane network resource partitioning, the data plane encapsulations, the control plane protocols and the management plane are analyzed.

3.1. Forwarding Plane Network Resource Partitioning

For the realization of IETF network slices, the network resources in the underlying forwarding plane needs to be partitioned into different network resource partitions (NRPs), each NRP is used as the underlay network construct to support one or a group of IETF network slice services. In order to support hierarchical network slices, the forwarding plane network resources needs to be able to be partitioned in a hierarchical manner. Taking a two-level hierarchical network slice as an example, the bandwidth resource of a physical interface needs to be partitioned in two levels. There can be different options in modeling the interface resources of network resource partition.
The first option is to treat the network resources in the first-level NRPs as a set of layer-3 sub-interfaces, each with dedicated link bandwidth, and the second-level NRPs are represented as virtual data channels under the layer-3 sub-interfaces, as shown in the figure below:

![Diagram of network resource partition on interface: Option 1](image)

**Figure 4. Modeling of Network Resource Partition on Interface: Option 1**

The second option is to treat the first-level NRPs as layer-2 sub-interface of the layer-3 interface, and the second-level NRPs are represented as virtual data channels under the layer-2 sub-interface, as shown in the figure below:
The options of the network resource partition modeling may have different impact to the control plane in terms of the number of control protocol sessions to be maintained, and the amount and types of information to be distributed in the control plane. Depends on the network deployment requirements, different resource partition modeling options may be used.

3.2. Data Plane Identifiers

Traffic of IETF network slices can be steered into the corresponding underlay network construct based on one or multiple fields in the data packet, so that the corresponding NRPs are used for processing and forwarding the packet. On the edge nodes of an IETF network slice, traffic flows can be classified and mapped to IETF network slices using flexible matching rules based on operators’ local policy. While on the intermediate network nodes, a dedicated data plane NRP Identifier [I-D.dong-teas-nrp-scalability] can facilitate the identification of the NRP and the set of network resources allocated on the network nodes for packet processing.

Layer-3 Physical Interface

Figure 5. Modeling of Network Resource Partition on Interface, Option 2
For hierarchical IETF network slices, such data plane identifiers may need to be able to identify both the first-level NRP and the second-level NRP. There are several options in the design of the data plane NRP identifier for hierarchical network slices.

The first option is to use a unified data plane identifier for both the first-level NRP and the second-level NRP. In this case, the first-level NRPs and the second-level NRPs are identified using distinct identifier values.

```
+-----------------------------------------+
| Unified NRP ID for different levels     |
+-----------------------------------------+
```

Figure 6. Unified NRP ID

The second option is to use a hierarchical identifiers for the first-level NRP and the second-level NRP. In this case, the first part of the identifier is used to identify the first-level NRP, and the second part of the identifier is used to identify the second-level NRP. Depends on the data plane technologies used, the hierarchical NRP may be encapsulated in a continuous field in the packet, or may be positioned in separate fields.

```
+--------------------+--------------------+
| Level-1 NRP ID     | Level-2 NRP ID     |
+--------------------+--------------------+
```

Figure 7. Hierarchical NRP IDs

3.3. Control Plane

The control plane may be used for the distribution of the attributes and states of the hierarchical NRPs and the associated data plane identifiers among network nodes in the NRP and also to the network controller. With different NRP modeling, the information may be advertised as either layer-3 or layer-2 network information, which may have different scalability implications to the control plane. And as the number of hierarchical network slices increases, some control plane optimization mechanisms may be needed to adopt to the amount of information advertised.
3.4. Management Plane

For the management hierarchical network slices, the management system of network operator needs to provide life-cycle management to both the first-level network slices and the second-level network slices. It should allow to manage the first-level and second-level network slices separately, while the relationship between the first-level and second-level network slices also need to be maintained in the management system. The management system may need to support additional functions and procedures for the management of hierarchical network slices. Further analysis about the requirement on the management plane is for further study.

4. Security Considerations

TBD

5. IANA Considerations

This document makes no request of IANA.

6. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

7. Acknowledgments

The authors would like to thank Yawei Zhang for the review and discussion of this document.

8. References

8.1. Normative References

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

8.2. Informative References

[I-D.dong-teas-nrp-scalability]
Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N., Mishra, G., Qin, F., Saad, T., and V. P. Beeram,

[I-D.ietf-teas-enhanced-vpn]

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
Scalability Considerations for Network Resource Partition
draft-dong-teas-nrp-scalability-01

Abstract

The IETF Network Slice service aims to meet the connectivity demands of a network slice customer with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network. A Network Resource Partition (NRP) is a set of network resources that are allocated from the underlay network to carry a specific set of network traffic and meet the required SLOs and SLEs. One or multiple IETF Network Slice services can be mapped to one NRP.

As the demand for IETF Network Slice services increases, scalability would become an important factor for the large scale deployment of IETF Network Slices. Although the scalability of IETF Network Slices can be improved by mapping a group of IETF Network Slices to one NRP, there are concerns about the scalability of NRPs. This document describes the scalability considerations about NRPs in the network control plane and data plane, and some optimization mechanisms are proposed.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
1. Introduction .................................................. 3
   3.1. Control Plane Scalability ................................ 5
      3.1.1. Distributed Control Plane .......................... 5
      3.1.2. Centralized Control Plane .......................... 6
   3.2. Data Plane Scalability ................................ 6
   3.3. Gap Analysis of the Existing Mechanism ..................... 7
4. Proposed Scalability Optimizations ............................ 8
   4.1. Control Plane Optimizations ................................ 8
      4.1.1. Distributed Control Plane Optimizations .......... 8
      4.1.2. Centralized Control Plane Optimization ........... 10
   4.2. Data Plane Optimizations ................................ 10
5. Solution Evolution for Improved Scalability ................... 12
6. Security Considerations ....................................... 13
7. IANA Considerations .............................................. 13
8. Contributors .................................................... 13
9. Acknowledgments .................................................. 13
10. References ...................................................... 13
10.1. Normative References ...................................... 13
1. Introduction

The IETF Network Slice service aims to meet the connectivity demands of a network slice customer with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network. [I-D.ietf-teas-ietf-network-slices] defines the terminologies and the characteristics of IETF Network Slices. It also discusses the general framework, the components and interfaces for requesting and operating IETF Network Slices. For the realization of IETF Network Slice services, a concept called Network Resource Partition (NRP) is introduced, which refers to a set of network resources that are available in the underlay network to ensure the requested SLOs and SLEs of IETF Network Slices can be met.

[I-D.ietf-teas-enhanced-vpn] describes the layered framework and candidate technologies for delivering enhanced VPN (VPN+) services. VPN+ aims to meet the needs of customers or applications, including the applications that are associated with 5G, which require connectivity services with advanced characteristics, such as the assurance of Service Level Objectives (SLOs) and specific Service Level Expectations (SLEs). VPN+ services can be delivered by mapping one or a group of overlay VPNs to a virtual underlay network built with a set of network resources. The VPN+ framework and technologies could be used for the realization of IETF Network Slice services. NRP could be used as the underlay network construct to support the VPN+ services.

As the demand for IETF Network Slice services increases, scalability would become an important factor for the large scale deployment of IETF Network Slices. Although the scalability of IETF Network Slices can be improved by mapping a group of IETF Network Slices to one NRP, there are concerns about the scalability of NRPs. This document describes the scalability considerations about NRPs in the network control plane and data plane, and some optimization mechanisms are proposed.

2. Network Resource Partition Scalability Requirements

As described in [I-D.ietf-teas-ietf-network-slices], IETF Network Slices may be grouped together according to characteristics (including SLOs and SLEs). This grouping allows an operator to host a number of slices on a particular set of resources to reduce the amount of state information needed in the network. This can help to avoid the maintenance of per IETF Network Slice state in the underlay network.
With the development and evolution of 5G and other services, it is expected that an increasing number of IETF Network Slices will be deployed. The number of network slices required depends on how IETF Network Slices will be used, and the progress of network slicing for the vertical industrial services. The potential number of IETF Network Slice services and NRPs is analyzed by classifying the network slice deployment into three typical scenarios:

1. IETF Network Slices can be used by a network operator for different types of service. For example, in a converged multi-service network, different IETF Network Slices can be created to carry mobile transport services, fixed broadband services and enterprise services respectively: each type of service could be managed by a separate department or management team. Some service types, such as multicast services may also be deployed in a dedicated NRP. In this case, a separate NRP may need to be created for each service type. It is also possible that a network infrastructure operator provides IETF Network Slices to other network operators as a wholesale service, and a NRP may also be needed for each wholesale service customer. In this scenario, the number of NRPs in a network could be relatively small, such as in the order of 10 or so. This could be one of the typical cases in the beginning of IETF Network Slice deployment.

2. IETF Network Slices can be requested by customers of industrial verticals, where the assurance of SLOs and the fulfilment of SLEs are quite important. At the early stage of the vertical industrial services, a few top customers in some industries will begin to use IETF Network Slices to provide performance assurance to their business, such as smart grid, manufacturing, public safety, on-line gaming, etc. The realization of such IETF Network Slices typically requires the provision of different NRPs for different industries, and some top customers can require dedicated NRPs for strict service performance guarantees. Considering the number of vertical industries, and the number of top customers in each industry, the number of NRPs needed may be in the order of 100.

3. With the evolution of 5G and cloud networks, IETF Network Slices could be widely used by various vertical industrial customers and enterprise customers who require guaranteed or predictable service performance. The total amount of IETF Network Slices may increase to thousands or more, although it is expected that the number of IETF Network Slices would still be less than the number of traditional VPN services in the network. Accordingly, the number of NRPs needed may be in the order of 1000.
In [TS23501], the 3GPP defines a 32-bit identifier for a 5G network slice with an 8-bit Slice/Service Type (SST) and a 24-bit Slice Differentiator (SD). This allows mobile networks (the RAN and mobile core networks) to potentially support a large number of 5G network slices. It is likely that multiple 5G network slices are mapped to the same IETF Network Slice, but in some cases (for example, for specific SST or SD) the mapping may be closer to one-to-one, and the required NRPs may increase as well.

Thus, there may be large numbers of IETF Network Slices in some scenarios and the realization of IETF Network Slices needs to meet the scalability requirements. Mapping multiple IETF Network Slices to the same NRP presents a significant scaling benefit, but there can still be a requirement for a large number of NRPs which presents its own scalability challenges.

3. Network Resource Partition Scalability Considerations

This section analyses the scalability of NRPs in the control plane and data plane to understand the possible gaps in meeting the scalability requirements of IETF Network Slices.

3.1. Control Plane Scalability

The control plane for establishing and managing NRPs could be based on the hybrid of a centralized controller and a distributed control plane. The subsections that follow consider the scalability of these two approaches: the resultant scalability of the control plane will depend on how the hybrid is constructed.

3.1.1. Distributed Control Plane

It is necessary to create multiple NRPs for the delivery of IETF network slice services. Each NRP can be associated with a customized logical topology. The network resource attributes and the associated logical topology information of each NRP may need to be exchanged among the network nodes. The scalability of the distributed control plane used for the distribution of NRP information needs to be considered in the following aspects:

* The number of control protocol instances maintained on each node
* The number of protocol sessions maintained on each link
* The number of routes advertised by each node
* The amount of attributes associated with each route
* The number of route computations (i.e., SPF computation) executed by each node

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

3.1.2. Centralized Control Plane

By introducing a centralized network controller, the Software Defined Network (SDN) approach may help to reduce the amount of computation overhead in the distributed control plane, while it may also 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 NRPs, the controller needs to keep the topology and resource information of all the NRPs 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 NRPs require 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 and the TE paths.

3.2. Data Plane Scalability

To provide different IETF Network Slice services with the required SLOs and SLEs, it is necessary to allocate different subsets of network resources as different NRPs to avoid or reduce the unexpected interference from other services in the network. As the number of NRPs increases, it is required that the underlying network can provide fine-granular network resource partitioning, which means the amount of state about the partitioned network resources to be maintained on the network nodes will also increase.
In packet forwarding, IETF Network Slice service traffic needs to be processed according to the topology and resource attributes of the NRP it is mapped to, this means that some fields in the data packet need to be used to identify the NRP and its associated topology either directly or implicitly. Different approaches of encapsulating the NRP information in data packet can have different scalability implications.

One practical approach is to reuse some of the existing fields in the data packet to additionally identify the NRP the packet belongs to. For example, the destination IP addresses or the MPLS forwarding labels may be reused to further identify the NRP. This can avoid the cost of introducing new fields in the data packet, while since it introduces additional semantics to the existing fields, the processing of the existing fields in packet forwarding may need to be changed. Moreover, introducing resource semantics to existing identifiers in the packet (e.g., IP addresses, MPLS forwarding labels, etc.) may result in the increase of the amount of the existing IDs in proportion to the number of the NRPs, which may cause scalability problem in networks where a relatively large number of NRPs is needed.

An alternative approach is to introduce a new dedicated field in the data packet for identifying the NRP. This could avoid the impacts to the existing fields in the packet. And if this new field carries a globally-significant identifier of the NRP, it could be used together with the existing fields to determine the packet forwarding behavior. The potential issue with this approach is the difficulty in introducing a new field in some of the data plane technologies.

In addition, the introduction of NRP specific packet forwarding has an impact on the scalability of the forwarding entries on network nodes, as a network node may need to maintain separate forwarding entries for each NRP it participates in.

3.3. Gap Analysis of the Existing Mechanism

This section provides a gap analysis for an existing mechanism to perform NRP identification in the data plane and the related information distribution in the control plane.

One existing mechanism of building NRPs is to use resource-aware Segment Identifiers (either SR-MPLS or SRv6) [I-D.ietf-spring-resource-aware-segments] to identify the allocated network resources in the data plane based on the mechanisms described in [I-D.ietf-spring-sr-for-enhanced-vpn], and distribute the resource attributes and the associated logical topology information in the control plane using mechanisms based on Multi-topology.
[I-D.ietf-lsr-isis-sr-vtn-mt] or Flex-Algo
[I-D.zhu-lsr-isis-sr-vtn-flexalgo]. This mechanism is suitable for networks where a small number of NRPs are needed. As the number of NRPs increases, there may be several scalability challenges with this approach:

1. The number of SR SIDs needed will increase in proportion to the number of NRPs in the network, which will bring challenges both to the distribution of SR SIDs and the related information in the control plane, and to the installation of forwarding entries for resource-aware SIDs in the data plane.

2. As each NRP is associated with an independent logical topology or algorithm, the number of route computations (e.g., SPF computations) will increase in proportion to the number of NRPs in the network, which may introduce significant overhead to the control plane of network nodes.

3. The maximum number of logical topologies supported by OSPF [RFC4915] is 128, and the maximum number of Flexible Algorithms [I-D.ietf-lsr-flex-algo] is 128, which may not meet the required number of NRPs in some network scenarios.

4. Proposed Scalability Optimizations

4.1. Control Plane Optimizations

4.1.1. Distributed Control Plane Optimizations

Several optimizations can be considered to reduce the distributed control plane overhead and improve its scalability.

The first optimization mechanism is to reduce the amount of control plane sessions used for the establishment and maintenance of the NRPs. For multiple NRPs which have the same connection relationship between two adjacent network nodes, it is proposed that one single control protocol session is used for each such group of NRPs. The information of different NRPs can be exchanged over the same session, with necessary identification information to distinguish the NRPs in the control message. This could reduce the overhead of maintaining a large number of separate control protocol sessions for each NRP, and could also reduce the amount of control plane messages flooded in the network.
The second optimization mechanism is to decouple the NRP information from the associated logical topology information in the control plane, so that the resource attributes and the topology attributes can be advertised and processed separately. In a network, it is possible that multiple NRPs associate with the same logical topology, and multiple NRPs may share the same set of network resources on a subset of network nodes and links. Then it is more efficient if only one copy of the topology information is advertised, and multiple NRPs sharing the same topology could refer to this topology information. More importantly, with this approach, the result of topology-based route computation could be shared by multiple NRPs, so that the overhead of per NRP route computation could be avoided. Similarly, information of a subset of network resources reserved on a particular network node or link could be advertised once and may be referred to by multiple NRPs which share the same set of resources.

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

NRP-1                  NRP-2

O-----O-----O
|     |     |
O-----O-----O
```

Legend

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>Virtual node</td>
</tr>
<tr>
<td>###</td>
<td>Virtual links with a set of reserved resources</td>
</tr>
<tr>
<td>***</td>
<td>Virtual links with another set of reserved resources</td>
</tr>
</tbody>
</table>

Figure 1. Topology Sharing between NRPs

Figure 1 gives an example of two NRPs which share the same logical topology. As shown in the figure, NRP-1 and NRP-2 are associated with the same topology, while the resource attributes of each NRP 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 shared by the two NRPs to generate the corresponding routing and forwarding tables.
Figure 2 gives another example of two NRPs which share the same set of network resources on some of the links. In this case, information about the resources allocated on each link only needs to be advertised once, then both NRP-1 and NRP-2 could refer to the reserved link resource for constraint based path computation.

4.1.2. Centralized Control Plane Optimization

For the optimization of the centralized control plane, it is suggested that the centralized controller is used as a complementary mechanism to the distributed control plane rather than a replacement, so that the workload for NRP specific path computation in control plane could be shared by both the centralized controller and the network nodes, and the scalability of both systems could be improved. In addition, the centralized controller may be realized with multiple network entities, each is responsible for one subset or region of the network. This is the typical approach for scale out of the centralized control plane.

4.2. Data Plane Optimizations

To support more IETF Network Slice services while keeping the amount of data plane state at a reasonable scale, one typical approach is to classify a set of IETF Network Slice services which have similar service characteristics and performance requirements into a group, and such groups of IETF Network Slice services are mapped to one NRP, which is allocated with an aggregated set of network resources and the union of the required logical topologies to meet the service requirement of the whole group of IETF Network Slice services. Different groups of IETF Network Slice services can be mapped to different NRPs with different set of network resources allocated from the underlay network. With appropriate grouping of IETF Network Slice services, a reasonable number of NRPs with network resources
reservation and aggregation could still meet the IETF Network Slice service requirements.

Another optimization in the data plane is to decouple the identifiers used for topology-based forwarding and the identifier used for the resource-specific processing introduced by NRP. One possible mechanism is to introduce a dedicated network-wide NRP Identifier (NRP-ID) in the packet header to uniquely identify the set of network resources allocated to a NRP on each involved network node and link for the processing and forwarding of the data packets mapped to the NRP. Then the existing identifiers in the packet header used for topology based forwarding (e.g., the destination IP address, MPLS forwarding labels) are kept unchanged. The benefit is the amount of the existing topology-specific identifiers will not be impacted by the increasing number of NRPs. Since this new global NRP-ID field will be used together with other existing fields to determine the packet forwarding behavior, this may require network nodes to support a hierarchical forwarding table in data plane. Figure 3 shows the concept of using different data plane identifiers for topology-specific and resource-specific packet forwarding and processing respectively.

![Figure 3. Decoupled Topology and Resource Identifiers in data packet](image)

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or the extension headers to carry the NRP-ID 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 NRP-ID needs to be parsed by every node along the path which is capable of NRP aware forwarding.

[I-D.dong-6man-enhanced-vpn-vtn-id] introduces the mechanism of carrying the VTN resource ID (which is equivalent to NRP-ID in the context of network slicing) in IPv6 Hop-by-Hop extension header.
In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated NRP-ID either in the MPLS label stack or following the MPLS label stack. This way, the existing MPLS forwarding labels are used for topology-specific packet forwarding towards the destination node, and the NRP-ID is used to determine the set of network resources for packet processing. This requires that both the forwarding label and the NRP-ID be parsed by nodes along the forwarding path of the packet, and the forwarding behavior may depend on the position of the NRP-ID in the packet. The detailed extensions to MPLS data plane are under discussion as part of the work in MPLS Open Design Team and is 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 NRP need to evolve to support the increasing number of IETF Network Slice services and the increasing number of NRPs in the network.

At 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 mechanism to define the logical topology, it could provide a solution for building a limited number of NRPs in the network, and can meet the requirements of a relatively small number of IETF Network Slice services. This mechanism is called the basic SR based NRP.

As the required number of IETF Network Slice services increases, more NRPs may be needed, then the control plane scalability could be improved by decoupling the topology attribute from the resource attribute, so that multiple NRPs could share the same topology or resource attribute to reduce the control plane and data plane overhead. The data plane can still be based on the resource-aware SIDs. This mechanism is called the scalable SR based NRP. Both the basic and the scalable SR based NRP mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

When the data plane scalability becomes a concern, a dedicated NRP-ID can be introduced in the data packet to decouple the resource-specific identifiers from the topology-specific identifiers in the data plane, this could help to reduce the number of IP addresses or SR SIDs needed to support a large number of NRPs. This mechanism is called the NRP-ID based mechanism.
6. Security Considerations

This document describes the scalability considerations about the network control plane and data plane of NRPs in the realization of IETF Network Slice services, and proposes the mechanisms for scalability optimization. As the number of NRPs supported in the data plane and control plane of the network can be limited, this may be exploited as an attack vector by requesting a large number of network slices, which then result in the creation of a large number of NRPs.

One protection against this is to improve the scalability of the system to support more NRPs. Another possible solution is to make the network slice controller aware of the scaling constraints of the system and dampen the arrival rate of new network slices and NRPs request, and raise alarms when the thresholds are crossed.

The security considerations in [I-D.ietf-teas-ietf-network-slices] and [I-D.ietf-teas-enhanced-vpn] also apply to this document.

7. IANA Considerations

This document makes no request of IANA.

8. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

Hongjie Yang
Email: hongjie.yang@huawei.com

9. Acknowledgments

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

10. References

10.1. Normative References

[I-D.ietf-teas-enhanced-vpn]
10.2. Informative References

[I-D.dong-6man-enhanced-vpn-vtn-id]

[I-D.ietf-lsr-flex-algo]

[I-D.ietf-lsr-isis-sr-vtn-mt]
[I-D.ietf-spring-resource-aware-segments]

[I-D.ietf-spring-sr-for-enhanced-vpn]

[I-D.zhu-lsr-isis-sr-vtn-flexalgo]


Authors' Addresses

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

Email: jie.dong@huawei.com
Fengwei Qin  
China Mobile  
No. 32 Xuanwumenxi Ave., Xicheng District  
Beijing  
China  

Email: qinfengwei@chinamobile.com

Tarek Saad  
Juniper Networks  

Email: tsaad@juniper.net

Vishnu Pavan Beeram  
Juniper Networks  

Email: vbeeram@juniper.net
Scalability Considerations for Network Resource Partition
draft-dong-teas-nrp-scalability-02

Abstract

The IETF Network Slice service aims to meet the connectivity demands of a network slice customer with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network. A Network Resource Partition (NRP) is a set of network resources that are allocated from the underlay network to carry a specific set of network traffic and meet the required SLOs and SLEs. One or multiple IETF Network Slice services can be mapped to one NRP.

As the demand for IETF Network Slice services increases, scalability would become an important factor for the large scale deployment of IETF Network Slices. Although the scalability of IETF Network Slices can be improved by mapping a group of IETF Network Slices to one NRP, there are concerns about the scalability of NRPs. This document describes the scalability considerations about NRPs in the network control plane and data plane, and some optimization mechanisms are proposed.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on 17 November 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction ........................................... 3
2. Network Resource Partition Scalability Requirements ... 3
3. Network Resource Partition Scalability Considerations ... 5
   3.1. Control Plane Scalability ......................... 5
       3.1.1. Distributed Control Plane .................... 5
       3.1.2. Centralized Control Plane .................... 6
   3.2. Data Plane Scalability ............................ 7
   3.3. Gap Analysis of the Existing Mechanism ............ 8
4. Proposed Scalability Optimizations ....................... 8
   4.1. Control Plane Optimizations ....................... 9
       4.1.1. Distributed Control Plane Optimizations ....... 9
       4.1.2. Centralized Control Plane Optimization ...... 11
   4.2. Data Plane Optimizations .......................... 12
5. Solution Evolution for Improved Scalability ............. 13
6. Operational Considerations ............................. 14
7. Security Considerations ................................ 14
8. IANA Considerations .................................... 14
9. Contributors ........................................... 14
10. Acknowledgments ....................................... 15
11. References ............................................ 15
1. Introduction

The IETF Network Slice service aims to meet the connectivity demands of a network slice customer with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network. [I-D.ietf-teas-ietf-network-slices] defines the terminologies and the characteristics of IETF Network Slices. It also discusses the general framework, the components and interfaces for requesting and operating IETF Network Slices. For the realization of IETF Network Slice services, a concept called Network Resource Partition (NRP) is introduced, which refers to a set of network resources that are available in the underlay network to ensure the requested SLOs and SLEs of IETF Network Slices can be met.

[I-D.ietf-teas-enhanced-vpn] describes the layered framework and candidate technologies for delivering enhanced VPN (VPN+) services. VPN+ aims to meet the needs of customers or applications, including the applications that are associated with 5G, which require connectivity services with advanced characteristics, such as the assurance of Service Level Objectives (SLOs) and specific Service Level Expectations (SLEs). VPN+ services can be delivered by mapping one or a group of overlay VPNs to a virtual underlay network built with a set of network resources. The VPN+ framework and technologies could be used for the realization of IETF Network Slice services. NRP could be used as the underlay network construct to support the VPN+ services.

As the demand for IETF Network Slice services increases, scalability would become an important factor for the large scale deployment of IETF Network Slices. Although the scalability of IETF Network Slices can be improved by mapping a group of IETF Network Slices to one NRP, there are concerns about the scalability of NRPs. This document describes the scalability considerations about NRPs in the network control plane and data plane, and some optimization mechanisms are proposed.

2. Network Resource Partition Scalability Requirements

As described in [I-D.ietf-teas-ietf-network-slices], IETF Network Slices may be grouped together according to their characteristics (including SLOs and SLEs) and mapped the same NRP. An operator may have customized policy on the grouping and mapping of IETF network slices. This allows an operator to host a large number of slices on a relatively small number of NRPs to reduce the amount of state
information needed in the network. This can help to avoid the maintenance of per IETF Network Slice state in the underlay network.

With the development and evolution of 5G and other services, it is expected that an increasing number of IETF Network Slices will be deployed. The number of network slices required depends on how IETF Network Slices will be used, and the progress of network slicing for the vertical industrial services. The potential number of IETF Network Slice services and NRPs is analyzed by classifying the network slice deployment into three typical scenarios:

1. IETF Network Slices can be used by a network operator for different types of service. For example, in a converged multi-service network, different IETF Network Slices can be created to carry mobile transport services, fixed broadband services and enterprise services respectively: each type of service could be managed by a separate department or management team. Some service types, such as multicast services may also be deployed in a dedicated NRP. In this case, a separate NRP may need to be created for each service type. It is also possible that a network infrastructure operator provides IETF Network Slices to other network operators as a wholesale service, and a NRP may also be needed for each wholesale service customer. In this scenario, the number of NRPs in a network could be relatively small, such as in the order of 10 or so. This could be one of the typical cases in the beginning of IETF Network Slice deployment.

2. IETF Network Slices can be requested by customers of industrial verticals, where the assurance of SLOs and the fulfilment of SLEs are quite important. At the early stage of the vertical industrial services, a few top customers in some industries will begin to use IETF Network Slices to provide performance assurance to their business, such as smart grid, manufacturing, public safety, on-line gaming, etc. The realization of such IETF Network Slices typically requires the provision of different NRPs for different industries, and some top customers can require dedicated NRPs for strict service performance guarantees. Considering the number of vertical industries, and the number of top customers in each industry, the number of NRPs needed may be in the order of 100.
3. With the evolution of 5G and cloud networks, IETF Network Slices could be widely used by various vertical industrial customers and enterprise customers who require guaranteed or predictable service performance. The total amount of IETF Network Slices may increase to thousands or more, although it is expected that the number of IETF Network Slices would still be less than the number of traditional VPN services in the network. Accordingly, the number of NRPs needed may be in the order of 1000.

In [TS23501], the 3GPP defines a 32-bit identifier for a 5G network slice with an 8-bit Slice/Service Type (SST) and a 24-bit Slice Differentiator (SD). This allows mobile networks (the RAN and mobile core networks) to potentially support a large number of 5G network slices. It is likely that multiple 5G network slices are mapped to the same IETF Network Slice, but in some cases (for example, for specific SST or SD) the mapping may be closer to one-to-one, and the required NRPs may increase as well.

Thus, there may be large numbers of IETF Network Slices in some scenarios and the realization of IETF Network Slices needs to meet the scalability requirements. Mapping multiple IETF Network Slices to the same NRP presents a significant scaling benefit, but there can still be a requirement for a large number of NRPs which presents its own scalability challenges.

3. Network Resource Partition Scalability Considerations

This section analyses the scalability of NRPs in the control plane and data plane to understand the possible gaps in meeting the scalability requirements of IETF Network Slices.

3.1. Control Plane Scalability

The control plane for establishing and managing NRPs could be based on the hybrid of a centralized controller and a distributed control plane. The subsections that follow consider the scalability of these two approaches: the resultant scalability of the control plane will depend on how the hybrid is constructed.

3.1.1. Distributed Control Plane

It is necessary to create multiple NRPs for the delivery of IETF network slice services. Each NRP can be associated with a customized logical topology. The network resource attributes and the associated logical topology information of each NRP may need to be exchanged among the network nodes. The scalability of the distributed control plane used for the distribution of NRP information needs to be considered in the following aspects:
* The number of control protocol instances maintained on each node
* The number of protocol sessions maintained on each link
* The number of routes advertised by each node
* The amount of attributes associated with each route
* The number of route computations (i.e., SPF computation) executed by each node

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

3.1.2. Centralized Control Plane

By introducing a centralized network controller, the Software Defined Network (SDN) approach may help to reduce the amount of computation overhead in the distributed control plane, while it may also 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 NRPs, the controller needs to keep the topology and resource information of all the NRPs up-to-date. And for some network events such as network failure, the resulting updates to the NRPs may need to be distributed to the controller in real time. To achieve this, depend on the mechanisms used, 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 NRPs require 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 and the TE paths.
3.2. Data Plane Scalability

To provide different IETF Network Slice services with the required SLOs and SLEs, it is necessary to allocate different subsets of network resources as different NRPs to avoid or reduce the unexpected interference from other services in the network. As the number of NRPs increases, it is required that the underlying network can provide fine-granular network resource partitioning, which means the amount of state about the partitioned network resources to be maintained on the network nodes will also increase.

In packet forwarding, IETF Network Slice service traffic needs to be processed according to the topology and resource attributes of the NRP it mapped to, this means that some fields in the data packet need to be used to identify the NRP and its associated topology either directly or implicitly. Different approaches of encapsulating the NRP information in data packet can have different scalability implications.

One practical approach is to reuse some of the existing fields in the data packet to additionally identify the NRP the packet belongs to. For example, the destination IP addresses or the MPLS forwarding labels may be reused to further identify the NRP. This can avoid the cost of introducing new fields in the data packet, while since it introduces additional semantics to the existing fields, the processing of the existing fields in packet forwarding may need to be changed. Moreover, introducing resource semantics to existing identifiers in the packet (e.g., IP addresses, MPLS forwarding labels, etc.) may result in the increase of the amount of the existing IDs in proportion to the number of the NRPs, which may cause scalability problem in networks where a relatively large number of NRPs is needed.

An alternative approach is to introduce a new dedicated field in the data packet for identifying the NRP. This could avoid the impacts to the existing fields in the packet. And if this new field carries a globally-significant NRP identifier, it could be used together with the existing fields to determine the packet forwarding behavior. The potential issue with this approach is the difficulty in introducing a new field in some of the data plane technologies.

In addition, the introduction of NRP specific packet forwarding has an impact on the scalability of the forwarding entries on network nodes, as a network node may need to maintain separate forwarding entries for each NRP it participates in.
3.3. Gap Analysis of the Existing Mechanism

This section provides a gap analysis for an existing mechanism to perform NRP identification in the data plane and the related information distribution in the control plane.

One existing mechanism of building NRPs is to use resource-aware Segment Identifiers (either SR-MPLS or SRv6) [I-D.ietf-spring-resource-aware-segments] to identify the allocated network resources in the data plane based on the mechanisms described in [I-D.ietf-spring-sr-for-enhanced-vpn], and distribute the resource attributes and the associated logical topology information in the control plane using mechanisms based on Multi-topology [I-D.ietf-1sr-isis-sr-vtn-mt] or Flex-Algo [I-D.zhu-1sr-isis-sr-vtn-flexalgo]. This mechanism is suitable for networks where a small number of NRPs are needed. As the number of NRPs increases, there may be several scalability challenges with this approach:

1. The number of SR SIDs needed will increase in proportion to the number of NRPs in the network, which will bring challenges both to the distribution of SR SIDs and the related information in the control plane, and to the installation of forwarding entries for resource-aware SIDs in the data plane.

2. As each NRP is associated with an independent logical topology or algorithm, the number of route computations (e.g., SPF computations) will increase in proportion to the number of NRPs in the network, which may introduce significant overhead to the control plane of network nodes.

3. The maximum number of logical topologies supported by OSPF [RFC4915] is 128, the maximum number of logical topologies supported by IS-IS [RFC5120] is 4096, and the maximum number of Flexible Algorithms [I-D.ietf-1sr-flex-algo] is 128, which may not meet the required number of NRPs in some network scenarios.

4. Proposed Scalability Optimizations

To support more IETF Network Slice services while keeping the amount of network state at a reasonable scale, one basic approach is to classify a set of IETF Network Slice services which have similar service characteristics and performance requirements into a group, and such group of IETF Network Slice services are mapped to one NRP, which is allocated with an aggregated set of network resources and the union of the required logical topologies to meet the service requirement of the whole group of IETF Network Slice services. Different groups of IETF Network Slice services can be mapped to
different NRPs, each is allocated with different set of network resources from the underlay network. With appropriate grouping of IETF Network Slice services, a reasonable number of NRPs with proper network resource allocation could still meet the IETF Network Slice service requirements.

4.1. Control Plane Optimizations

4.1.1. Distributed Control Plane Optimizations

Several optimizations can be considered to reduce the distributed control plane overhead and improve its scalability.

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

The second optimization mechanism is to decouple the NRP information from the associated logical topology information in the control plane, so that the resource attributes and the topology attributes can be advertised and processed separately. In a network, it is possible that multiple NRPs are associated with the same logical topology, or multiple NRPs may share the same set of network resources on a subset of network nodes and links. For the topology sharing case, it is more efficient if only one copy of the topology information is advertised, and multiple NRPs sharing the same topology could simply refer to this topology information. More importantly, with this approach, the result of topology-based route computation could be shared by multiple NRPs, so that the overhead of per NRP route computation could be avoided. Similarly, for the resource sharing case, information of a subset of network resources reserved on a particular network node or link could be advertised once and then be referred to by multiple NRPs which share the same set of resources.
Figure 2 gives an example of two NRPs which share the same logical topology. As shown in the figure, NRP-1 and NRP-2 are associated with the same topology, while the resource attributes of each NRP are different. In this case, the information of the shared network topology can be advertised using either MT or Flex-Algo, then the two NRPs can be associated with the same MT or Flex-Algo, and the topology-based route computation result can be shared by the two NRPs to generate the corresponding routing and forwarding entries.
Figure 2. Resource Sharing between NRPs

Figure 3 gives another example of two NRPs which have different logical topologies, while share the same set of network resources on a subset of the links. In this case, the information about the shared resources allocated on the links only needs to be advertised once, then both NRP-1 and NRP-2 could refer to the common set of reserved link resource for constraint based path computation.

4.1.2. Centralized Control Plane Optimization

For the optimization of the centralized control plane, it is suggested that the centralized controller is used as a complementary mechanism to the distributed control plane rather than a replacement, so that the workload for NRP specific path computation in control plane could be shared by both the centralized controller and the network nodes, and the scalability of both systems could be improved. In addition, the centralized controller may be realized with multiple network entities, each is responsible for one subset or region of the network. This is the typical approach for scale out of the centralized control plane.
4.2. Data Plane Optimizations

One optimization in the data plane is to decouple the identifiers used for topology-based forwarding and the identifier used for the resource-specific processing introduced by NRP. One possible mechanism is to introduce a dedicated network wide NRP Identifier (NRP-ID) in the packet header to uniquely identify the set of local network resources allocated to a NRP on each involved network node and link for the processing and forwarding of the received packets. Then the existing identifiers in the packet header used for topology based forwarding (e.g., the destination IP address, MPLS forwarding labels) are kept unchanged. The benefit is the amount of the existing topology-specific identifiers will not be impacted by the increasing number of NRPs. Since this new NRP-ID field will be used together with other existing fields to determine the packet forwarding behavior, this may require network nodes to support a hierarchical forwarding table in data plane. Figure 4 shows the concept of using different data plane identifiers for topology-specific and resource-specific packet forwarding and processing respectively.

```
+--------------------------+
|       Packet Header      |
|                          |
| +----------------------+ |
| | Topology-specific IDs| |
| +----------------------+ |
|                          |
| +----------------------+ |
| |    Global NRP-ID     | |
| +----------------------+ |
+--------------------------+
```

Figure 3. Decoupled Topology and Resource Identifiers in data packet

In an IPv6 [RFC8200] based network, this could be achieved by introducing a dedicated field in either the IPv6 fixed header or the extension headers to carry the NRP-ID 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 NRP-ID needs to be parsed by every node along the path which is capable of NRP aware forwarding.

[I-D.ietf-6man-enhanced-vpn-vtn-id] introduces the mechanism of carrying the VTN resource ID (which is equivalent to NRP-ID in the context of network slicing) in IPv6 Hop-by-Hop extension header.
In an MPLS [RFC3032] based network, this may be achieved by introducing a dedicated NRP-ID either in the MPLS label stack or following the MPLS label stack. This way, the existing MPLS forwarding labels are used for topology-specific packet forwarding towards the destination node, and the NRP-ID is used to determine the set of network resources for packet processing. This requires that both the forwarding label and the NRP-ID be parsed by nodes along the forwarding path of the packet, and the forwarding behavior may depend on the position of the NRP-ID in the packet. The detailed extensions to MPLS data plane are under discussion as part of the work in MPLS Open Design Team and is 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 NRP need to evolve to support the increasing number of IETF Network Slice services and the increasing number of NRPs in the network. This section describes the possible solution evolution taking the SR based NRP solutions as an example, while the analysis and optimization in this document are generic and not specific to SR.

At the first step, by introducing resource-awareness to SR SIDs [I-D.ietf-spring-resource-aware-segments], and using Multi-Topology or Flex-Algo as the control plane mechanism to define the logical topology of the NRP, it could provide a solution for building a limited number of NRPs in the network, and can meet the requirements of a relatively small number of IETF Network Slice services. This mechanism is called the basic SR based NRP.

As the required number of IETF Network Slice services increases, more NRPs may be needed, then the control plane scalability could be improved by decoupling the topology attribute from the resource attribute, so that multiple NRPs could share the same topology or resource attribute to reduce the control plane and data plane overhead. The data plane can still be based on the resource-aware SIDs. This mechanism is called the scalable SR based NRP. Both the basic and the scalable SR based NRP mechanisms are described in [I-D.ietf-spring-sr-for-enhanced-vpn].

When the data plane scalability becomes a concern, a dedicated NRP-ID can be introduced in the data packet to decouple the resource-specific identifiers from the topology-specific identifiers in the data plane, this could help to reduce the number of IP addresses or SR SIDs needed to support a large number of NRPs. This mechanism is called the NRP-ID based mechanism.
6. Operational Considerations

The instantiation of NRP requires to perform NRP specific configurations on the involved network nodes and links. There can also be the cases in which the topology or the set of network resources allocated to a existing NRP needs to be modified. With the number of NRPs increases, the amount of configurations for NRP instantiation and modification will increase accordingly.

For the management and operation of NRPs and the optimization of paths within the NRPs, the status of NRPs needs to be monitored and reported to the network controller. The increasing number of NRPs would require additional NRP status information to be monitored and reported.

The configuration and operation of NRP could be achieved using mechanisms such as Netconf/YANG, the details are out of the scope of this document.

7. Security Considerations

This document describes the scalability considerations for the network control plane and data plane of NRPs in the realization of IETF Network Slice services, and proposes some mechanisms for scalability optimization. As the number of NRPs supported in the data plane and control plane of the network can be limited, this may be exploited as an attack vector by requesting a large number of network slices, which then result in the creation of a large number of NRPs.

One protection against this is to improve the scalability of the system to support more NRPs. Another possible solution is to make the network slice controller aware of the scaling constraints of the system and dampen the arrival rate of new network slices and NRPs request, and raise alarms when the thresholds are crossed.

The security considerations in [I-D.ietf-teas-ietf-network-slices] and [I-D.ietf-teas-enhanced-vpn] also apply to this document.

8. IANA Considerations

This document makes no request of IANA.

9. Contributors
10. Acknowledgments

The authors would like to thank Adrian Farrel, Dhruv Dhody, Donald Eastlake and Kenichi Ogaki for their review and valuable comments to this document.

11. References

11.1. Normative References

[I-D.ietf-teas-enhanced-vpn]

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


11.2. Informative References

[I-D.ietf-6man-enhanced-vpn-vtn-id]
Dong, J., Li, Z., Xie, C., Ma, C., and G. Mishra, "Carrying Virtual Transport Network (VTN) Identifier in IPv6 Extension Header", Work in Progress, Internet-Draft,

[I-D.ietf-lsr-flex-algo]

[I-D.ietf-lsr-isis-sr-vtn-mt]

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

[I-D.ietf-spring-sr-for-enhanced-vpn]

[I-D.zhu-lsr-isis-sr-vtn-flexalgo]


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

Liyan Gong
China Mobile
No. 32 Xuanwumenxi Ave., Xicheng District
Beijing
China
Email: gongliyan@chinamobile.com

Guangming Yang
China Telecom
No.109 West Zhongshan Ave., Tianhe District
Guangzhou
China
Email: yangguangm@chinatelecom.cn
Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Packet Optical Integration (POI)

draft-ietf-teas-actn-poi-applicability-06

Abstract

This document considers the applicability of Abstraction and Control of TE Networks (ACTN) architecture to Packet Optical Integration (POI) in the context of IP/MPLS and optical internetworking. It identifies the YANG data models being defined by the IETF to support this deployment architecture and specific scenarios relevant for Service Providers.

Existing IETF protocols and data models are identified for each multi-layer (packet over optical) scenario with a specific focus on the MPI (Multi-Domain Service Coordinator to Provisioning Network Controllers Interface) in the ACTN architecture.

Status of this Memo

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

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

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

1. Introduction...................................................3
   1.1. Terminology...............................................5
2. Reference network architecture..................................7
   2.1. Multi-domain Service Coordinator (MDSC) functions.......9
       2.1.1. Multi-domain L2/L3 VPN network services...........11
       2.1.2. Multi-domain and multi-layer path computation......14
   2.2. IP/MPLS Domain Controller and NE Functions...............17
   2.3. Optical Domain Controller and NE Functions...............19
3. Interface protocols and YANG data models for the MPIs.......19
   3.1. RESTCONF protocol at the MPIs............................19
   3.2. YANG data models at the MPIs............................20
       3.2.1. Common YANG data models at the MPIs...............20
       3.2.2. YANG models at the Optical MPIs...................21
       3.2.3. YANG data models at the Packet MPIs...............21
   3.3. PCEP.....................................................22
4. Inventory, service and network topology discovery.............23
   4.1. Optical topology discovery................................25
   4.2. Optical path discovery...................................26
   4.3. Packet topology discovery................................27
   4.4. SR-TE path discovery.....................................27
   4.5. Inter-domain link discovery..............................28
The complete automation of the management and control of Service Providers transport networks (IP/MPLS, optical, and microwave transport networks) is vital for meeting emerging demand for high-bandwidth use cases, including 5G and fiber connectivity services. The Abstraction and Control of TE Networks (ACTN) architecture and interfaces facilitate the automation and operation of complex optical and IP/MPLS networks through standard interfaces and data models. This allows a wide range of network services that can be requested by the upper layers fulfilling almost any kind of service level requirements from a network perspective (e.g. physical diversity, latency, bandwidth, topology, etc.)

Packet Optical Integration (POI) is an advanced use case of traffic engineering. In wide-area networks, a packet network based on the Internet Protocol (IP), and often Multiprotocol Label Switching (MPLS) or Segment Routing (SR), is typically realized on top of an
optical transport network that uses Dense Wavelength Division Multiplexing (DWDM) (and optionally an Optical Transport Network (OTN) layer).

In many existing network deployments, the packet and the optical networks are engineered and operated independently. As a result, there are technical differences between the technologies (e.g., routers compared to optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP, versus dealing with physical impairments in DWDM, or very different time scales). In addition, customers needs can be different between a packet and an optical network, and it is not uncommon to use different vendors in both domains. The operation of these complex packet and optical networks is often siloed, as these technology domains require specific skills sets.

The packet/optical network deployment and operation separation are inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by integrating the packet and the optical networks. Multi-layer online topology insight can speed up troubleshooting (e.g., alarm correlation) and network operation (e.g., coordination of maintenance events), multi-layer offline topology inventory can improve service quality (e.g., detection of diversity constraint violations) and multi-layer traffic engineering can use the available network capacity more efficiently (e.g., coordination of restoration). In addition, provisioning workflows can be simplified or automated as needed across layers (e.g., to achieve bandwidth-on-demand or to perform activities during maintenance windows).

ACTN framework enables this complete multi-layer and multi-vendor integration of packet and optical networks through Multi-Domain Service Coordinator (MDSC) and packet and optical Provisioning Network Controllers (PNCs).

In this document, critical scenarios for POI are described from the packet service layer perspective and identified the required coordination between packet and optical layers to improve POI deployment and operation. Precise definitions of scenarios can help with achieving a common understanding across different disciplines. The focus of the scenarios are multi-domain packet networks operated as a client of optical networks.

This document analyses the case where the packet networks support multi-domain SR-TE paths and the optical networks could be either a DWDM network or an OTN network (without DWDM layer) or multi-layer
OTN/DWDM network. DWDM networks could be either fixed-grid or flexible-grid.

Multi-layer and multi-domain scenarios, based on reference network described in section 2, and very relevant for Service Providers, are described in section 4 and in section 5.

For each scenario, existing IETF protocols and data models, identified in section 3.1 and section 3.2, are analysed with particular focus on the MPI in the ACTN architecture.

For each multi-layer scenario, the document analyzes how to use the interfaces and data models of the ACTN architecture.

A summary of the gaps identified in this analysis is provided in section 6.

Understanding the level of standardization and the possible gaps will help assess the feasibility of integration between packet and optical DWDM domains (and optionally OTN layer) in an end-to-end multi-vendor service provisioning perspective.

1.1. Terminology

This document uses the ACTN terminology defined in [RFC8453]

In addition this document uses the following terminology.

Customer service:

the end-to-end service from CE to CE

Network service:

the PE to PE configuration including both the network service layer (VRFs, RT import/export policies configuration) and the network transport layer (e.g. RSVP-TE LSPs). This includes the configuration (on the PE side) of the interface towards the CE (e.g. VLAN, IP adress, routing protocol etc.)

Port:

the physical entity that transmits and receives physical signals
Interface:

a physical or logical entity that transmits and receives traffic

Link:

an association between two interfaces that can exchange traffic directly

Ethernet link:

a link between two Ethernet interfaces

IP link:

a link between two IP interfaces

Cross-layer link:

an Ethernet link between an Ethernet interface on a router and an Ethernet interface on an optical NE

Intra-domain single-layer Ethernet link:

an Ethernet link between two Ethernet interfaces on physically adjacent routers that belong to the same P-PNC domain

Intra-domain single-layer IP link:

an IP link supported by an intra-domain single-layer Ethernet link

Inter-domain single-layer Ethernet link:

an Ethernet link between two Ethernet interfaces on physically adjacent routers which belong to different P-PNC domains

Inter-domain single-layer IP link:

an IP link supported by an inter-domain single-layer Ethernet link

Intra-domain multi-layer Ethernet link:

an Ethernet link supported by two cross-layer links and an optical tunnel in between
Intra-domain multi-layer IP link:

an IP link supported an intra-domain multi-layer Ethernet link

2. Reference network architecture

This document analyses several deployment scenarios for Packet and Optical Integration (POI) in which ACTN hierarchy is deployed to control a multi-layer and multi-domain network, with two optical domains and two packet domains, as shown in Figure 1:

---

The ACTN architecture, defined in [RFC8453], is used to control this multi-layer and multi-domain network where each Packet PNC (P-PNC) is responsible for controlling its packet domain and where each Optical PNC (O-PNC) in the above topology is responsible for controlling its
optical domain. The packet domains controlled by the P-PNCs can be Autonomous Systems (ASes), defined in [RFC1930], or IGP areas, within the same operator network.

The routers between the packet domains can be either AS Boundary Routers (ASBR) or Area Border Router (ABR): in this document, the generic term Border Router (BR) is used to represent either an ASBR or an ABR.

The MDSC is responsible for coordinating the whole multi-domain multi-layer (packet and optical) network. A specific standard interface (MPI) permits MDSC to interact with the different Provisioning Network Controller (O/P-PNCs).

The MPI interface presents an abstracted topology to MDSC hiding technology-specific aspects of the network and hiding topology details depending on the policy chosen regarding the level of abstraction supported. The level of abstraction can be obtained based on P-PNC and O-PNC configuration parameters (e.g., provide the potential connectivity between any PE and any BR in an SR-TE network).

In the reference network of Figure 1, it is assumed that:

- The domain boundaries between the packet and optical domains are congruent. In other words, one optical domain supports connectivity between routers in one and only one packet domain;
- There are no inter-domain physical links between optical domains. Inter-domain physical links exist only:
  - between packet domains (i.e., between BRs belonging to different packet domains): these links are called inter-domain Ethernet or IP links within this document;
  - between packet and optical domains (i.e., between routers and optical NEs): these links are called cross-layer links within this document;
  - between customer sites and the packet network (i.e., between CE devices and PE routers): these links are called access links within this document.
- All the physical interfaces at inter-domain links are Ethernet physical interfaces.
Although the new optical technologies (e.g., QSFP-DD ZR 400G) allows providing DWDM pluggable interfaces on the routers, the deployment of those pluggable optics is not yet widely adopted by the operators. The reason is that most operators are not yet ready to manage packet and optical networks in a single unified domain. The analysis of the unified use case is outside the scope of this draft.

This document analyses scenarios where all the multi-layer IP links, supported by the optical network, are intra-domain (intra-AS/intra-area), such as PE-BR, PE-P, BR-P, P-P IP links. Therefore the inter-domain IP links are always single-layer links supported by Ethernet physical links.

The analysis of scenarios with multi-layer inter-domain IP links is outside the scope of this document.

Therefore, if inter-domain links between the optical domains exist, they would be used to support multi-domain optical services, which are outside the scope of this document.

The optical network elements (NEs) within the optical domains can be ROADMs or OTN switches, with or without an integrated ROADM function.

2.1. Multi-domain Service Coordinator (MDSC) functions

The MDSC in Figure 1 is responsible for multi-domain and multi-layer coordination across multiple packet and optical domains, as well as to provide multi-layer/multi-domain L2/L3 VPN network services requested by an OSS/Orchestration layer.

From an implementation perspective, the functions associated with MDSC and described in [RFC8453] may be grouped in different ways.

1. Both the service- and network-related functions are collapsed into a single, monolithic implementation, dealing with the end customer service requests received from the CMI (Customer MDSC Interface) and adapting the relevant network models. An example is represented in Figure 2 of [RFC8453].

2. An implementation can choose to split the service-related and the network-related functions into different functional entities, as described in [RFC8309] and in section 4.2 of [RFC8453]. In this case, MDSC is decomposed into a top-level Service Orchestrator, interfacing the customer via the CMI, and into a Network Orchestrator interfacing at the southbound with the PNCs. The interface between the Service Orchestrator and the Network Orchestrator is not specified in [RFC8453].
3. Another implementation can choose to split the MDSC functions between an "higher-level MDSC" (MDSC-H) responsible for packet and optical multi-layer coordination, interfacing with one Optical "lower-level MDSC" (MDSC-L), providing multi-domain coordination between the O-PNCs and one Packet MDSC-L, providing multi-domain coordination between the P-PNCs (see for example Figure 9 of [RFC8453]).

4. Another implementation can also choose to combine the MDSC and the P-PNC functions together.

In the current service provider’s network deployments, at the North Bound of the MDSC, instead of a CNC, typically there is an OSS/Orchestration layer. In this case, the MDSC would implement only the Network Orchestration functions, as in [RFC8309] and described in point 2 above. Therefore, the MDSC is dealing with the network services requests received from the OSS/Orchestration layer.

The functionality of the OSS/Orchestration layer and the interface toward the MDSC are usually operator-specific and outside the scope of this draft. Therefore, this document assumes that the OSS/Orchestrator requests the MDSC to set up L2/L3 VPN network services through mechanisms that are outside the scope of this document.

There are two prominent workflow cases when the MDSC multi-layer coordination is initiated:

- Initiated by a request from the OSS/Orchestration layer to setup L2/L3 VPN network services that requires multi-layer/multi-domain coordination;

- Initiated by the MDSC itself to perform multi-layer/multi-domain optimizations and/or maintenance activities (e.g. rerouting LSPs with their associated services when putting a resource, like a fibre, in maintenance mode during a maintenance window). Unlike service fulfillment, these workflows are not related to a network service provisioning request being received from the OSS/Orchestration layer.

The latter workflow cases are outside the scope of this document.

This document analyses the use cases where multi-layer coordination is triggered by a network service request received from the OSS/Orchestration layer.
2.1.1. Multi-domain L2/L3 VPN network services

Figure 2 provides an example of an hub & spoke multi-domain L2/L3 VPN with three PEs where the hub PE (PE13) and one spoke PE (PE14) are within the same packet domain and the other spoke PE (PE23) is within a different packet domain.
There are many options to implement multi-domain L2/L3 VPNs, including:

1. BGP-LU (seamless MPLS)
2. Inter-domain RSVP-TE
3. Inter-domain SR-TE

This document provides an analysis of the inter-domain SR-TE option. The analysis of other options is outside the scope of this draft.

It is also assumed that:

- each packet domain in Figure 2 is implementing SR-TE and the stitching between two domains is done using end-to-end/multi-domain SR-TE;
- the bandwidth of each intra-domain SR-TE path is managed by its respective P-PNC;
- binding SID is used for the end-to-end SR-TE path stitching;
- each packet domain in Figure 2 is using TI-LFA, with SRLG awareness, for local protection within each domain.

In this scenario, one of the key MDSC functions is to identify the multi-domain/multi-layer SR-TE paths to be used to carry the L2/L3 VPN traffic between PEs belonging to different packet domains and to relay this information to the P-PNCs, to ensure that the PEs’ forwarding tables (e.g., VRF) are properly configured to steer the L2/L3 VPN traffic over the intended multi-domain/multi-layer SR-TE paths.

The selection of the SR-TE path should take into account the TE requirements and the binding requirements for the L2/L3 VPN network service.

In general the binding requirements for a network service (e.g. L2/L3 VPN), can be summarized within three cases:

1. The customer is asking for VPN isolation dynamically creating and binding tunnels to the service such that they are not shared by others services (e.g. VPN).
   The level of isolation can be different:
   a) Hard isolation with deterministic latency that means L2/L3 VPN requiring a set of dedicated TE Tunnels (neither sharing with other services nor competing for bandwidth with other tunnels) providing deterministic latency performances
   b) Hard isolation but without deterministic characteristics
c) Soft isolation that means the tunnels associated with L2/L3 VPN are dedicated to that but can compete for bandwidth with other tunnels.

2. The customer does not ask isolation, and could request a VPN service where associated tunnels can be shared across multiple VPNs.

For each SR-TE path required to support the L2/L3 VPN network service, it is possible that:

1. A SR-TE path that meets the TE and binding requirements already exist in the network.

2. An existing SR-TE path could be modified (e.g., through bandwidth increase) to meet the TE and binding requirements:
   a. The SR-TE path characteristics can be modified only in the packet layer.
   b. One or more new underlay optical tunnels need to be setup to support the requested changes of the overlay SR-TE paths (multi-layer coordination is required).

3. A new SR-TE path needs to be setup to meet the TE and binding requirements:
   a. The new SR-TE path reuses existing underlay optical tunnels;
   b. One or more new underlay optical tunnels need to be setup to support the setup of the new SR-TE path (multi-layer coordination is required).

2.1.2. Multi-domain and multi-layer path computation

When a new SR-TE path needs to be setup, the MDSC is also responsible to coordinate the multi-layer/multi-domain path computation.

Depending on the knowledge that MDSC has of the topology and configuration of the underlying network domains, three approaches for performing multi-layer/multi-domain path computation are possible:
1. Full Summarization: In this approach, the MDSC has an abstracted TE topology view of all of its, packet and optical, underlying domains.

In this case, the MDSC does not have enough TE topology information to perform multi-layer/multi-domain path computation. Therefore the MDSC delegates the P-PNCs and O-PNCs to perform local path computation within their respective controlled domains and it uses the information returned by the P-PNCs and O-PNCs to compute the optimal multi-domain/multi-layer path.

This approach presents an issue to P-PNC, which does not have the capability of performing a single-domain/multi-layer path computation, since it can not retrieve the topology information from the O-PNCs nor delegate the O-PNC to perform optical path computation.

A possible solution could be to include a CNC function within the P-PNC to request the MDSC multi-domain optical path computation, as shown in Figure 10 of [RFC8453].

Another solution could be to rely on the MDSC recursive hierarchy, as defined in section 4.1 of [RFC8453], where, for each IP and optical domain pair, a "lower-level MDSC" (MDSC-L) provides the essential multi-layer correlation and the "higher-level MDSC" (MDSC-H) provides the multi-domain coordination.

In this case, the MDSC-H can get an abstract view of the underlying multi-layer domain topologies from its underlying MDSC-L. Each MDSC-L gets the full view of the IP domain topology from P-PNC and can get an abstracted view of the optical domain topology from its underlying O-PNC. In other words, topology abstraction is possible at the MPIs between MDSC-L and O-PNC and between MDSC-L and MDSC-H.
2. Partial summarization: In this approach, the MDSC has full visibility of the TE topology of the packet network domains and an abstracted view of the TE topology of the optical network domains.

The MDSC then has only the capability of performing multi-domain/single-layer path computation for the packet layer (the path can be computed optimally for the two packet domains).

Therefore, the MDSC still needs to delegate the O-PNCs to perform local path computation within their respective domains and it uses the information received by the O-PNCs, together with its TE topology view of the multi-domain packet layer, to perform multi-layer/multi-domain path computation.

3. Full knowledge: In this approach, the MDSC has the complete and enough detailed view of the TE topology of all the network domains (both optical and packet).

In such case MDSC has all the information needed to perform multi-domain/multi-layer path computation, without relying on PNCs.

This approach may present, as a potential drawback, scalability issues and, as discussed in section 2.2. of [PATH-COMPUTE], performing path computation for optical networks in the MDSC is quite challenging because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

This document analyses scenarios where the MDSC uses the partial summarization approach to coordinate multi-domain/multi-layer path computation.

Typically, the O-PNCs are responsible for the optical path computation of services across their respective single domains. Therefore, when setting up the network service, they must consider the connection requirements such as bandwidth, amplification, wavelength continuity, and non-linear impairments that may affect the network service path.

The methods and types of path requirements and impairments, such as those detailed in [OIA-TOPO], used by the O-PNC for optical path computation are not exposed at the MPI and therefore out of scope for this document.
2.2. IP/MPLS Domain Controller and NE Functions

As highlighted in section 2.1.1, SR-TE is used in the packet domain. Each domain, corresponding to either an IGP area or an Autonomous System (AS) within the same operator network, is controlled by a packet domain controller (P-PNC).

P-PNCs are responsible to setup the SR-TE paths between any two PEs or BRs in their respective controlled domains, as requested by MDSC, and to provide topology information to the MDSC.

With reference to Figure 2, a bidirectional SR-TE path from PE13 in domain 1 to PE23 in domain 2 requires the MDSC to coordinate the actions of:

- P-PNC1 to push a SID list to PE13 including the Binding SID associated to the SR-TE path in Domain 2 with PE23 as the target destination (forward direction);
- P-PNC2 to push a SID list to PE23 with including the Binding SID associated to the SR-TE path in Domain 1 with PE13 as the target destination (reverse direction).

With reference to Figure 3, P-PNCs are then responsible:

1. To expose to MDSC their respective detailed TE topology
2. To perform single-layer single-domain local SR-TE path computation, when requested by MDSC between two PEs (for single-domain end-to-end SR-TE path) or between PEs and BRs for an inter-domain SR-TE path selected by MDSC;
3. To configure the ingress PE or BR router in their respective domain with the SID list associated with an SR-TE path;
4. To configure finally the VRF and PE-CE interfaces (Service access points) of the intra-domain and inter-domain network services requested by the MDSC.
When requesting the setup of a new SR-TE path, the MDSC provides the P-PNCs with the explicit path to be created or modified. In other words, the MDSC can communicate to the P-PNCs the full list of nodes involved in the path (strict mode). In this case, the P-PNC is just responsible to push to headend PE or BR the list of SIDs to create that explicit SR-TE path.

For scalability purposes, in large packet domains, where multiple engineered paths are available between any two nodes, the MDSC can request a loose path, together with per-domain TE constraints, to allow the P-PNC selecting the intra-domain SR-TE path meeting these constraints.

In such a case it is mandatory that P-PNC signals back to the MDSC which path it has chosen so that the MDSC keeps track of the relevant resources utilization.

An example of that comes from Figure 2. The SR-TE path requested by the MDSC touches PE13 - P16 - BR12 - BR21 - PE23. P-PNC2 knows of two possible paths with the same topology metric, e.g. BR21 - P24 - PE23 and BR21 - BR22 - PE23, but with different load. It may prefer then to steer the traffic on the latter because it is less loaded.
This exception is mentioned here for the sake of completeness but since the network considered in this document does not fall in this scenario, in the rest of the paper the assumption is that the MDSC always provides the explicit list of SID(s) to the P-PNCs to setup or modify the SR-TE path.

2.3. Optical Domain Controller and NE Functions

The optical network provides the underlay connectivity services to IP/MPLS networks. The packet and optical multi-layer coordination is done by the MDSC, as shown in Figure 1.

The O-PNC is responsible to:

- provide to the MDSC an abstract TE topology view of its underlying optical network resources;
- perform single-domain local path computation, when requested by the MDSC;
- perform optical tunnel setup, when requested by the MDSC.

The mechanisms used by O-PNC to perform intra-domain topology discovery and path setup are usually vendor-specific and outside the scope of this document.

Depending on the type of optical network, TE topology abstraction, path computation and path setup can be single-layer (either OTN or WDM) or multi-layer OTN/WDM. In the latter case, the multi-layer coordination between the OTN and WDM layers is performed by the O-PNC.

3. Interface protocols and YANG data models for the MPIs

This section describes general assumptions applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, to support the scenarios discussed in this document.

3.1. RESTCONF protocol at the MPIs

The RESTCONF protocol, as defined in [RFC8040], using the JSON representation defined in [RFC7951], is assumed to be used at these interfaces. In addition, extensions to RESTCONF, as defined in [RFC8527], to be compliant with Network Management Datastore Architecture (NMDA) defined in [RFC8342], are assumed to be used as well at these MPI interfaces and also at MDSC NBI interfaces.
3.2. YANG data models at the MPIs

The data models used on these interfaces are assumed to use the YANG 1.1 Data Modeling Language, as defined in [RFC7950].

3.2.1. Common YANG data models at the MPIs

As required in [RFC8040], the "ietf-yang-library" YANG module defined in [RFC8525] is used to allow the MDSC to discover the set of YANG modules supported by each PNC at its MPI.

Both Optical and Packet PNCs use the following common topology YANG data models at the MPI:

- The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345];
- The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model;
- The TE Topology Model, defined in the "ietf-te-topology" YANG module of [RFC8795], which augments the Base Network Topology Model.

Both Optical and Packet PNCs use the common TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL], at the MPI.

All the common YANG data models are generic and augmented by technology-specific YANG modules, as described in the following sections.

Both Optical and Packet PNCs also use the Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO], which augments the TE Topology Model with Ethernet technology-specific information.

Both Optical and Packet PNCs use the following common notifications YANG data models at the MPI:

- Dynamic Subscription to YANG Events and Datastores over RESTCONF as defined in [RFC8650];
- Subscription to YANG Notifications for Datastores updates as defined in [RFC8641].
PNCs and MDSCs are compliant with subscription requirements as stated in [RFC7923].

3.2.2. YANG models at the Optical MPIs

The Optical PNC uses at least one of the following technology-specific topology YANG data models, which augment the generic TE Topology Model:

- The WSON Topology Model, defined in the "ietf-wson-topology" YANG module of [RFC9094];

- the Flexi-grid Topology Model, defined in the "ietf-flexi-grid-topology" YANG module of [Flexi-TOPO];

- the OTN Topology Model, as defined in the "ietf-otn-topology" YANG module of [OTN-TOPO].

The optical PNC uses at least one of the following technology-specific tunnel YANG data models, which augments the generic TE Tunnel Model:

- The WSON Tunnel Model, defined in the "ietf-wson-tunnel" YANG modules of [WSON-TUNNEL];

- the Flexi-grid Tunnel Model, defined in the "ietf-flexi-grid-tunnel" YANG module of [Flexi-TUNNEL];

- the OTN Tunnel Model, defined in the "ietf-otn-tunnel" YANG module of [OTN-TUNNEL].

The optical PNC can optionally use the generic Path Computation YANG RPC, defined in the "ietf-te-path-computation" YANG module of [PATH-COMPUTE].

Note that technology-specific augmentations of the generic path computation RPC for WSON, Flexi-grid and OTN path computation RPCs have been identified as a gap.

The optical PNC uses the Ethernet Client Signal Model, defined in the "ietf-eth-tran-service" YANG module of [CLIENT-SIGNAL].

3.2.3. YANG data models at the Packet MPIs

The Packet PNC also uses at least the following technology-specific topology YANG data models:
The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG module of [RFC8346], which augments the Base Network Topology Model;

- the L3 specific data model including extended TE attributes (e.g. performance derived metrics like latency), defined in "ietf-l3-te-topology" and in "ietf-te-topology-packet" YANG modules of [L3-TE-TOPO];

- the SR Topology Model, defined in the "ietf-sr-mpls-topology" YANG module of [SR-TE-TOPO].

Need to check the need/applicability of the "ietf-l3-te-topology" in this scenario since it is not described in [SR-TE-TOPO].

The packet PNC uses at least the following YANG data models:

- L3VPN Network Model (L3NM), defined in the "ietf-l3vpn-ntw" YANG module of [RFC9182];

- L3NM TE Service Mapping, defined in the "ietf-l3nm-te-service-mapping" YANG module of [TSM];

- L2VPN Network Model (L2NM), defined in the "ietf-l2vpn-ntw" YANG module of [L2NM];

- L2NM TE Service Mapping, defined in the "ietf-l2nm-te-service-mapping" YANG module of [TSM].

3.3. PCEP

[RFC8637] examines the applicability of a Path Computation Element (PCE) [RFC5440] and PCE Communication Protocol (PCEP) to the ACTN framework. It further describes how the PCE architecture applies to ACTN and lists the PCEP extensions that are needed to use PCEP as an ACTN interface. The stateful PCE [RFC8231], PCE-Initiation [RFC8281], stateful Hierarchical PCE (H-PCE) [RFC8751], and PCE as a central controller (PCECC) [RFC8283] are some of the key extensions that enable the use of PCE/PCEP for ACTN.

Since the PCEP supports path computation in the packet and optical networks, PCEP is well suited for inter-layer path computation. [RFC5623] describes a framework for applying the PCE-based architecture to interlayer (G)MPLS traffic engineering. Furthermore, the section 6.1 of [RFC8751] states the H-PCE applicability for inter-layer or POI.
[RFC8637] lists various PCEP extensions that apply to ACTN. It also lists the PCEP extension for optical network and POI.

Note that the PCEP can be used in conjunction with the YANG data models described in the rest of this document. Depending on whether ACTN is deployed in a greenfield or brownfield, two options are possible:

1. The MDSC uses a single RESTCONF/YANG interface towards each PNC to discover all the TE information and request TE tunnels. It may either perform full multi-layer path computation or delegate path computation to the underneath PNCs.

   This approach is desirable for operators from an multi-vendor integration perspective as it is simple, and we need only one type of interface (RESTCONF) and use the relevant YANG data models depending on the operator use case considered. Benefits of having only one protocol for the MPI between MDSC and PNC have been already highlighted in [PATH-COMPUTE].

4. The MDSC uses the RESTCONF/YANG interface towards each PNC to discover all the TE information and requests the creation of TE tunnels. However, it uses PCEP for hierarchical path computation.

   As mentioned in Option 1, from an operator perspective, this option can add integration complexity to have two protocols instead of one, unless the RESTCONF/YANG interface is added to an existing PCEP deployment (brownfield scenario).

Section 4 and section 5 of this draft analyse the case where a single RESTCONF/YANG interface is deployed at the MPI (i.e., option 1 above).

4. Inventory, service and network topology discovery

   In this scenario, the MSDC needs to discover through the underlying PNCs:

   o the network topology, at both optical and IP layers, in terms of nodes and links, including the access links, inter-domain IP links as well as cross-layer links;

   o the optical tunnels supporting multi-layer intra-domain IP links;

   o both intra-domain and inter-domain L2/L3 VPN network services deployed within the network;
o the SR-TE paths supporting those L2/L3 VPN network services;

o the hardware inventory information of IP and optical equipment.

The O-PNC and P-PNC could discover and report the hardware network inventory information of their equipment that is used by the different management layers. In the context of POI, the inventory information of IP and optical equipment can complement the topology views and facilitate the packet/optical multi-layer view, e.g., by providing a mapping between the lowest level LTPs in the topology view and corresponding physical port in the network inventory view.

The MDSC could also discover the entire network inventory information of both IP and optical equipment and correlate this information with the links reported in the network topology.

Reporting the entire inventory and detailed topology information of packet and optical networks to the MDSC may present, as a potential drawback, scalability issues. The analysis of the scalability of this approach and mechanisms to address potential issues is outside the scope of this document.

Each PNC provides to the MDSC the topology view of the domain it controls, as described in section 4.1 and 4.3. The MDSC uses this information to discover the complete topology view of the multi-layer multi-domain network it controls.

The MDSC should also maintain up-to-date inventory, service and network topology databases of both IP and optical layers through the use of IETF notifications through MPI with the PNCs when any network inventory/topology/service change occurs.

It should be possible also to correlate information coming from IP and optical layers (e.g., which port, lambda/OTS1, and direction, is used by a specific IP service on the WDM equipment).

In particular, for the cross-layer links, it is key for MDSC to automatically correlate the information from the PNC network databases about the physical ports from the routers (single link or bundle links for LAG) to client ports in the ROADM.

The analysis of multi-layer fault management is outside the scope of this document. However, the discovered information should be sufficient for the MDSC to easily correlate optical and IP layers alarms to speed-up troubleshooting.
Alarms and event notifications are required between MDSC and PNCs so that any network changes are reported almost in real-time to the MDSC (e.g., NE or link failure). As specified in [RFC7923], MDSC must subscribe to specific objects from PNC YANG datastores for notifications.

4.1. Optical topology discovery

The WSON Topology Model or, alternatively, the Flexi-grid Topology model is used to report the DWDM network topology (e.g., ROADM nodes and links), depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The OTN Topology Model is used to report the OTN network topology (e.g., OTN switching nodes and links), when the OTN switching layer is deployed within the optical domain.

In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware inventory information, the O-PNCs report an abstract optical network topology where:

- one TE node is reported for each optical NE deployed within the optical network domain; and
- one TE link is reported for each OMS link and, optionally, for each OTN link.

The Ethernet Topology Model is used to report the Ethernet client LTPs that terminate the cross-layer links: one Ethernet client LTP is reported for each Ethernet client interface on the optical NEs.

Since the MDSC delegates optical path computation to its underlay O-PNCs, the following information can be abstracted and not reported at the MPI:

- the optical parameters required for optical path computation, such as those detailed in [OIA-TOPO];
- the underlay OTS links and ILAs of OMS links;
- the physical connectivity between the optical transponders and the ROADM.

The optical transponders and, optionally, the OTN access cards, are abstracted at MPI by the O-PNC as Trail Termination Points (TTPs),
defined in [RFC8795], within the optical network topology. This abstraction is valid independently of the fact that optical transponders are physically integrated within the same WDM node or are physically located on a device external to the WDM node since in both cases the optical transponders and the WDM node are under the control of the same O-PNC.

The association between the Ethernet LTPs terminating the Ethernet cross-layer links and the optical TTPs is reported using the Inter Layer Lock (ILL) identifiers, defined in [RFC8795].

All the optical links are intra-domain and they are discovered by O-PNCs, using mechanisms which are outside the scope of this document, and reported at the MPIs within the optical network topology.

In case of a multi-layer DWDM/OTN network domain, multi-layer intra-domain OTN links are supported by underlay DWDM tunnels, which can be either WSON tunnels or, alternatively, Flexi-grid tunnels, depending on whether the DWDM optical network is based on fixed grid or flexible-grid. This relationship is reported by the mechanisms described in section 4.2.

4.2. Optical path discovery

The WSON Tunnel Model or, alternatively, the Flexi-grid Tunnel model, depending on whether the DWDM optical network is based on fixed grid or flexible-grid, is used to report all the DWDM tunnels established within the optical network.

When the OTN switching layer is deployed within the optical domain, the OTN Tunnel Model is used to report all the OTN tunnels established within the optical network.

The Ethernet client signal Model is used to report all the Ethernet connectivity provided by the underlay optical tunnels between Ethernet client LTPs. The underlay optical tunnels can be either DWDM tunnels or, when the optional OTN switching layer is deployed, OTN tunnels.

The DWDM tunnels can be used as underlay tunnels to support either Ethernet client connectivity or multi-layer intra-domain OTN links. In the latter case, the hierarchical-link container, defined in [TE-TUNNEL], is used to reference which multi-layer intra-domain OTN links are supported by the underlay DWDM tunnels.
The O-PNCs report in their operational datastores all the Ethernet client connectivities and all the optical tunnels deployed within their optical domain regardless of the mechanisms being used to set them up, such as the mechanisms described in section 5.2, as well as other mechanism (e.g., static configuration), which are outside the scope of this document.

4.3. Packet topology discovery

The L3 Topology Model, SR Topology Model, TE Topology Model and the TE Packet Topology Model are used together to report the SR-TE network topology, as described in figure 2 of [SR-TE-TOPO].

In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware inventory information as well as to perform multi-domain SR-TE path computation, the P-PNCs report the full SR-TE network, including all the information that is required by the MDSC to perform SR-TE path computation. In particular, one TE node is reported for each router and one TE link is reported for each intra-domain IP link. The SR-TE topology also reports the IP LTPs terminating the inter-domain IP links.

All the intra-domain IP links are discovered by the P-PNCs, using mechanisms, such as LLDP [IEEE 802.1AB], which are outside the scope of this document, and reported at the MPIs within the SR-TE network topology.

The Ethernet Topology Model is used to report the intra-domain Ethernet links supporting the intra-domain IP links as well as the Ethernet LTPs that might terminate cross-layer links, inter-domain Ethernet links or access links, as described in detail in section 4.5 and in section 4.6.

4.4. SR-TE path discovery

This version of the draft assumes that discovery of existing SR-TE paths, including their bandwidth, at the MPI is done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with SR-TE specific augmentations, as outlined in section 1 of [TE-TUNNEL].

Note that technology-specific augmentations of the generic path TE tunnel model for SR-TE path setup and discovery have been identified as a gap.
To enable MDSC to discover the full end-to-end SR-TE path configuration, the SR-TE specific augmentation of the [TE-TUNNEL] should allow the P-PNC to report the SID list assigned to an SR-TE path within its domain.

For example, considering the L3VPN in Figure 2, the PE13-P16-PE14 SR-TE path and the SR-TE path in the reverse direction (between PE14 and PE13) could be reported by the P-PNC1 to the MDSC as TE paths of the same TE tunnel instance. The bandwidth of these TE paths represents the bandwidth allocated by P-PNC1 to the two SR-TE paths, which can be symmetric or asymmetric in the two directions.

The P-PNCs use the TE tunnel model to report, at the MPI, all the SR-TE paths established within their packet domain regardless of the mechanism being used to set them up. In other words, the TE tunnel data model reports within the operational datastore both the SR-TE paths being setup by the MDSC at the MPI, using the mechanisms described in section 5.3, as well as the SR-TE paths being setup by other means, such as static configuration, which are outside the scope of this document.

4.5. Inter-domain link discovery

In the reference network of Figure 1, there are three types of inter-domain links:

- Inter-domain Ethernet links supporting inter-domain IP links between two adjacent IP domains;
- Cross-layer links between an IP domain and an adjacent optical domain;
- Access links between a CE device and a PE router.

All the three types of links are Ethernet links.

It is worth noting that the P-PNC may not be aware whether an Ethernet interface terminates a cross-layer link, an inter-domain Ethernet link or an access link.

It is not yet clarified which model can be used to report the access links between CEs and PEs (e.g., by using the Ethernet Topology Model defined in [CLIENT-TOPO] or by using the SAP Model defined in [SAP]). This has been identified as a gap.
The inter-domain Ethernet links and cross-layer links are discovered by the MDSC using the plug-id attribute, as described in section 4.3 of [RFC8795].

More detailed description of how the plug-id can be used to discover inter-domain links is also provided in section 5.1.4 of [TNBI].

This document considers the following two options for discovering inter-domain links:

1. Static configuration
2. LLDP [IEEE 802.1AB] automatic discovery

Other options are possible but not described in this document.

As outlined in [TNBI], the encoding of the plug-id namespace and the specific LLDP information reported within the plug-id value, such as the Chassis ID and Port ID mandatory TLVs, is implementation specific and needs to be consistent across all the PNCs within the network.

The static configuration requires an administrative burden to configure network-wide unique identifiers: it is therefore more viable for inter-domain Ethernet links. For the cross-layer links, the automatic discovery solution based on LLDP snooping is preferable when possible.

The routers exchange standard LLDP packets as defined in [IEEE 802.1AB] and the optical NEs snoop the LLDP packets received from the local Ethernet interface and report to the O-PNCs the extracted information, such as the Chassis ID, the Port ID, System Name TLVs.

Note that the optical NEs do not actively participate in the LLDP packet exchange and does not send any LLDP packets.

4.5.1. Cross-layer link discovery

The MDSC can discover a cross-layer link by matching the plug-id values of the two Ethernet LTPs reported by two adjacent O-PNC and P-PNC: in case LLDP snooping is used, the P-PNC reports the LLDP information sent by the corresponding Ethernet interface on the router while the O-PNC reports the LLDP information received by the corresponding Ethernet interface on the optical NE, e.g., between LTP 5-0 on PE13 and LTP 7-0 on NE11, as shown in Figure 4.
Notes:

=====

(*) Supporting LTP

Legenda:

=======

O   LTP

----> Supporting LTP

<...> Link discovered by the MDSC

{   } LTP Plug-id reported by the PNC

Figure 4 - Cross-layer link discovery

It is worth noting that the discovery of cross-layer links is based only on the LLDP information sent by the Ethernet interfaces of the routers and received by the Ethernet interfaces of the optical NEs, Therefore the MDSC can discover these links also before overlay multi-layer IP links are setup.
4.5.2. Inter-domain IP link discovery

The MDSC can discover an inter-domain Ethernet link which supports an inter-domain IP link, by matching the plug-id values of the two Ethernet LTPs reported by the two adjacent P-PNCs: the two P-PNCs report the LLDP information being sent and being received from the corresponding Ethernet interfaces, e.g., between the Ethernet LTP 3-1 on BR11 and the Ethernet LTP 4-1 on BR21 shown in Figure 5.
Different information is required to be encoded within the plug-id attribute of the Ethernet LTPs to discover cross-layer links and inter-domain Ethernet links.
If the P-PNC does not know a priori whether an Ethernet interface on a router terminates a cross-layer link or an inter-domain Ethernet link, it has to report at the MPI two Ethernet LTPs representing the same Ethernet interface, e.g., both the Ethernet LTP 3-0 and the Ethernet LTP 3-1, supported by LTP 3-0, shown in Figure 5:

o The physical Ethernet LTP is used to represent the physical adjacency between the router Ethernet interface and either the adjacent router Ethernet interface (in case of a single-layer Ethernet link) or the optical NE Ethernet interface (in case of a multi-layer Ethernet link). Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent by the corresponding router Ethernet interface;

o The logical Ethernet LTP, supported by a physical Ethernet LTP, is used to discover the logical adjacency between router Ethernet interfaces, which can be either single-layer or multi-layer. Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent and received by the corresponding router Ethernet interface.

It is worth noting that in case of an inter-domain single-layer Ethernet link, the physical adjacency between the two router Ethernet interfaces cannot be discovered by the MDSC, using the LLDP information reported in the plug-id attributes, as shown in Figure 5. However, the MDSC may infer these links if it knows a priori, using mechanisms which are outside the scope of this document, that inter-domain Ethernet links are always single-layer, e.g., as shown in Figure 5.

The P-PNC can omit reporting the physical Ethernet LTPs when it knows, by mechanisms which are outside the scope of this document, that the corresponding router Ethernet interfaces terminate single-layer inter-domain Ethernet links.

The MDSC can then discover an inter-domain IP link between the two IP LTPs that are supported by the two Ethernet LTPs terminating an inter-domain Ethernet link, discovered as described in section 4.5.2, e.g., between the IP LTP 3-2 on BR21 and the IP LTP 4-2 on BR22, supported respectively by the Ethernet LTP 3-1 on BR11 and by the Ethernet LTP 4-1 on BR21, as shown in Figure 5.

4.6. Multi-layer IP link discovery

A multi-layer intra-domain IP link and its supporting multi-layer intra-domain Ethernet link are discovered by the P-PNC like any other
intra-domain IP and Ethernet links, as described in section 4.3, and reported at the MPI within the SR-TE and Ethernet network topologies, e.g., as shown in Figure 6.
Notes:
=====
(*) Supporting LTP

Legenda:
========
O   LTP
----> Supporting LTP or Supporting Link or Underlay tunnel
<==> Link discovered by the PNC and reported at the MPI
<...> Link discovered by the MDSC
<~~~> Link inferred by the MDSC
x---x Ethernet client signal
X===X Optical tunnel

Figure 6 - Multi-layer intra-domain Ethernet and IP link discovery

The P-PNC does not report any plug-id information on the Ethernet LTPs terminating intra-domain Ethernet links since these links are discovered by the PNC.

In addition, the P-PNC also reports the physical Ethernet LTPs that terminate the cross-layer links supporting the multi-layer intra-domain Ethernet links, e.g., the Ethernet LTP 5-0 on PE13 and the Ethernet LTP 6-0 on BR11, shown in Figure 6.

The MDSC discovers, using the mechanisms described in section 4.5, which Ethernet cross-layer links support the multi-layer intra-domain Ethernet links, e.g. as shown in Figure 6.

The MDSC also discovers, from the information provided by the O-PNC and described in section 4.2, which optical tunnels support the multi-layer intra-domain IP links and therefore the path within the optical network that supports a multi-layer intra-domain IP link, e.g., as shown in Figure 6.

4.6.1. Single-layer intra-domain IP links

It is worth noting that the P-PNC may not be aware of whether an Ethernet interface on the router terminates a multi-layer or a single-layer intra-domain Ethernet link.

In this case, the P-PNC, always reports two Ethernet LTPs for each Ethernet interface on the router, e.g., the Ethernet LTP 1-0 and 1-1 on PE13, shown in Figure 7.
In this case, the MDSC, using the plug-id information reported in the physical Ethernet LTPs, does not discover any cross-layer link being terminated by the corresponding Ethernet interface. The MDSC may infer the physical intra-domain Ethernet link, e.g., between LTP 1-0 on PE13 and LTP 2-0 on P16, as shown in Figure 7, if it knows a
priori, by mechanisms which are outside the scope of this document, that all the Ethernet interfaces on the routers either terminates a cross-layer link or a single-layer intra-domain Ethernet link.

The P-PNC can omit reporting the physical Ethernet LTP if it knows, by mechanisms which are outside the scope of this document, that the intra-domain Ethernet link is single-layer.

4.7. LAG discovery

TBA

4.8. L2/L3 VPN network services discovery

TBA

4.9. Inventory discovery

The are no YANG data models in IETF that could be used to report at the MPI the whole inventory information discovered by a PNC.

[ RFC8345] had foreseen some work for inventory as an augmentation of the network model, but no YANG data model has been developed so far.

There are also no YANG data models in IETF that could be used to correlate topology information, e.g., a link termination point (LTP), with inventory information, e.g., the physical port supporting an LTP, if any.

Inventory information through MPI and correlation with topology information is identified as a gap requiring further work and outside of the scope of this draft.

5. Establishment of L2/L3 VPN network services with TE requirements

In this scenario the MDSC needs to setup a multi-domain L2VPN or a multi-domain L3VPN with some SLA requirements.

The MDSC receives the request to setup a L2/L3 VPN network service from the OSS/Orchestration layer (see Appendix A).

The MDSC translates the L2/L3 VPN SLA requirements into TE requirements (e.g., bandwidth, TE metric bounds, SRLG disjointness, nodes/links/domains inclusion/exclusion) and find the SR-TE paths that meet these TE requirements (see section 2.1.1).
For example, considering the L3VPN in Figure 2, the MDSC finds that:

- a PE13-P16-PE14 SR-TE path already exists but have not enough bandwidth to support the new L3VPN, as described in section 4.4;
- the IP link(s) between P16 and PE14 has not enough bandwidth to support increasing the bandwidth of that SR-TE path, as described in section 4.3;
- a new underlay optical tunnel could be setup to increase the bandwidth IP link(s) between P16 and PE14 to support increasing the bandwidth of that overlay SR-TE path, as described in section 5.2. The dimensioning of the underlay optical tunnel is decided by the MDSC based on the bandwidth requested by the SR-TE path and on its multi-layer optimization policy, which is an internal MDSC implementation issue.

Considering for example the L3VPN in Figure 2, the MDSC can also decide that a new multi-domain SR-TE path needs to be setup between PE13 and PE23, e.g., either because existing SR-TE paths between PE13 and PE23 are not able to meet the TE and binding requirements of the L2/L3 VPN service or because there is no SR-TE path between PE13 and PE23.

As described in section 2.1.2, with partial summarization, the MDSC will use the TE topology information provided by the P-PNCs and the results of the path computation requests sent to the O-PNCs, as described in section 5.1, to compute the multi-layer/multi-domain path between PE13 and PE23.

For example, the multi-layer/multi-domain performed by the MDSC could require the setup of:

- a new underlay optical tunnel between PE13 and BR11, supporting a new IP link, as described in section 5.2;
- a new underlay optical tunnel between BR21 and P24 to increase the bandwidth of the IP link(s) between BR21 and P24, as described in section 5.2.

When the setup of the L2/L3 VPN network service requires multi-domain and multi-layer coordination, the MDSC is also responsible for coordinating the network configuration required to realize the request network service across the appropriate optical and packet domains.
The MDSC would therefore request:

- the O-PNC1 to setup a new optical tunnel between the ROADMs connected to P16 and PE14, as described in section 5.2;

- the P-PNC1 to update the configuration of the existing IP link, in case of LAG, or configure a new IP link, in case of ECMP, between P16 and PE14, as described in section 5.2;

- the P-PNC1 to update the bandwidth of the selected SR-TE path between PE13 and PE14, as described in section 5.3.

After that, the MDSC requests P-PNC2 to setup an SR-TE path between BR21 and PE23, with an explicit path (BR21, P24, PE23) to constraint this new SR-TE path to use the new underlay optical tunnel setup between BR21 and P24, as described in section 5.3. The P-PNC2, knowing the node and the adjacency SIDs assigned within its domain, can install the proper SR policy, or hierarchical policies, within BR21 and returns to the MDSC the binding SID it has assigned to this policy in BR21.

Then the MDSC requests P-PNC1 to setup an SR-TE path between PE13 and BR11, with an explicit path (PE13, BR11) to constraint this new SR-TE path to use the new underlay optical tunnel setup between PE13 and BR11, specifying also which inter-domain link should be used to send traffic to BR21 and the binding SID that has been assigned by P-PNC2 to the corresponding SR policy in BR21, to be used for the end-to-end SR-TE path stitching, as described in section 5.3. The P-PNC1, knowing also the node and the adjacency SIDs assigned within its domain and the EPE SID assigned by P-PNC1 to the inter-domain link between BR11 and BR21, and the binding SID assigned by P-PNC2, installs the proper policy, or policies, within PE13.

Once the SR-TE paths have been selected and, if needed, setup/modified, the MDSC can request to both P-PNCs to configure the L3VPN and its binding with the selected SR-TE paths using the [RFC9182] and [TSM] YANG data models.

[Editor’s Note] Further investigation is needed to understand how the binding between a L3VPN and this new end-to-end SR-TE path can be configured.

5.1. Optical Path Computation

As described in section 2.1.2, the optical path computation is usually performed by the O-PNCs.
When performing multi-layer/multi-domain path computation, the MDSC can delegate the O-PNC for single-domain optical path computation.

As discussed in [PATH-COMPUTE], there are two options to request an O-PNC to perform optical path computation: either via a "compute-only" TE tunnel path, using the generic TE tunnel YANG data model defined in [TE-TUNNEL] or via the path computation RPC defined in [PATH-COMPUTE].

This draft assumes that the path computation RPC is used.

As described in sections 4.1 and 4.5, there is a one-to-one relationship between the router ports, the cross-layer links and the optical TTPs. Therefore, the properties of an optical path between two optical TTPs, as computed by the O-PNC, can be used by the MDSC to infer the properties of the multi-layer single-domain IP link between the router ports associated with the two optical TTPs.

There are no YANG data models in IETF that could be used to augment the generic path computation RPC with technology-specific attributes.

Optical technology-specific augmentation for the path computation RPC is identified as a gap requiring further work outside of this draft’s scope.

5.2. Multi-layer IP link Setup

To setup a new multi-layer IP link between two router ports, the MDSC requires the O-PNC to setup an optical tunnel (either a WSON Tunnel or a Flexi-grid Tunnel or an OTN Tunnel) within the optical network between the two TTPs associated, as described in section 5.1, with these two router Ethernet interfaces.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two cross-layer links over the optical tunnel using the Ethernet Client Signal Model.

After the optical tunnel has been setup and the client traffic steering configured, the two IP routers can exchange Ethernet packets between themselves, including LLDP messages.

If LLDP [IEEE 802.1AB] or any other discovery mechanisms, which are outside the scope of this document, is used between the adjacency between the two routers’ ports, the P-PNC can automatically discover the underlay multi-layer single-domain Ethernet link being set up by the MDSC and report it to the P-PNC.
Otherwise, if there are no automatic discovery mechanisms, the MDSC can configure this multi-layer single-domain Ethernet link at the MPI of the P-PNC.

The two Ethernet LTPs terminating this multi-layer single-domain Ethernet link are supported by the two underlay Ethernet LTPs terminating the two cross-layer links, e.g., as shown in Figure 6.

After the multi-layer single-domain Ethernet link has been configured, the corresponding multi-lyaer single-domain IP link can also be configured either by the MDSC or by the P-PNC.

This document assumes that this IP link is configured by the P-PNC, when the underlying multi-layer single-domain Ethernet link is either discovered by the P-PNC or configured by the MDSC at the MPI.

[Editor’s Note] Add text for IP link update in case of LAG either here or in a new section.

[Editor’s Note] Add text about the configuration of multi-layer SRLG information (issue #45).

It is worth noting that the list of SRLGs for a multi-layer IP link can be quite long. Implementation-specific mechanisms can be implemented by the MDSC or by the O-PNC to summarize the SRLGs of an optical tunnel. These mechanisms are implementation-specific and have no impact on the YANG models nor on the interoperability at the MPI, but cares have to be taken to avoid missing information.

5.3. SR-TE Path Setup and Update

This version of the draft assumes that SR-TE path setup and update at the MPI could be done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with SR-TE specific augmentations, as also outlined in section 1 of [TE-TUNNEL].

When a new SR-TE path needs to be setup, the MDSC can use the [TE-TUNNEL] model to request the P-PNC to setup TE paths, properly specifying the path constraints, such as the explicit path, to force the P-PNC to setup an SR-TE path that meets the end-to-end TE and biding constraints and uses the optical tunnels setup by the MDSC for the purpose of supporting this new SR-TE path.

The [TE-TUNNEL] model supports requesting the setup of both end-to-end as well as segment TE tunnels (within one domain).
In the latter case, SR-TE specific augmentations of the [TE-TUNNEL] model should be defined to allow the MDSC to configure the binding SIDs to be used for the end to-end SR-TE path stitching and to allow the P-PNC to report the binding SID assigned to the segment TE paths.

The assigned binding SID should be persistent in case router or P-PNC rebooting.

The MDSC can also use the [TE-TUNNEL] model to request the P-PNC to increase the bandwidth allocated to an existing TE path, and, if needed, also on its reverse TE path. The [TE-TUNNEL] model supports both symmetric and asymmetric bandwidth configuration in the two directions.

[Editor’s Note:] Add some text about the protection options (to further discuss whether to put this text here or in section 4.2.2).

The MDSC also request the P-PNC to configure TI-LFA local protection: the mechanisms to request the configuration TI-LFA local protection for SR-TE paths using the [TE-TUNNEL] are a gap in the current YANG models.

The TI-LFA local protection within the P-PNC domain is configured by the P-PNC through implementation specific mechanisms which are outside the scope of this document. The P-PNC takes into account the multi-layer SRLG information, configured by the MDSC as described in section 5.2, when computing the TI-LFA post-convergence path for multi-layer single-domain IP links.

SR-TE path setup and update (e.g., bandwidth increase) through MPI is identified as a gap requiring further work, which is outside of the scope of this draft.

6. Conclusions

The analysis provided in this document has shown that the IETF YANG models described in 3.2 provides useful support for Packet Optical Integration (POI) scenarios for resource discovery (network topology, service, tunnels and network inventory discovery) as well as for supporting multi-layer/multi-domain L2/L3 VPN network services.

Few gaps have been identified to be addressed by the relevant IETF Working Groups:
- network inventory model: this gap has been identified in section 4.9 and the solution in [NETWORK-INVENTORY] has been proposed to resolve it;

- technology-specific augmentations of the path computation RPC, defined in [PATH-COMPUTE] for optical networks: this gap has been identified in section 5.1 and the solution in [OPTICAL-PATH-COMPUTE] has been proposed to resolve it;

- relationship between a common discovery mechanisms applicable to access links, inter-domain IP links and cross-layer links and the UNI topology discover mechanism defined in [SAP]: this gap has been identified in section 4.3;

- a mechanism applicable to the P-PNC NBI to configure the SR-TE paths. Technology-specific augmentations of TE Tunnel model, defined in [TE-TUNNEL], are foreseen in section 1 of [TE-TUNNEL] but not yet defined: this gap has been identified in section 5.3.

7. Security Considerations

Several security considerations have been identified and will be discussed in future versions of this document.

8. Operational Considerations

Telemetry data, such as collecting lower-layer networking health and consideration of network and service performance from POI domain controllers, may be required. These requirements and capabilities will be discussed in future versions of this document.

9. IANA Considerations

This document requires no IANA actions.

10. References

10.1. Normative References


[Flexi-TOPO] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.

Peruzzini et al. Expires September 7, 2022
10.2. Informative References


Appendix A. OSS/Orchestration Layer

The OSS/Orchestration layer is a vital part of the architecture framework for a service provider:

- to abstract (through MDSC and PNCs) the underlying transport network complexity to the Business Systems Support layer;
- to coordinate NFV, Transport (e.g. IP, optical and microwave networks), Fixed Access, Core and Radio domains enabling full automation of end-to-end services to the end customers;
- to enable catalogue-driven service provisioning from external applications (e.g. Customer Portal for Enterprise Business services), orchestrating the design and lifecycle management of these end-to-end transport connectivity services, consuming IP and/or optical transport connectivity services upon request.

As discussed in section 2.1, in this document, the MDSC interfaces with the OSS/Orchestration layer and, therefore, it performs the functions of the Network Orchestrator, defined in [RFC8309].

The OSS/Orchestration layer requests the creation of a network service to the MDSC specifying its end-points (PEs and the interfaces towards the CEs) as well as the network service SLA and then proceeds to configuring accordingly the end-to-end customer service between the CEs in the case of an operator managed service.

A.1. MDSC NBI

As explained in section 2, the OSS/Orchestration layer can request the MDSC to setup L2/L3VPN network services (with or without TE requirements).

Although the OSS/Orchestration layer interface is usually operator-specific, typically it would be using a RESTCONF/YANG interface with a more abstracted version of the MPI YANG data models used for network configuration (e.g. L3NM, L2NM).

Figure 8 shows an example of possible control flow between the OSS/Orchestration layer and the MDSC to instantiate L2/L3 VPN network services, using the YANG data models under the definition in [VN], [L2NM], [RFC9182] and [TSM].
The VN YANG data model, defined in [VN], whose primary focus is the CMI, can also provide VN Service configuration from an orchestrated network service point of view when the L2/L3 VPN network service has TE requirements. However, this model is not used to setup L2/L3 VPN service with no TE requirements.

- It provides the profile of VN in terms of VN members, each of which corresponds to an edge-to-edge link between customer end-points (VNAPs). It also provides the mappings between the VNAPs with the LTPs and the connectivity matrix with the VN member. The associated traffic matrix (e.g., bandwidth, latency, protection level, etc.) of VN member is expressed (i.e., via the TE-topology’s connectivity matrix).

- The model also provides VN-level preference information (e.g., VN member diversity) and VN-level admin-status and operational-status.

- The L2NM and L3NM YANG data models, defined in [L2NM] and [RFC9182], whose primary focus is the MPI, can also be used to provide L2VPN and L3VPN network service configuration from an orchestrated connectivity service point of view.

- The TE & Service Mapping YANG data model [TSM] provides TE-service mapping.
o TE-service mapping provides the mapping between a L2/L3 VPN instance and the corresponding VN instances.

o The TE-service mapping also provides the binding requirements as to how each L2/L3 VPN/VN instance is created concerning the underlay TE tunnels (e.g., whether they require a new and isolated set of TE underlay tunnels or not).

o Site mapping provides the site reference information across L2/L3 VPN Site ID, VN Access Point ID, and the LTP of the access link.
Appendix B. Multi-layer and multi-domain resiliency

B.1. Maintenance Window

Before planned maintenance operation on DWDM network takes place, IP traffic should be moved hitless to another link.

MDSC must reroute IP traffic before the events takes place. It should be possible to lock IP traffic to the protection route until the maintenance event is finished, unless a fault occurs on such path.

B.2. Router port failure

The focus is on client-side protection scheme between IP router and reconfigurable ROADM. Scenario here is to define only one port in the routers and in the ROADM muxponder board at both ends as back-up ports to recover any other port failure on client-side of the ROADM (either on router port side or on muxponder side or on the link between them). When client-side port failure occurs, alarms are raised to MDSC by IP-PNC and O-PNC (port status down, LOS etc.). MDSC checks with OP-PNC(s) that there is no optical failure in the optical layer.

There can be two cases here:

a) LAG was defined between the two end routers. MDSC, after checking that optical layer is fine between the two end ROADMs, triggers the ROADM configuration so that the router back-up port with its associated muxponder port can reuse the OCh that was already in use previously by the failed router port and adds the new link to the LAG on the failure side.

While the ROADM reconfiguration takes place, IP/MPLS traffic is using the reduced bandwidth of the IP link bundle, discarding lower priority traffic if required. Once back-up port has been reconfigured to reuse the existing OCh and new link has been added to the LAG then original Bandwidth is recovered between the end routers.

Note: in this LAG scenario let assume that BFD is running at LAG level so that there is nothing triggered at MPLS level when one of the link member of the LAG fails.
b) If there is no LAG then the scenario is not clear since a router port failure would automatically trigger (through BFD failure) first a sub-50ms protection at MPLS level :FRR (MPLS RSVP-TE case) or TI-LFA (MPLS based SR-TE case) through a protection port. At the same time MDSC, after checking that optical network connection is still fine, would trigger the reconfiguration of the back-up port of the router and of the ROADM muxponder to re-use the same OCh as the one used originally for the failed router port. Once everything has been correctly configured, MDSC Global PCE could suggest to the operator to trigger a possible re-optimization of the back-up MPLS path to go back to the MPLS primary path through the back-up port of the router and the original OCh if overall cost, latency etc. is improved. However, in this scenario, there is a need for protection port PLUS back-up port in the router which does not lead to clear port savings.

Acknowledgments

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

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

Contributors

Sergio Belotti
Nokia

Email: sergio.belotti@nokia.com

Gabriele Galimberti
Cisco

Email: ggalimbe@cisco.com

Zheng Yanlei
China Unicom

Email: zhengyanlei@chinaunicom.cn
Anton Snitser
Sedona
Email: antons@sedonasys.com

Washington Costa Pereira Correia
TIM Brasil
Email: wcorreia@timbrasil.com.br

Michael Scharf
Hochschule Esslingen - University of Applied Sciences
Email: michael.scharf@hs-esslingen.de

Young Lee
Sung Kyun Kwan University
Email: youngleetx@gmail.com

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

Paolo Volpato
Huawei
Email: paolo.volpato@huawei.com

Brent Foster
Cisco
Email: brfoster@cisco.com
Authors’ Addresses

Fabio Peruzzini
TIM
Email: fabio.peruzzini@telecomitalia.it

Jean-François Bouquier
Vodafone
Email: jeff.bouquier@vodafone.com

Italo Busi
Huawei
Email: Italo.busi@huawei.com

Daniel King
Old Dog Consulting
Email: daniel@olddog.co.uk

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com
Applicability of Abstraction and Control of Traffic Engineered Networks (ACTN) to Packet Optical Integration (POI)

draft-ietf-teas-actn-poi-applicability-07

Abstract

This document considers the applicability of Abstraction and Control of TE Networks (ACTN) architecture to Packet Optical Integration (POI) in the context of IP/MPLS and optical internetworking. It identifies the YANG data models being defined by the IETF to support this deployment architecture and specific scenarios relevant for Service Providers.

Existing IETF protocols and data models are identified for each multi-layer (packet over optical) scenario with a specific focus on the MPI (Multi-Domain Service Coordinator to Provisioning Network Controllers Interface) in the ACTN architecture.

Status of this Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."
The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

This Internet-Draft will expire on April 9, 2021.

Copyright Notice

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

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

Table of Contents

1. Introduction...................................................3
   1.1. Terminology...............................................5
2. Reference network architecture...................................7
   2.1. Multi-domain Service Coordinator (MDSC) functions.........9
       2.1.1. Multi-domain L2/L3 VPN network services...........11
       2.1.2. Multi-domain and multi-layer path computation......14
   2.2. IP/MPLS Domain Controller and NE Functions.............17
   2.3. Optical Domain Controller and NE Functions...............19
3. Interface protocols and YANG data models for the MPIs.........19
   3.1. RESTCONF protocol at the MPIs............................19
   3.2. YANG data models at the MPIs.............................20
       3.2.1. Common YANG data models at the MPIs.............20
       3.2.2. YANG models at the Optical MPIs..................21
       3.2.3. YANG data models at the Packet MPIs.............22
   3.3. PCEP.....................................................23
4. Inventory, service and network topology discovery.............24
   4.1. Optical topology discovery................................25
   4.2. Optical path discovery...................................27
   4.3. Packet topology discovery................................27
   4.4. TE path discovery.......................................28
   4.5. Inter-domain link discovery..............................29
1. Introduction

The complete automation of the management and control of Service Providers transport networks (IP/MPLS, optical, and microwave transport networks) is vital for meeting emerging demand for high-bandwidth use cases, including 5G and fiber connectivity services. The Abstraction and Control of TE Networks (ACTN) architecture and interfaces facilitate the automation and operation of complex optical and IP/MPLS networks through standard interfaces and data models. This allows a wide range of network services that can be requested by the upper layers fulfilling almost any kind of service level requirements from a network perspective (e.g. physical diversity, latency, bandwidth, topology, etc.)
Packet Optical Integration (POI) is an advanced use case of traffic engineering. In wide-area networks, a packet network based on the Internet Protocol (IP), and often Multiprotocol Label Switching (MPLS) or Segment Routing (SR), is typically realized on top of an optical transport network that uses Dense Wavelength Division Multiplexing (DWDM) (and optionally an Optical Transport Network (OTN) layer).

In many existing network deployments, the packet and the optical networks are engineered and operated independently. As a result, there are technical differences between the technologies (e.g., routers compared to optical switches) and the corresponding network engineering and planning methods (e.g., inter-domain peering optimization in IP, versus dealing with physical impairments in DWDM, or very different time scales). In addition, customers needs can be different between a packet and an optical network, and it is not uncommon to use different vendors in both domains. The operation of these complex packet and optical networks is often siloed, as these technology domains require specific skills sets.

The packet/optical network deployment and operation separation are inefficient for many reasons. Both capital expenditure (CAPEX) and operational expenditure (OPEX) could be significantly reduced by integrating the packet and the optical networks. Multi-layer online topology insight can speed up troubleshooting (e.g., alarm correlation) and network operation (e.g., coordination of maintenance events), multi-layer offline topology inventory can improve service quality (e.g., detection of diversity constraint violations) and multi-layer traffic engineering can use the available network capacity more efficiently (e.g., coordination of restoration). In addition, provisioning workflows can be simplified or automated as needed across layers (e.g., to achieve bandwidth-on-demand or to perform activities during maintenance windows).

ACTN framework enables this complete multi-layer and multi-vendor integration of packet and optical networks through Multi-Domain Service Coordinator (MDSC) and packet and optical Provisioning Network Controllers (PNCs).

In this document, critical scenarios for POI are described from the packet service layer perspective and identified the required coordination between packet and optical layers to improve POI deployment and operation. Precise definitions of scenarios can help with achieving a common understanding across different disciplines. The focus of the scenarios are multi-domain packet networks operated as a client of optical networks.
This document analyses the case where the packet networks support multi-domain SR-TE paths and the optical networks could be either a DWDM network or an OTN network (without DWDM layer) or multi-layer OTN/DWDM network. DWDM networks could be either fixed-grid or flexible-grid.

Multi-layer and multi-domain scenarios, based on reference network described in section 2, and very relevant for Service Providers, are described in section 4 and in section 5.

For each scenario, existing IETF protocols and data models, identified in section 3.1 and section 3.2, are analysed with particular focus on the MPI in the ACTN architecture.

For each multi-layer scenario, the document analyzes how to use the interfaces and data models of the ACTN architecture.

A summary of the gaps identified in this analysis is provided in section 6.

Understanding the level of standardization and the possible gaps will help assess the feasibility of integration between packet and optical DWDM domains (and optionally OTN layer) in an end-to-end multi-vendor service provisioning perspective.

1.1. Terminology

This document uses the ACTN terminology defined in [RFC8453]

In addition this document uses the following terminology.

Customer service:

the end-to-end service from CE to CE

Network service:

the PE to PE configuration including both the network service layer (VRFs, RT import/export policies configuration) and the network transport layer (e.g. RSVP-TE LSPs). This includes the configuration (on the PE side) of the interface towards the CE (e.g. VLAN, IP address, routing protocol etc.)

Port:

the physical entity that transmits and receives physical signals
Interface:
   a physical or logical entity that transmits and receives traffic

Link:
   an association between two interfaces that can exchange traffic directly

Ethernet link:
   a link between two Ethernet interfaces

IP link:
   a link between two IP interfaces

Cross-layer link:
   an Ethernet link between an Ethernet interface on a router and an Ethernet interface on an optical NE

Intra-domain single-layer Ethernet link:
   an Ethernet link between two Ethernet interfaces on physically adjacent routers that belong to the same P-PNC domain

Intra-domain single-layer IP link:
   an IP link supported by an intra-domain single-layer Ethernet link

Inter-domain single-layer Ethernet link:
   an Ethernet link between two Ethernet interfaces on physically adjacent routers which belong to different P-PNC domains

Inter-domain single-layer IP link:
   an IP link supported by an inter-domain single-layer Ethernet link.

Intra-domain multi-layer Ethernet link:
   an Ethernet link supported by two cross-layer links and an optical tunnel in between
Intra-domain multi-layer IP link:

an IP link supported an intra-domain multi-layer Ethernet link

2. Reference network architecture

This document analyses several deployment scenarios for Packet and Optical Integration (POI) in which ACTN hierarchy is deployed to control a multi-layer and multi-domain network, with two optical domains and two packet domains, as shown in Figure 1:

![Reference Network Diagram](attachment:image.png)

Figure 1 - Reference Network

The ACTN architecture, defined in [RFC8453], is used to control this multi-layer and multi-domain network where each Packet PNC (P-PNC) is responsible for controlling its packet domain and where each Optical PNC (O-PNC) in the above topology is responsible for controlling its...
optical domain. The packet domains controlled by the P-PNCs can be Autonomous Systems (ASes), defined in [RFC1930], or IGP areas, within the same operator network.

The routers between the packet domains can be either AS Boundary Routers (ASBR) or Area Border Router (ABR): in this document, the generic term Border Router (BR) is used to represent either an ASBR or an ABR.

The MDSC is responsible for coordinating the whole multi-domain multi-layer (packet and optical) network. A specific standard interface (MPI) permits MDSC to interact with the different Provisioning Network Controller (O/P-PNCs).

The MPI interface presents an abstracted topology to MDSC hiding technology-specific aspects of the network and hiding topology details depending on the policy chosen regarding the level of abstraction supported. The level of abstraction can be obtained based on P-PNC and O-PNC configuration parameters (e.g., provide the potential connectivity between any PE and any BR in an SR-TE network).

In the reference network of Figure 1, it is assumed that:

- The domain boundaries between the packet and optical domains are congruent. In other words, one optical domain supports connectivity between routers in one and only one packet domain;
- There are no inter-domain physical links between optical domains. Inter-domain physical links exist only:
  - between packet domains (i.e., between BRs belonging to different packet domains): these links are called inter-domain Ethernet or IP links within this document;
  - between packet and optical domains (i.e., between routers and optical NEs): these links are called cross-layer links within this document;
  - between customer sites and the packet network (i.e., between CE devices and PE routers): these links are called access links within this document;
- All the physical interfaces at inter-domain links are Ethernet physical interfaces.
Although the new optical technologies (e.g., QSFP-DD ZR 400G) allows providing DWDM pluggable interfaces on the routers, the deployment of those pluggable optics is not yet widely adopted by the operators. The reason is that most operators are not yet ready to manage packet and optical networks in a single unified domain. The analysis of the unified use case is outside the scope of this draft.

This document analyses scenarios where all the multi-layer IP links, supported by the optical network, are intra-domain (intra-AS/intra-area), such as PE-BR, PE-P, BR-P, P-P IP links. Therefore the inter-domain IP links are always single-layer links supported by Ethernet physical links.

The analysis of scenarios with multi-layer inter-domain IP links is outside the scope of this document.

Therefore, if inter-domain links between the optical domains exist, they would be used to support multi-domain optical services, which are outside the scope of this document.

The optical network elements (NEs) within the optical domains can be ROADMs or OTN switches, with or without an integrated ROADM function.

2.1. Multi-domain Service Coordinator (MDSC) functions

The MDSC in Figure 1 is responsible for multi-domain and multi-layer coordination across multiple packet and optical domains, as well as to provide multi-layer/multi-domain L2/L3 VPN network services requested by an OSS/Orchestration layer.

From an implementation perspective, the functions associated with MDSC and described in [RFC8453] may be grouped in different ways.

1. Both the service- and network-related functions are collapsed into a single, monolithic implementation, dealing with the end customer service requests received from the CMI (Customer MDSC Interface) and adapting the relevant network models. An example is represented in Figure 2 of [RFC8453].

2. An implementation can choose to split the service-related and the network-related functions into different functional entities, as described in [RFC8309] and in section 4.2 of [RFC8453]. In this case, MDSC is decomposed into a top-level Service Orchestrator, interfacing the customer via the CMI, and into a Network Orchestrator interfacing at the southbound with the PNCs. The interface between the Service Orchestrator and the Network Orchestrator is not specified in [RFC8453].
3. Another implementation can choose to split the MDSC functions between an "higher-level MDSC" (MDSC-H) responsible for packet and optical multi-layer coordination, interfacing with one Optical "lower-level MDSC" (MDSC-L), providing multi-domain coordination between the O-PNCs and one Packet MDSC-L, providing multi-domain coordination between the P-PNCs (see for example Figure 9 of [RFC8453]).

4. Another implementation can also choose to combine the MDSC and the P-PNC functions together.

In the current service provider’s network deployments, at the North Bound of the MDSC, instead of a CNC, typically there is an OSS/Orchestration layer. In this case, the MDSC would implement only the Network Orchestration functions, as in [RFC8309] and described in point 2 above. Therefore, the MDSC is dealing with the network services requests received from the OSS/Orchestration layer.

The functionality of the OSS/Orchestration layer and the interface toward the MDSC are usually operator-specific and outside the scope of this draft. Therefore, this document assumes that the OSS/Orchestrator requests the MDSC to set up L2/L3 VPN network services through mechanisms that are outside the scope of this document.

There are two prominent workflow cases when the MDSC multi-layer coordination is initiated:

- Initiated by a request from the OSS/Orchestration layer to setup L2/L3 VPN network services that requires multi-layer/multi-domain coordination;

- Initiated by the MDSC itself to perform multi-layer/multi-domain optimizations and/or maintenance activities (e.g. rerouting LSPs with their associated services when putting a resource, like a fibre, in maintenance mode during a maintenance window). Unlike service fulfillment, these workflows are not related to a network service provisioning request being received from the OSS/Orchestration layer.

The latter workflow cases are outside the scope of this document.

This document analyses the use cases where multi-layer coordination is triggered by a network service request received from the OSS/Orchestration layer.
2.1.1. Multi-domain L2/L3 VPN network services

Figure 2 provides an example of an hub & spoke multi-domain L2/L3 VPN with three PEs where the hub PE (PE13) and one spoke PE (PE14) are within the same packet domain and the other spoke PE (PE23) is within a different packet domain.
There are many options to implement multi-domain L2/L3 VPNs, including:

1. BGP-LU (seamless MPLS)
2. Inter-domain RSVP-TE
3. Inter-domain SR-TE

This document provides an analysis of the inter-domain TE options, such as inter-domain SR-TE, for which the TE tunnel model, defined in [TE-TUNNEL], could be used at the MPI for intra-domain or inter-domain TE configuration. The analysis of other options is outside the scope of this draft.

It is also assumed that:

- the bandwidth of each intra-domain TE path is managed by its respective P-PNC;
- technology-specific mechanisms (such as, in case of inter-domain SR-TE, the binding SID) are used for the inter-domain TE path stitching;
- each packet domain in Figure 2 is using technology-specific local protection mechanisms (such as, in case of SR-TE, TI-LFA), with SRLG awareness.

In case of inter-domain SR-TE, it is also assumed that each packet domain in Figure 2 is implementing SR-TE and the stitching between two domains is done using inter-domain SR-TE.

In this scenario, one of the key MDSC functions is to identify the multi-domain/multi-layer TE paths to be used to carry the L2/L3 VPN traffic between PEs belonging to different packet domains and to relay this information to the P-PNCs, to ensure that the PEs’ forwarding tables (e.g., VRF) are properly configured to steer the L2/L3 VPN traffic over the intended multi-domain/multi-layer TE paths.

The selection of the TE path should take into account the TE requirements and the binding requirements for the L2/L3 VPN network service.

In general the binding requirements for a network service (e.g. L2/L3 VPN), can be summarized within three cases:

1. The customer is asking for VPN isolation dynamically creating and binding tunnels to the service such that they are not shared by others services (e.g. VPN).
   The level of isolation can be different:
   a) Hard isolation with deterministic latency that means L2/L3 VPN requiring a set of dedicated TE Tunnels (neither
sharing with other services nor competing for bandwidth with other tunnels) providing deterministic latency performances

b) Hard isolation but without deterministic characteristics
c) Soft isolation that means the tunnels associated with L2/L3 VPN are dedicated to that but can compete for bandwidth with other tunnels.

2. The customer does not ask isolation, and could request a VPN service where associated tunnels can be shared across multiple VPNs.

For each TE path (e.g., each SR-TE path) required to support the L2/L3 VPN network service, it is possible that:

1. A TE path that meets the TE and binding requirements already exist in the network.

2. An existing TE path could be modified (e.g., through bandwidth increase) to meet the TE and binding requirements:
   a. The TE path characteristics can be modified only in the packet layer.
   b. One or more new underlay optical tunnels need to be setup to support the requested changes of the overlay TE paths (multi-layer coordination is required).

3. A new TE path needs to be setup to meet the TE and binding requirements:
   a. The new TE path reuses existing underlay optical tunnels;
   b. One or more new underlay optical tunnels need to be setup to support the setup of the new TE path (multi-layer coordination is required).

2.1.2. Multi-domain and multi-layer path computation

When a new TE path (e.g., a new SR-TE path) needs to be setup, the MDSC is also responsible to coordinate the multi-layer/multi-domain path computation.

Depending on the knowledge that MDSC has of the topology and configuration of the underlying network domains, three approaches for performing multi-layer/multi-domain path computation are possible:
1. Full Summarization: In this approach, the MDSC has an abstracted TE topology view of all of its, packet and optical, underlying domains.

In this case, the MDSC does not have enough TE topology information to perform multi-layer/multi-domain path computation. Therefore the MDSC delegates the P-PNCs and O-PNCs to perform local path computation within their respective controlled domains and it uses the information returned by the P-PNCs and O-PNCs to compute the optimal multi-domain/multi-layer path.

This approach presents an issue to P-PNC, which does not have the capability of performing a single-domain/multi-layer path computation, since it can not retrieve the topology information from the O-PNCs nor delegate the O-PNC to perform optical path computation.

A possible solution could be to include a CNC function within the P-PNC to request the MDSC multi-domain optical path computation, as shown in Figure 10 of [RFC8453].

Another solution could be to rely on the MDSC recursive hierarchy, as defined in section 4.1 of [RFC8453], where, for each IP and optical domain pair, a "lower-level MDSC" (MDSC-L) provides the essential multi-layer correlation and the "higher-level MDSC" (MDSC-H) provides the multi-domain coordination.

In this case, the MDSC-H can get an abstract view of the underlying multi-layer domain topologies from its underlying MDSC-L. Each MDSC-L gets the full view of the IP domain topology from P-PNC and can get an abstracted view of the optical domain topology from its underlying O-PNC. In other words, topology abstraction is possible at the MPIs between MDSC-L and O-PNC and between MDSC-L and MDSC-H.
2. Partial summarization: In this approach, the MDSC has full visibility of the TE topology of the packet network domains and an abstracted view of the TE topology of the optical network domains.

The MDSC then has only the capability of performing multi-domain/single-layer path computation for the packet layer (the path can be computed optimally for the two packet domains).

Therefore, the MDSC still needs to delegate the O-PNCs to perform local path computation within their respective domains and it uses the information received by the O-PNCs, together with its TE topology view of the multi-domain packet layer, to perform multi-layer/multi-domain path computation.

3. Full knowledge: In this approach, the MDSC has the complete and enough detailed view of the TE topology of all the network domains (both optical and packet).

In such case MDSC has all the information needed to perform multi-domain/multi-layer path computation, without relying on PNCs.

This approach may present, as a potential drawback, scalability issues and, as discussed in section 2.2. of [PATH-COMPUTE], performing path computation for optical networks in the MDSC is quite challenging because the optimal paths depend also on vendor-specific optical attributes (which may be different in the two domains if they are provided by different vendors).

This document analyses scenarios where the MDSC uses the partial summarization approach to coordinate multi-domain/multi-layer path computation.

Typically, the O-PNCs are responsible for the optical path computation of services across their respective single domains. Therefore, when setting up the network service, they must consider the connection requirements such as bandwidth, amplification, wavelength continuity, and non-linear impairments that may affect the network service path.

The methods and types of path requirements and impairments, such as those detailed in [OIA-TOPO], used by the O-PNC for optical path computation are not exposed at the MPI and therefore out of scope for this document.
2.2. IP/MPLS Domain Controller and NE Functions

Each packet domain in Figure 1, corresponding to either an IGP area or an Autonomous System (AS) within the same operator network, is controlled by a packet domain controller (P-PNC).

P-PNCs are responsible to setup the TE paths between any two PEs or BRs in their respective controlled domains, as requested by MDSC, and to provide topology information to the MDSC.

For example, with reference to Figure 2, a bidirectional SR-TE path from PE13 in domain 1 to PE23 in domain 2 requires the MDSC to coordinate the actions of:

- P-PNC1 to push a SID list to PE13 including the Binding SID associated to the SR-TE path in Domain 2 with PE23 as the target destination (forward direction);
- P-PNC2 to push a SID list to PE23 with including the Binding SID associated to the SR-TE path in Domain 1 with PE13 as the target destination (reverse direction).

With reference to Figure 3, P-PNCs are then responsible:

1. To expose to MDSC their respective detailed TE topology

2. To perform single-layer single-domain local TE path computation, when requested by MDSC between two PEs (for single-domain end-to-end TE path) or between PEs and BRs for an inter-domain TE path selected by MDSC;

3. To configure the routers in their respective domain to setup a TE path: for example, in case of SR-TE, to configure the ingress PE or BR router with the SID list associated with an SR-TE path;

4. To configure finally the VRF and PE-CE interfaces (Service access points) of the intra-domain and inter-domain network services requested by the MDSC.
When requesting the setup of a new TE path, the MDSC provides the P-PNCs with the explicit path to be created or modified. In other words, the MDSC can communicate to the P-PNCs the full list of nodes involved in the path (strict mode). In this case, the P-PNC is just responsible to set up that explicit TE path. For example:

- with SR-TE, the P-PNC just pushes to headend PE or BR the list of SIDs to create the explicit SR-TE path, provided by the MDSC;
- with RSVP-TE, the P-PNC requests the headend PE or BR to start the signaling of the explicit RSVP-TE path, provided by the MDSC.

For scalability purposes, in large packet domains using SR-TE, where multiple engineered paths are available between any two nodes, the MDSC can request a loose path, together with per-domain TE constraints, to allow the P-PNC selecting the intra-domain SR-TE path meeting these constraints.

In such a case it is mandatory that P-PNC signals back to the MDSC which path it has chosen so that the MDSC keeps track of the relevant resources utilization.

An example of that comes from Figure 2. The SR-TE path requested by the MDSC touches PE13 - P16 - BR12 - BR21 - PE23. P-PNC2 knows of two
possible paths with the same topology metric, e.g. BR21 - P24 - PE23 and BR21 - BR22 - PE23, but with different load. It may prefer then to steer the traffic on the latter because it is less loaded.

This exception is mentioned here for the sake of completeness but since the network considered in this document does not fall in this scenario, in the rest of the paper the assumption is that the MDSC always provides the explicit list of SID(s) to the P-PNCs to setup or modify the SR-TE path.

2.3. Optical Domain Controller and NE Functions

The optical network provides the underlay connectivity services to IP/MPLS networks. The packet and optical multi-layer coordination is done by the MDSC, as shown in Figure 1.

The O-PNC is responsible to:

- provide to the MDSC an abstract TE topology view of its underlying optical network resources;
- perform single-domain local path computation, when requested by the MDSC;
- perform optical tunnel setup, when requested by the MDSC.

The mechanisms used by O-PNC to perform intra-domain topology discovery and path setup are usually vendor-specific and outside the scope of this document.

Depending on the type of optical network, TE topology abstraction, path computation and path setup can be single-layer (either OTN or WDM) or multi-layer OTN/WDM. In the latter case, the multi-layer coordination between the OTN and WDM layers is performed by the O-PNC.

3. Interface protocols and YANG data models for the MPIs

This section describes general assumptions applicable at all the MPI interfaces, between each PNC (Optical or Packet) and the MDSC, to support the scenarios discussed in this document.

3.1. RESTCONF protocol at the MPIs

The RESTCONF protocol, as defined in [RFC8040], using the JSON representation defined in [RFC7951], is assumed to be used at these
interfaces. In addition, extensions to RESTCONF, as defined in [RFC8527], to be compliant with Network Management Datastore Architecture (NMDA) defined in [RFC8342], are assumed to be used as well at these MPI interfaces and also at MDSC NBI interfaces.

3.2. YANG data models at the MPIs

The data models used on these interfaces are assumed to use the YANG 1.1 Data Modeling Language, as defined in [RFC7950].

3.2.1. Common YANG data models at the MPIs

As required in [RFC8040], the "ietf-yang-library" YANG module defined in [RFC8525] is used to allow the MDSC to discover the set of YANG modules supported by each PNC at its MPI.

Both Optical and Packet PNCs use the following common topology YANG data models at the MPI:

- The Base Network Model, defined in the "ietf-network" YANG module of [RFC8345];
- The Base Network Topology Model, defined in the "ietf-network-topology" YANG module of [RFC8345], which augments the Base Network Model;
- The TE Topology Model, defined in the "ietf-te-topology" YANG module of [RFC8795], which augments the Base Network Topology Model.

Both Optical and Packet PNCs use the common TE Tunnel Model, defined in the "ietf-te" YANG module of [TE-TUNNEL], at the MPI.

All the common YANG data models are generic and augmented by technology-specific YANG modules, as described in the following sections.

Both Optical and Packet PNCs also use the Ethernet Topology Model, defined in the "ietf-eth-te-topology" YANG module of [CLIENT-TOPO], which augments the TE Topology Model with Ethernet technology-specific information.

Both Optical and Packet PNCs use the following common notifications YANG data models at the MPI:
o Dynamic Subscription to YANG Events and Datastores over RESTCONF as defined in [RFC8650];

o Subscription to YANG Notifications for Datastores updates as defined in [RFC8641].

PNCs and MDSCs are compliant with subscription requirements as stated in [RFC7923].

3.2.2. YANG models at the Optical MPIs

The Optical PNC uses at least one of the following technology-specific topology YANG data models, which augment the generic TE Topology Model:

o The WSON Topology Model, defined in the "ietf-wson-topology" YANG module of [RFC9094];

o the Flexi-grid Topology Model, defined in the "ietf-flexi-grid-topology" YANG module of [Flexi-TOPO];

o the OTN Topology Model, as defined in the "ietf-otn-topology" YANG module of [OTN-TOPO].

The optical PNC uses at least one of the following technology-specific tunnel YANG data models, which augments the generic TE Tunnel Model:

o The WSON Tunnel Model, defined in the "ietf-wson-tunnel" YANG modules of [WSON-TUNNEL];

o the Flexi-grid Tunnel Model, defined in the "ietf-flexi-grid-tunnel" YANG module of [Flexi-TUNNEL];

o the OTN Tunnel Model, defined in the "ietf-otn-tunnel" YANG module of [OTN-TUNNEL].

The optical PNC can optionally use the generic Path Computation YANG RPC, defined in the "ietf-te-path-computation" YANG module of [PATH-COMPUTE].

Note that technology-specific augmentations of the generic path computation RPC for WSON, Flexi-grid and OTN path computation RPCs have been identified as a gap.
The optical PNC uses the Ethernet Client Signal Model, defined in the "ietf-eth-tran-service" YANG module of [CLIENT-SIGNAL].

3.2.3. YANG data models at the Packet MPIs

The Packet PNC also at least the following technology-specific topology YANG data models:

- The L3 Topology Model, defined in the "ietf-l3-unicast-topology" YANG module of [RFC8346], which augments the Base Network Topology Model;
- the L3 specific data model including extended TE attributes (e.g. performance derived metrics like latency), defined in "ietf-l3-te-topology" and in "ietf-te-topology-packet" YANG modules of [L3-TE-TOPO];

Need to check the need/applicability of the "ietf-l3-te-topology" in this scenario since it is not described in [SR-TE-TOPO].

The Packet PNC also uses at least one the following technology-specific topology YANG data models:

- The MPLS-TE Topology Model, defined in the "ietf-te-mpls-topology" YANG module of [MPLS-TE-TOPO], which augments the TE Packet Topology Model;
- the SR Topology Model, defined in the "ietf-sr-mpls-topology" YANG module of [SR-TE-TOPO].

The Packet PNC uses at least one of the following technology-specific tunnel YANG data models, which augments the generic TE Tunnel Model:

- The MPLS-TE Tunnel Model, defined in the "ietf-te-mpls" YANG modules of [MPLS-TE-TUNNEL];
- the SR-TE Tunnel Model which is to be defined as described in section 6.

The packet PNC uses at least the following YANG data models:

- L3VPN Network Model (L3NM), defined in the "ietf-l3vpn-ntw" YANG module of [RFC9182];
- L3NM TE Service Mapping, defined in the "ietf-l3nm-te-service-mapping" YANG module of [TSM];
L2VPN Network Model (L2NM), defined in the "ietf-l2vpn-ntw" YANG module of [L2NM];

o L2NM TE Service Mapping, defined in the "ietf-l2nm-te-service-mapping" YANG module of [TSM].

3.3. PCEP

[RFC8637] examines the applicability of a Path Computation Element (PCE) [RFC5440] and PCE Communication Protocol (PCEP) to the ACTN framework. It further describes how the PCE architecture applies to ACTN and lists the PCEP extensions that are needed to use PCEP as an ACTN interface. The stateful PCE [RFC8231], PCE-Initiation [RFC8281], stateful Hierarchical PCE (H-PCE) [RFC8751], and PCE as a central controller (PCECC) [RFC8283] are some of the key extensions that enable the use of PCE/PCEP for ACTN.

Since the PCEP supports path computation in the packet and optical networks, PCEP is well suited for inter-layer path computation. [RFC5623] describes a framework for applying the PCE-based architecture to interlayer (G)MPLS traffic engineering. Furthermore, the section 6.1 of [RFC8751] states the H-PCE applicability for inter-layer or POI.

[RFC8637] lists various PCEP extensions that apply to ACTN. It also lists the PCEP extension for optical network and POI.

Note that the PCEP can be used in conjunction with the YANG data models described in the rest of this document. Depending on whether ACTN is deployed in a greenfield or brownfield, two options are possible:

1. The MDSC uses a single RESTCONF/YANG interface towards each PNC to discover all the TE information and request TE tunnels. It may either perform full multi-layer path computation or delegate path computation to the underneath PNCs.

   This approach is desirable for operators from an multi-vendor integration perspective as it is simple, and we need only one type of interface (RESTCONF) and use the relevant YANG data models depending on the operator use case considered. Benefits of having only one protocol for the MPI between MDSC and PNC have been already highlighted in [PATH-COMPUTE].
4. The MDSC uses the RESTCONF/YANG interface towards each PNC to discover all the TE information and requests the creation of TE tunnels. However, it uses PCEP for hierarchical path computation.

As mentioned in Option 1, from an operator perspective, this option can add integration complexity to have two protocols instead of one, unless the RESTCONF/YANG interface is added to an existing PCEP deployment (brownfield scenario).

Section 4 and section 5 of this draft analyse the case where a single RESTCONF/YANG interface is deployed at the MPI (i.e., option 1 above).

4. Inventory, service and network topology discovery

In this scenario, the MSDC needs to discover through the underlying PNCs:

- the network topology, at both optical and IP layers, in terms of nodes and links, including the access links, inter-domain IP links as well as cross-layer links;
- the optical tunnels supporting multi-layer intra-domain IP links;
- both intra-domain and inter-domain L2/L3 VPN network services deployed within the network;
- the TE paths supporting those L2/L3 VPN network services;
- the hardware inventory information of IP and optical equipment.

The O-PNC and P-PNC could discover and report the hardware network inventory information of their equipment that is used by the different management layers. In the context of POI, the inventory information of IP and optical equipment can complement the topology views and facilitate the packet/optical multi-layer view, e.g., by providing a mapping between the lowest level LTPs in the topology view and corresponding physical port in the network inventory view.

The MDSC could also discover the entire network inventory information of both IP and optical equipment and correlate this information with the links reported in the network topology.

Reporting the entire inventory and detailed topology information of packet and optical networks to the MDSC may present, as a potential drawback, scalability issues. The analysis of the scalability of this
approach and mechanisms to address potential issues is outside the scope of this document.

Each PNC provides to the MDSC the topology view of the domain it controls, as described in section 4.1 and 4.3. The MDSC uses this information to discover the complete topology view of the multi-layer multi-domain network it controls.

The MDSC should also maintain up-to-date inventory, service and network topology databases of both IP and optical layers through the use of IETF notifications through MPI with the PNCs when any network inventory/topology/service change occurs.

It should be possible also to correlate information coming from IP and optical layers (e.g., which port, lambda/OTSi, and direction, is used by a specific IP service on the WDM equipment).

In particular, for the cross-layer links, it is key for MDSC to automatically correlate the information from the PNC network databases about the physical ports from the routers (single link or bundle links for LAG) to client ports in the ROADM.

The analysis of multi-layer fault management is outside the scope of this document. However, the discovered information should be sufficient for the MDSC to easily correlate optical and IP layers alarms to speed-up troubleshooting.

Alarms and event notifications are required between MDSC and PNCs so that any network changes are reported almost in real-time to the MDSC (e.g., NE or link failure). As specified in [RFC7923], MDSC must subscribe to specific objects from PNC YANG datastores for notifications.

4.1. Optical topology discovery

The WSON Topology Model or, alternatively, the Flexi-grid Topology model is used to report the DWDM network topology (e.g., ROADM nodes and links), depending on whether the DWDM optical network is based on fixed grid or flexible-grid.

The OTN Topology Model is used to report the OTN network topology (e.g., OTN switching nodes and links), when the OTN switching layer is deployed within the optical domain.

In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware
inventory information, the O-PNCs report an abstract optical network topology where:

- one TE node is reported for each optical NE deployed within the optical network domain; and
- one TE link is reported for each OMS link and, optionally, for each OTN link.

The Ethernet Topology Model is used to report the Ethernet client LTPs that terminate the cross-layer links: one Ethernet client LTP is reported for each Ethernet client interface on the optical NEs.

Since the MDSC delegates optical path computation to its underlay O-PNCs, the following information can be abstracted and not reported at the MPI:

- the optical parameters required for optical path computation, such as those detailed in [OIA-TOPO];
- the underlay OTS links and ILAs of OMS links;
- the physical connectivity between the optical transponders and the ROADMs.

The optical transponders and, optionally, the OTN access cards, are abstracted at MPI by the O-PNC as Trail Termination Points (TTPs), defined in [RFC8795], within the optical network topology. This abstraction is valid independently of the fact that optical transponders are physically integrated within the same WDM node or are physically located on a device external to the WDM node since it both cases the optical transponders and the WDM node are under the control of the same O-PNC.

The association between the Ethernet LTPs terminating the Ethernet cross-layer links and the optical TTPs is reported using the Inter Layer Lock (ILL) identifiers, defined in [RFC8795].

All the optical links are intra-domain and they are discovered by O-PNCs, using mechanisms which are outside the scope of this document, and reported at the MPIs within the optical network topology.
In case of a multi-layer DWDM/OTN network domain, multi-layer intra-domain OTN links are supported by underlay DWDM tunnels, which can be either WSON tunnels or, alternatively, Flexi-grid tunnels, depending on whether the DWDM optical network is based on fixed grid or flexible-grid. This relationship is reported by the mechanisms described in section 4.2.

4.2. Optical path discovery

The WSON Tunnel Model or, alternatively, the Flexi-grid Tunnel model, depending on whether the DWDM optical network is based on fixed grid or flexible-grid, is used to report all the DWDM tunnels established within the optical network.

When the OTN switching layer is deployed within the optical domain, the OTN Tunnel Model is used to report all the OTN tunnels established within the optical network.

The Ethernet client signal Model is used to report all the Ethernet connectivity provided by the underlay optical tunnels between Ethernet client LTPs. The underlay optical tunnels can be either DWDM tunnels or, when the optional OTN switching layer is deployed, OTN tunnels.

The DWDM tunnels can be used as underlay tunnels to support either Ethernet client connectivity or multi-layer intra-domain OTN links. In the latter case, the hierarchical-link container, defined in [TE-TUNNEL], is used to reference which multi-layer intra-domain OTN links are supported by the underlay DWDM tunnels.

The O-PNCs report in their operational datastores all the Ethernet client connectivities and all the optical tunnels deployed within their optical domain regardless of the mechanisms being used to set them up, such as the mechanisms described in section 5.2, as well as other mechanism (e.g., static configuration), which are outside the scope of this document.

4.3. Packet topology discovery

The L3 Topology Model, SR Topology Model, TE Topology Model and the TE Packet Topology Model are used together to report the SR-TE network topology, as described in figure 2 of [SR-TE-TOPO].

The L3 Topology Model, TE Topology Model, TE Packet Topology Model and MPLS-TE Topology Model are used together to report the SR-TE network topology, as described in [MPLS-TE-TOPO].
In order to allow the MDSC to discover the complete multi-layer and multi-domain network topology and to correlate it with the hardware inventory information as well as to perform multi-domain TE path computation, the P-PNCs report the full packet network, including all the information that is required by the MDSC to perform TE path computation. In particular, one TE node is reported for each router and one TE link is reported for each intra-domain IP link. The packet topology also reports the IP LTPs terminating the inter-domain IP links.

All the intra-domain IP links are discovered by the P-PNCs, using mechanisms, such as LLDP [IEEE 802.1AB], which are outside the scope of this document, and reported at the MPIs within the packet network topology.

The Ethernet Topology Model is used to report the intra-domain Ethernet links supporting the intra-domain IP links as well as the Ethernet LTPs that might terminate cross-layer links, inter-domain Ethernet links or access links, as described in detail in section 4.5 and in section 4.6.

4.4. TE path discovery

This version of the draft assumes that discovery of existing TE paths, including their bandwidth, at the MPI is done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with packet technology-specific (e.g., MPLS-TE or SR-TE) augmentations.

Note that technology-specific augmentations of the generic path TE tunnel model for SR-TE path setup and discovery is outlined in section 1 of [TE-TUNNEL] but currently identified as a gap in section 6.

To enable MDSC to discover the full end-to-end TE path configuration, the technology-specific augmentation of the [TE-TUNNEL] should allow the P-PNC to report the TE path within its domain (e.g., the SID list assigned to an SR-TE path).

For example, considering the L3VPN in Figure 2, the PE13-P16-PE14 TE path and the TE path in the reverse direction (between PE14 and PE13) could be reported by the P-PNC1 to the MDSC as primary and reverse primary TE paths of the same TE tunnel instance. The bandwidth of these TE paths represents the bandwidth allocated by P-PNC1 to the two TE paths, which can be symmetric or asymmetric in the two directions.
The P-PNCs use the TE tunnel model to report, at the MPI, all the TE paths established within their packet domain regardless of the mechanism being used to set them up; i.e., independently on whether the mechanisms described in section 5.3 or other means, such as static configuration, which are outside the scope of this document, are used.

4.5. Inter-domain link discovery

In the reference network of Figure 1, there are three types of inter-domain links:

- Inter-domain Ethernet links supporting inter-domain IP links between two adjacent IP domains;
- Cross-layer links between an IP domain and an adjacent optical domain;
- Access links between a CE device and a PE router.

All the three types of links are Ethernet links.

It is worth noting that the P-PNC may not be aware whether an Ethernet interface terminates a cross-layer link, an inter-domain Ethernet link or an access link.

It is not yet clarified which model can be used to report the access links between CEs and PEs (e.g., by using the Ethernet Topology Model defined in [CLIENT-TOPO] or by using the SAP Model defined in [SAP]). This has been identified as a gap.

The inter-domain Ethernet links and cross-layer links are discovered by the MDSC using the plug-id attribute, as described in section 4.3 of [RFC8795].

More detailed description of how the plug-id can be used to discover inter-domain links is also provided in section 5.1.4 of [TNBI].

This document considers the following two options for discovering inter-domain links:

1. Static configuration
2. LLDP [IEEE 802.1AB] automatic discovery

Other options are possible but not described in this document.
As outlined in [TNBI], the encoding of the plug-id namespace and the specific LLDP information reported within the plug-id value, such as the Chassis ID and Port ID mandatory TLVs, is implementation specific and needs to be consistent across all the PNCs within the network.

The static configuration requires an administrative burden to configure network-wide unique identifiers: it is therefore more viable for inter-domain Ethernet links. For the cross-layer links, the automatic discovery solution based on LLDP snooping is preferable when possible.

The routers exchange standard LLDP packets as defined in [IEEE 802.1AB] and the optical NEs snoop the LLDP packets received from the local Ethernet interface and report to the O-PNCs the extracted information, such as the Chassis ID, the Port ID, System Name TLVs.

Note that the optical NEs do not actively participate in the LLDP packet exchange and does not send any LLDP packets.

4.5.1. Cross-layer link discovery

The MDSC can discover a cross-layer link by matching the plug-id values of the two Ethernet LTPs reported by two adjacent O-PNC and P-PNC: in case LLDP snooping is used, the P-PNC reports the LLDP information sent by the corresponding Ethernet interface on the router while the O-PNC reports the LLDP information received by the corresponding Ethernet interface on the optical NE, e.g., between LTP 5-0 on PE13 and LTP 7-0 on NE11, as shown in Figure 4.
Notes:

=====

(*) Supporting LTP

Legenda:

=====

O  LTP

----> Supporting LTP

<...> Link discovered by the MDSC

{   } LTP Plug-id reported by the PNC

Figure 4 - Cross-layer link discovery

It is worth noting that the discovery of cross-layer links is based only on the LLDP information sent by the Ethernet interfaces of the routers and received by the Ethernet interfaces of the optical NEs, Therefore the MDSC can discover these links also before overlay multi-layer IP links are setup.
4.5.2. Inter-domain IP link discovery

The MDSC can discover an inter-domain Ethernet link which supports an inter-domain IP link, by matching the plug-id values of the two Ethernet LTPs reported by the two adjacent P-PNCs: the two P-PNCs report the LLDP information being sent and being received from the corresponding Ethernet interfaces, e.g., between the Ethernet LTP 3-1 on BR11 and the Ethernet LTP 4-1 on BR21 shown in Figure 5.
Notes:

 =====
(*) Supporting LTP
(1) {BR11,3,BR21,4}
(2) {BR11,3}

Legenda:

 =====
O LTP

----> Supporting LTP
<"..." Link discovered by the MDSC
<"~~" Link inferred by the MDSC
{ } LTP Plug-id reported by the PNC

Figure 5 - Inter-domain Ethernet and IP link discovery

Different information is required to be encoded within the plug-id attribute of the Etherent LTPs to discover cross-layer links and inter-domain Ethernet links.
If the P-PNC does not know a priori whether an Ethernet interface on a router terminates a cross-layer link or an inter-domain Ethernet link, it has to report at the MPI two Ethernet LTPs representing the same Ethernet interface, e.g., both the Ethernet LTP 3-0 and the Ethernet LTP 3-1, supported by LTP 3-0, shown in Figure 5:

- The physical Ethernet LTP is used to represent the physical adjacency between the router Ethernet interface and either the adjacent router Ethernet interface (in case of a single-layer Ethernet link) or the optical NE Ethernet interface (in case of a multi-layer Ethernet link). Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent by the corresponding router Ethernet interface;

- The logical Ethernet LTP, supported by a physical Ethernet LTP, is used to discover the logical adjacency between router Ethernet interfaces, which can be either single-layer or multi-layer. Therefore, this LTP reports, within the plug-id attribute, the LLDP information sent and received by the corresponding router Ethernet interface.

It is worth noting that in case of an inter-domain single-layer Ethernet link, the physical adjacency between the two router Ethernet interfaces cannot be discovered by the MDSC, using the LLDP information reported in the plug-id attributes, as shown in Figure 5. However, the MDSC may infer these links if it knows a priori, using mechanisms which are outside the scope of this document, that inter-domain Ethernet links are always single-layer, e.g., as shown in Figure 5.

The P-PNC can omit reporting the physical Ethernet LTPs when it knows, by mechanisms which are outside the scope of this document, that the corresponding router Ethernet interfaces terminate single-layer inter-domain Ethernet links.

The MDSC can then discover an inter-domain IP link between the two IP LTPs that are supported by the two Ethernet LTPs terminating an inter-domain Ethernet link, discovered as described in section 4.5.2, e.g., between the IP LTP 3-2 on BR21 and the IP LTP 4-2 on BR22, supported respectively by the Ethernet LTP 3-1 on BR11 and by the Ethernet LTP 4-1 on BR21, as shown in Figure 5.

4.6. Multi-layer IP link discovery

A multi-layer intra-domain IP link and its supporting multi-layer intra-domain Ethernet link are discovered by the P-PNC like any other
intra-domain IP and Ethernet links, as described in section 4.3, and reported at the MPI within the SR-TE and Ethernet network topologies, e.g., as shown in Figure 6.
Notes:
=====
(*) Supporting LTP

Legenda:
=========
O  LTP
-----> Supporting LTP or Supporting Link or Underlay tunnel
<==>  Link discovered by the PNC and reported at the MPI
<....> Link discovered by the MDSC
<~~~> Link inferred by the MDSC
x---x Ethernet client signal
X===X Optical tunnel

Figure 6 - Multi-layer intra-domain Ethernet and IP link discovery

The P-PNC does not report any plug-id information on the Ethernet LTPs terminating intra-domain Ethernet links since these links are discovered by the PNC.

In addition, the P-PNC also reports the physical Ethernet LTPs that terminate the cross-layer links supporting the multi-layer intra-domain Ethernet links, e.g., the Ethernet LTP 5-0 on PE13 and the Ethernet LTP 6-0 on BR11, shown in Figure 6.

The MDSC discovers, using the mechanisms described in section 4.5, which Ethernet cross-layer links support the multi-layer intra-domain Ethernet links, e.g. as shown in Figure 6.

The MDSC also discovers, from the information provided by the O-PNC and described in section 4.2, which optical tunnels support the multi-layer intra-domain IP links and therefore the path within the optical network that supports a multi-layer intra-domain IP link, e.g., as shown in Figure 6.

4.6.1. Single-layer intra-domain IP links

It is worth noting that the P-PNC may not be aware of whether an Ethernet interface on the router terminates a multi-layer or a single-layer intra-domain Ethernet link.

In this case, the P-PNC, always reports two Ethernet LTPs for each Ethernet interface on the router, e.g., the Ethernet LTP 1-0 and 1-1 on PE13, shown in Figure 7.
Notes:

=====

(*) Supporting LTP
(1) {PE13,1}
(2) {P16,2}

Legenda:

========

O  LTP
-----> Supporting LTP
<=<<< Link discovered by the PNC and reported at the MPI
<=~~~ Link inferred by the MDSC
{   } LTP Plug-id reported by the PNC

Figure 7 - Single-layer intra-domain Ethernet and IP link discovery

In this case, the MDSC, using the plug-id information reported in the physical Ethernet LTPs, does not discover any cross-layer link being terminated by the corresponding Ethernet interface. The MDSC may infer the physical intra-domain Ethernet link, e.g., between LTP 1-0 on PE13 and LTP 2-0 on P16, as shown in Figure 7, if it knows a
priori, by mechanisms which are outside the scope of this document, that all the Ethernet interfaces on the routers either terminates a cross-layer link or a single-layer intra-domain Ethernet link.

The P-PNC can omit reporting the physical Ethernet LTP if it knows, by mechanisms which are outside the scope of this document, that the intra-domain Ethernet link is single-layer.

4.7. LAG discovery

The P-PNCs can discover the configuration of the LAG groups within its domain and report each intra-domain LAG as an Ethernet bundle link, within the Ethernet topology exposed at the MPI.

This is done bundling multiple single-domain Ethernet links, as shown in Figure 8. For example, the Ethernet bundled link between the Ethernet LTP 5-1 on BR21 and the Ethernet LTP 6-1 on P24, is built from the Ethernet links setup respectively:

- between the Ethernet LTP 1-1 on BR21 and the Ethernet LTP 2-1 on P24; and
- between the Ethernet LTP 3-1 on BR21 and the Ethernet LTP 4-1 on P24.
Figure 8 - LAG

The mechanisms used by the MDSC to discover single-layer and multi-layer intra-domain LAG link is the same (the only difference being whether the bundled links are single-layer or multi-layer).

Instead, the mechanisms used by the MDSC to discover single-layer inter-domain LAG links between two BRs are different and outside the scope of this document since they do not imply any cross-layer coordination between packet and optical domains.

As described in section 4.3, the mechanisms used by the P-PNC to discover the configuration of the LAG groups within its domain, such as LLDP [IEEE 802.1AB], are outside the scope of this document.
However, it is worth noting that according to [IEEE 802.1AB], LLDP can be configured on a LAG group (Aggregated Port) and/or on any number of its LAG members (Aggregation Ports).

If LLDP is enabled on both LAG members and groups, two types of LLDP packets are transmitted by the routers and received by the optical NEs on some cross-layer links: one sent for the LLDP session configured at LAG member (Aggregation Port) level and another one for the LLDP session configured at LAG group (Aggregated Port) level. This could cause some issues when LLDP snooping is used to discover the cross-layer links, as defined in section 4.5.1.

The cross-layer link discovery is based only on the LLDP session configured on the LAG members (Aggregation Ports) to allow discovery of these links independently from the configuration of the underlay optical tunnel or from the LAG group.

To avoid any ambiguity on how the optical NEs can identify which LLDP packets belong to which LLDP session, the P-PNC can disable the LLDP sessions on the LAG groups configured by the MDSC (e.g., the multi-layer single-domain LAG groups configured using the mechanisms described in section 5.2.1), keeping the LLDP sessions on the LAG members enabled.

Another option is to rely on other mechanisms (e.g., the Port type field in the Link Aggregation TLV defined in Annex F of [IEEE 802.1AX]) that allow the optical NE to identify which LLDP packets belong to which LLDP session: the O-PNC can then use only the LLDP information from the LLDP sessions configured on the LAG members to support the cross-layer link discovery mechanisms defined in section 4.5.1.

4.8. L2/L3 VPN network services discovery

TBA

4.9. Inventory discovery

The are no YANG data models in IETF that could be used to report at the MPI the whole inventory information discovered by a PNC.

[RFC8345] had foreseen some work for inventory as an augmentation of the network model, but no YANG data model has been developed so far.

There are also no YANG data models in IETF that could be used to correlate topology information, e.g., a link termination point (LTP),
with inventory information, e.g., the physical port supporting an LTP, if any.

Inventory information through MPI and correlation with topology information is identified as a gap requiring further work and outside of the scope of this draft.

5. Establishment of L2/L3 VPN network services with TE requirements

In this scenario the MDSC needs to setup a multi-domain L2VPN or a multi-domain L3VPN with some SLA requirements.

The MDSC receives the request to setup a L2/L3 VPN network service from the OSS/Orchestration layer (see Appendix A).

The MDSC translates the L2/L3 VPN SLA requirements into TE requirements (e.g., bandwidth, TE metric bounds, SRLG disjointness, nodes/links/domains inclusion/exclusion) and find the TE paths that meet these TE requirements (see section 2.1.1).

For example, considering the L3VPN in Figure 2, the MDSC finds that:

- PE13-P16-PE14 SR-TE path already exists but have not enough bandwidth to support the new L3VPN, as described in section 4.4;
- and that:
  - the IP link(s) between PE13 and P16 has not enough bandwidth to support increasing the bandwidth of that TE path, as described in section 4.3;
  - a new underlay optical tunnel could be setup to increase the bandwidth of the IP link(s) between PE13 and P16 to support increasing the bandwidth of that overlay TE path, as described in section 5.1. The dimensioning of the underlay optical tunnel is decided by the MDSC based on the TE requirements (e.g., the bandwidth) requested by the TE path and on its multi-layer optimization policy, which is an internal MDSC implementation issue;
  - a new multi-domain TE path needs to be setup between PE13 and PE23, e.g., either because existing TE paths between PE13 and PE23 are not able to meet the TE and binding requirements of the L2/L3 VPN service or because there is no TE path between PE13 and PE23.
As described in section 2.1.2, with partial summarization, the MDSC will use the TE topology information provided by the P-PNCs and the results of the path computation requests sent to the O-PNCs, as described in section 5.1, to compute the multi-layer/multi-domain path between PE13 and PE23.

For example, the multi-layer/multi-domain performed by the MDSC could require the setup of:

- a new underlay optical tunnel between PE13 and BR11, supporting a new IP link, as described in section 5.2;
- a new underlay optical tunnel between BR21 and P24 to increase the bandwidth of the IP link(s) between BR21 and P24, as described in section 5.2.

When the setup of the L2/L3 VPN network service requires multi-domain and multi-layer coordination, the MDSC is also responsible for coordinating the network configuration required to realize the request network service across the appropriate optical and packet domains.

The MDSC would therefore request:

- the O-PNC1 to setup a new optical tunnel between the ROADMs connected to PE13 and P16, as described in section 5.2;
- the P-PNC1 to update the configuration of the existing IP link, in case of LAG, or configure a new IP link, in case of ECMP, between PE13 and P16, as described in section 5.2;
- the P-PNC1 to update the bandwidth of the selected TE path between PE13 and PE14, as described in section 5.3.

After that, the MDSC requests P-PNC2 to setup a TE path between BR21 and PE23, with an explicit path (BR21, P24, PE23) to constrain this new TE path to use the new underlay optical tunnel setup between BR21 and P24, as described in section 5.3. The P-PNC2 properly configures the routers within its domain to setup the requested path and returns to the MDSC the information which is needed for multi-domain TE path stitching. For example, in case of inter-domain SR-TE, the P-PNC2, knowing the node and the adjacency SIDs assigned within its domain, can install the proper SR policy, or hierarchical policies, within BR21 and returns to the MDSC the binding SID it has assigned to this policy in BR21.
Then the MDSC requests P-PNC1 to setup a TE path between PE13 and BR11, with an explicit path (PE13, BR11) to constrain this new TE path to use the new underlay optical tunnel setup between PE13 and BR11, specifying also which inter-domain link should be used to send traffic to BR21 and the information to be used for the multi-domain TE path stitching, as described in section 4.4 (e.g., in case of inter-domain SR-TE, the binding SID that has been assigned by P-PNC2 to the corresponding SR policy in BR21). The P-PNC1 properly configures the routers within its domain to setup the requested path and the multi-domain TE path stitching. For example, in case of inter-domain SR-TE, the P-PNC1, knowing also the node and the adjacency SIDs assigned within its domain and the EPE SID assigned by P-PNC1 to the inter-domain link between BR11 and BR21, and the binding SID assigned by P-PNC2, installs the proper policy, or policies, within PE13.

Once the TE paths have been selected and, if needed, setup/modified, the MDSC can request to both P-PNCs to configure the L3VPN and its binding with the selected TE paths using the [RFC9182] and [TSM] YANG data models.

[Editor's Note] Further investigation is needed to understand how the binding between a L3VPN and this new end-to-end SR-TE path can be configured.

5.1. Optical Path Computation

As described in section 2.1.2, the optical path computation is usually performed by the O-PNCs.

When performing multi-layer/multi-domain path computation, the MDSC can delegate the O-PNC for single-domain optical path computation.

As discussed in [PATH-COMPUTE], there are two options to request an O-PNC to perform optical path computation: either via a "compute-only" TE tunnel path, using the generic TE tunnel YANG data model defined in [TE-TUNNEL] or via the path computation RPC defined in [PATH-COMPUTE].

This draft assumes that the path computation RPC is used.

As described in sections 4.1 and 4.5, there is a one-to-one relationship between the router ports, the cross-layer links and the optical TTPs. Therefore, the properties of an optical path between two optical TTPs, as computed by the O-PNC, can be used by the MDSC.
to infer the properties of the multi-layer single-domain IP link between the router ports associated with the two optical TTPs.

There are no YANG data models in IETF that could be used to augment the generic path computation RPC with technology-specific attributes.

Optical technology-specific augmentation for the path computation RPC is identified as a gap requiring further work outside of this draft’s scope.

5.2. Multi-layer IP link Setup

To setup a new multi-layer IP link between two router ports, the MDSC requires the O-PNC to setup an optical tunnel (either a WSON Tunnel or a Flexi-grid Tunnel or an OTN Tunnel) within the optical network between the two TTPs associated, as described in section 5.1, with these two router Ethernet interfaces.

The MDSC also requires the O-PNC to steer the Ethernet client traffic between the two cross-layer links over the optical tunnel using the Ethernet Client Signal Model.

After the optical tunnel has been setup and the client traffic steering configured, the two IP routers can exchange Ethernet packets between themselves, including LLDP messages.

For example, with a reference to Figure 6, the MDSC can request the O-PNC1 to setup an optical tunnel between the TTPs within NE11 and NE14 to steer over this tunnel the Ethernet traffic between LTP (7-0) on NE11 and LTP (8-0) on NE14.

If LLDP [IEEE 802.1AB] or any other discovery mechanisms, which are outside the scope of this document, is used between the adjacency between the two routers’ ports, the P-PNC can automatically discover the underlay multi-layer single-domain Ethernet link being set up by the MDSC and report it to the P-PNC, as described in section 4.6.

Otherwise, if there are no automatic discovery mechanisms, the MDSC can configure this multi-layer single-domain Ethernet link at the MPI of the P-PNC.

The two Ethernet LTPs terminating this multi-layer single-domain Ethernet link are supported by the two underlay Ethernet LTPs terminating the two cross-layer links, e.g., the LTP 5-1 on PE13 and 6-1 on BR11 shown in Figure 6.
After the multi-layer single-domain Ethernet link has been configured by the MDSC or discovered by the P-PNC, the corresponding multi-layer single-domain IP link can also be configured either by the MDSC or by the P-PNC.

This document assumes that this IP link is configured by the P-PNC.

It is worth noting that if LAG is not supported within the domain controlled by the P-PNC, the P-PNC can configure the multi-layer single-domain IP link as soon as the underlay multi-layer single-domain Ethernet link is either discovered by the P-PNC or configured by the MDSC at the MPI. However, if LAG is supported the P-PNC has not enough information to know whether the discovered/configured multi-layer single-domain Ethernet link would be:

1. Used to support a multi-layer single-domain IP link;
2. Used to create a new LAG group;
3. Added to an existing LAG group.

Therefore the P-PNC does not take any further action after a multi-layer single-domain Ethernet link is discovered or configured by the MDSC at the MPI.

The MDSC can request the P-PNC to configure a new multi-layer single-domain IP link, supported by the the just discovered or configured multi-layer single-domain Ethernet link, by creating an IP link within the running datastore of the P-PNC MPI. Only the IP link, IP LTPs and the reference to the supporting multi-layer single-domain Ethernet link are configured by the MDSC. All the other configuration is provided by the P-PNC.

For example, with a reference to Figure 6, the MDSC can request the P-PNC1 to setup a multi-layer single-domain IP Link between IP LTP 5-2 on PE13 and IP LTP 6-2 on BR11 supported by the multi-layer single-domain Ethernet link between ETH LTP 5-1 on PE13 and ETH LTP 6-1 on BR11.

The P-PNC configures the requested multi-layer single-domain IP link and, once finished, reports it to the MDSC within the IP topology exposed at its MPI.
5.2.1. Multi-layer LAG setup

The P-PNC configures a new LAG group between two routers when the MDSC creates at the MPI a new Ethernet bundled link (using the bundled-link container defined in [RFC8795]) bundling the multi-layer single-domain Ethernet link(s) being created, as described above.

It is worth noting that a new LAG group can be created to bundle one or more multi-layer single-domain Ethernet link(s).

For example, with a reference to Figure 8, the MDSC can request the P-PNC2 to setup an Ethernet bundled link between the Ethernet LTP 5-1 on BR21 and the Ethernet LTP 6-1 on P24, bundling the multi-layer single-domain Ethernet link between the Ethernet LTP 1-1 on BR21 and the Ethernet LTP 2-1 on P24.

It is worth noting that the MDSC needs to create also the Ethernet LTPs terminating the Ethernet bundled link.

The MDSC can request the P-PNC to configure a new multi-layer single-domain IP link, supported by the just configured Ethernet bundled link, following the same procedure described in section 5.2 above.

For example, with a reference to Figure 8, the MDSC can request the P-PNC2 to setup a multi-layer single-domain IP Link between IP LTP 5-2 on BR21 and IP LTP 6-2 on P24 supported by the Ethernet bundle link between ETH LTP 5-1 on BR21 and the Ethernet LTP 6-1 on P24.

5.2.2. Multi-layer LAG update

The P-PNC adds new member(s) to an existing LAG group when the MDSC updates at the MPI the configuration of an existing Ethernet bundled link adding the multi-layer single-domain Ethernet link(s) being created, as described above.

For example, with a reference to Figure 8, the MDSC can request the P-PNC2 to add the multi-layer single-domain Ethernet link setup between the Ethernet LTP 3-1 on BR21 and the Ethernet LTP 4-1 on P24 to the existing Ethernet bundle link setup between the Ethernet LTP 5-1 on node BR21 and the Ethernet LTP 6-1 on node P24.

After the LAG configuration has been updated, the P-PNC can also update the bandwidth information of the multi-layer single-domain IP link supported by the updated Ethernet bundled link.
5.2.3. Multi-layer SRLG configuration

[Editor’s Note] Add text about the configuration of multi-layer SRLG information (issue #45).

It is worth noting that the list of SRLGs for a multi-layer IP link can be quite long. Implementation-specific mechanisms can be implemented by the MDSC or by the O-PNC to summarize the SRLGs of an optical tunnel. These mechanisms are implementation-specific and have no impact on the YANG models nor on the interoperability at the MPI, but cares have to be taken to avoid missing information.

5.3. TE Path Setup and Update

This version of the draft assumes that TE path setup and update at the MPI could be done using the generic TE tunnel YANG data model, defined in [TE-TUNNEL], with packet technology-specific augmentations, described in section 3.2.3.

When a new TE path needs to be setup, the MDSC can use the [TE-TUNNEL] model to request the P-PNC to set it up, properly specifying the path constraints, such as the explicit path, to force the P-PNC to setup an TE path that meets the end-to-end TE and binding constraints and uses the optical tunnels setup by the MDSC for the purpose of supporting this new TE path.

The [TE-TUNNEL] model supports requesting the setup of both end-to-end as well as segment TE tunnels (within one domain).

In the latter case, the technology-specific augmentations should allow the configuration of the information needed for multi-domain TE path stitching.

For example, the SR-TE specific augmentations of the [TE-TUNNEL] model should be defined to allow the MDSC to configure the binding SIDs to be used for the multi-domain SR-TE path stitching and to allow the P-PNC to report the binding SID assigned to the segment TE paths. Note that the assigned binding SID should be persistent in case router or P-PNC rebooting.

The MDSC can also use the [TE-TUNNEL] model to request the P-PNC to increase the bandwidth allocated to an existing TE path, and, if needed, also on its reverse TE path. The [TE-TUNNEL] model supports both symmetric and asymmetric bandwidth configuration in the two directions.
[Editor’s Note:] Add some text about the protection options (to further discuss whether to put this text here or in section 4.2.2).

The MDSC also request the P-PNC to configure local protection mechanisms. For example, in case of SR-TE domain, the TI-LFA local protection: the mechanisms to request the configuration TI-LFA local protection for SR-TE paths using the [TE-TUNNEL] are a gap in the current YANG models.

The requested local protection mechanisms within the P-PNC domain are configured by the P-PNC through implementation specific mechanisms which are outside the scope of this document.

The P-PNC takes into account the multi-layer SRLG information, configured by the MDSC as described in section 5.2, when computing the protection configuration (e.g., in case of SR-TE domains, the TI-LFA post-convergence path for multi-layer single-domain IP links).

SR-TE path setup and update (e.g., bandwidth increase) through MPI is identified as a gap requiring further work, which is outside of the scope of this draft.

6. Conclusions

The analysis provided in this document has shown that the IETF YANG models described in 3.2 provides useful support for Packet Optical Integration (POI) scenarios for resource discovery (network topology, service, tunnels and network inventory discovery) as well as for supporting multi-layer/multi-domain L2/L3 VPN network services.

Few gaps have been identified to be addressed by the relevant IETF Working Groups:

- network inventory model: this gap has been identified in section 4.9 and the solution in [NETWORK-INVENTORY] has been proposed to resolve it;

- technology-specific augmentations of the path computation RPC, defined in [PATH-COMPUTE] for optical networks: this gap has been identified in section 5.1 and the solution in [OPTICAL-PATH-COMPUTE] has been proposed to resolve it;

- relationship between a common discovery mechanisms applicable to access links, inter-domain IP links and cross-layer links and the UNI topology discover mechanism defined in [SAP]: this gap has been identified in section 4.3;
o a mechanism applicable to the P-PNC NBI to configure the SR-TE paths. Technology-specific augmentations of TE Tunnel model, defined in [TE-TUNNEL], are foreseen in section 1 of [TE-TUNNEL] but not yet defined: this gap has been identified in section 5.3.

7. Security Considerations

Several security considerations have been identified and will be discussed in future versions of this document.

8. Operational Considerations

Telemetry data, such as collecting lower-layer networking health and consideration of network and service performance from POI domain controllers, may be required. These requirements and capabilities will be discussed in future versions of this document.

9. IANA Considerations

This document requires no IANA actions.

10. References

10.1. Normative References


[Flexi-TOPO] Lopez de Vergara, J. E. et al., "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang, work in progress.


10.2. Informative References


Appendix A. OSS/Orchestration Layer

The OSS/Orchestration layer is a vital part of the architecture framework for a service provider:

- to abstract (through MDSC and PNCs) the underlying transport network complexity to the Business Systems Support layer;

- to coordinate NFV, Transport (e.g. IP, optical and microwave networks), Fixed Acess, Core and Radio domains enabling full automation of end-to-end services to the end customers;

- to enable catalogue-driven service provisioning from external applications (e.g. Customer Portal for Enterprise Business services), orchestrating the design and lifecycle management of these end-to-end transport connectivity services, consuming IP and/or optical transport connectivity services upon request.

As discussed in section 2.1, in this document, the MDSC interfaces with the OSS/Orchestration layer and, therefore, it performs the functions of the Network Orchestrator, defined in [RFC8309].

The OSS/Orchestration layer requests the creation of a network service to the MDSC specifying its end-points (PEs and the interfaces towards the CEs) as well as the network service SLA and then proceeds to configuring accordingly the end-to-end customer service between the CEs in the case of an operator managed service.

A.1. MDSC NBI

As explained in section 2, the OSS/Orchestration layer can request the MDSC to setup L2/L3VPN network services (with or without TE requirements).

Although the OSS/Orchestration layer interface is usually operator-specific, typically it would be using a RESTCONF/YANG interface with a more abstracted version of the MPI YANG data models used for network configuration (e.g. L3NM, L2NM).

Figure 9 shows an example of possible control flow between the OSS/Orchestration layer and the MDSC to instantiate L2/L3 VPN network services, using the YANG data models under the definition in [VN], [L2NM], [RFC9182] and [TSM].

Peruzzini et al. Expires January 10, 2023 [Page 55]
The VN YANG data model, defined in [VN], whose primary focus is the CMI, can also provide VN Service configuration from an orchestrated network service point of view when the L2/L3 VPN network service has TE requirements. However, this model is not used to setup L2/L3 VPN service with no TE requirements.

- It provides the profile of VN in terms of VN members, each of which corresponds to an edge-to-edge link between customer end-points (VNAPs). It also provides the mappings between the VNAPs with the LTPs and the connectivity matrix with the VN member. The associated traffic matrix (e.g., bandwidth, latency, protection level, etc.) of VN member is expressed (i.e., via the TE-topology’s connectivity matrix).

- The model also provides VN-level preference information (e.g., VN member diversity) and VN-level admin-status and operational-status.

- The L2NM and L3NM YANG data models, defined in [L2NM] and [RFC9182], whose primary focus is the MPI, can also be used to provide L2VPN and L3VPN network service configuration from an orchestrated connectivity service point of view.

- The TE & Service Mapping YANG data model [TSM] provides TE-service mapping.
o TE-service mapping provides the mapping between a L2/L3 VPN instance and the corresponding VN instances.

o The TE-service mapping also provides the binding requirements as to how each L2/L3 VPN/VN instance is created concerning the underlay TE tunnels (e.g., whether they require a new and isolated set of TE underlay tunnels or not).

o Site mapping provides the site reference information across L2/L3 VPN Site ID, VN Access Point ID, and the LTP of the access link.
Appendix B. Multi-layer and multi-domain resiliency

B.1. Maintenance Window

Before planned maintenance operation on DWDM network takes place, IP traffic should be moved hitless to another link.

MDSC must reroute IP traffic before the events takes place. It should be possible to lock IP traffic to the protection route until the maintenance event is finished, unless a fault occurs on such path.

B.2. Router port failure

The focus is on client-side protection scheme between IP router and reconfigurable ROADM. Scenario here is to define only one port in the routers and in the ROADM muxponder board at both ends as back-up ports to recover any other port failure on client-side of the ROADM (either on router port side or on muxponder side or on the link between them). When client-side port failure occurs, alarms are raised to MDSC by IP-PNC and O-PNC (port status down, LOS etc.). MDSC checks with OP-PNC(s) that there is no optical failure in the optical layer.

There can be two cases here:

a) LAG was defined between the two end routers. MDSC, after checking that optical layer is fine between the two end ROADMs, triggers the ROADM configuration so that the router back-up port with its associated muxponder port can reuse the OCh that was already in use previously by the failed router port and adds the new link to the LAG on the failure side.

While the ROADM reconfiguration takes place, IP/MPLS traffic is using the reduced bandwidth of the IP link bundle, discarding lower priority traffic if required. Once back-up port has been reconfigured to reuse the existing OCh and new link has been added to the LAG then original Bandwidth is recovered between the end routers.

Note: in this LAG scenario let assume that BFD is running at LAG level so that there is nothing triggered at MPLS level when one of the link member of the LAG fails.
b) If there is no LAG then the scenario is not clear since a router port failure would automatically trigger (through BFD failure) first a sub-50ms protection at MPLS level : FRR (MPLS RSVP-TE case) or TI-LFA (MPLS based SR-TE case) through a protection port. At the same time MDSC, after checking that optical network connection is still fine, would trigger the reconfiguration of the back-up port of the router and of the ROADM muxponder to re-use the same OCh as the one used originally for the failed router port. Once everything has been correctly configured, MDSC Global PCE could suggest to the operator to trigger a possible re-optimization of the back-up MPLS path to go back to the MPLS primary path through the back-up port of the router and the original OCh if overall cost, latency etc. is improved. However, in this scenario, there is a need for protection port PLUS back-up port in the router which does not lead to clear port savings.

Acknowledgments

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

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

Contributors

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

Gabriele Galimberti
Cisco
Email: ggalimbe@cisco.com

Zheng Yanlei
China Unicom
Email: zhengyanlei@chinaunicom.cn
Anton Snitser
Sedona
Email: antons@sedonasys.com

Washington Costa Pereira Correia
TIM Brasil
Email: wcorreia@timbrasil.com.br

Michael Scharf
Hochschule Esslingen - University of Applied Sciences
Email: michael.scharf@hs-esslingen.de

Young Lee
Sung Kyun Kwan University
Email: younglee.tx@gmail.com

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

Paolo Volpato
Huawei
Email: paolo.volpato@huawei.com

Brent Foster
Cisco
Email: brfoster@cisco.com
Authors’ Addresses

Fabio Peruzzini
TIM
Email: fabio.peruzzini@telecomitalia.it

Jean-François Bouquier
Vodafone
Email: jeff.bouquier@vodafone.com

Italo Busi
Huawei
Email: Italo.busi@huawei.com

Daniel King
Old Dog Consulting
Email: daniel@olddog.co.uk

Daniele Ceccarelli
Ericsson
Email: daniele.ceccarelli@ericsson.com
IETF Network Slice Service YANG Model
draft-ietf-teas-ietf-network-slice-nbi-yang-01

Abstract

This document defines a YANG model for the IETF Network Slice service model. The model can be used by a IETF Network Slice customer to manage IETF Network Slice from an IETF Network Slice Controller (NSC).

Status of This Memo

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

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

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

This Internet-Draft will expire on 5 September 2022.

Copyright Notice

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

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

This document defines a YANG [RFC7950] data model for the IETF Network Slice service model.

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-slices], which is used to operate IETF Network Slices during the IETF Network Slice instantiation. This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring.
The IETF Network Slice Controller (NSC) is a logical entity that allows customers to manage IETF network slices. Customers operate on abstract IETF network slices. Details related to the production of slices that fulfill the request are internal to the entity that operates the network. Such details are deployment- and implementation-specific.

The NSC receives requests from its customer-facing interface (e.g., from a management system). This interface carries data objects the IETF network slice user provides, describing the needed IETF network slices in terms of topology, target service level objectives (SLO), and also monitoring and reporting requirements. These requirements are then translated into technology-specific actions that are implemented in the underlying network using a network-facing interface. The details of how the IETF network slices are put into effect are out of scope for this document.

The YANG model discussed in this document describes the requirements of an IETF Network Slice from the point of view of the customer. It is thus classified as customer service model in [RFC8309].

Editorial Note: (To be removed by RFC Editor)

This draft contains several placeholder values that need to be replaced with finalized values at the time of publication. Please apply the following replacements:

* "XXXX" --> the assigned RFC value for this draft both in this draft and in the YANG models under the revision statement.

* The "revision" date in model, in the format XXXX-XX-XX, needs to be updated with the date the draft gets approved.

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:
* client
* configuration data
* state data

This document makes use of the terms defined in [RFC7950].

The tree diagram used in this document follow the notation defined in [RFC8340].

This document also makes use of the terms introduced in the Framework for IETF Network Slices [I-D.ietf-teas-ietf-network-slices].

This document defines the following terms:

* IETF Network Slice Connection (NS-Connection): Refers to connectivity construct defined in [I-D.ietf-teas-ietf-network-slices]. An IETF Network Slice can have one or multiple NS-Connections.

* IETF Network Slice Connection (NS-Connection-group): When an IETF Network Slice has multiple NS-connections. The connections with similar SLO or SLE are treated as one NS-connection group. An IETF Network Slice can have one or multiple NS-Connection-groups.

2.1. Acronyms

The following acronyms are used in the document:

CE Customer Edge
NSC Network Slice Controller
NSE Network Slice Endpoint
MTU Maximum Transmission Unit
PE Provider Edge
SLE Service Level Expectation
SLO Service Level Objective

3. IETF Network Slice Service Model Usage

The intention of the IETF Network Slice service model is to allow the customer to manage IETF Network Slices. In particular, the model allows customers to operate in an abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.
According to the [I-D.ietf-teas-ietf-network-slices] description, IETF Network Slices are applicable to use cases such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV (Network Function Virtualization) connectivity, Data Center Interconnect, and 5G E2E network slice.

As shown in Figure 1, in all these use-cases, the model is used by the higher management system to communicate with NSC for life cycle manage of IETF Network Slices including both enablement and monitoring. For example, in 5G E2E (End-to-end) 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.

4. IETF Network Slice Service Model Overview

As defined in [I-D.ietf-teas-ietf-network-slices], an IETF Network Slice service is specified in terms of a set of endpoints, a set of one or more connectivity constructs (point-to-point (P2P), point-to-multipoint (P2MP), or multipoint-to-multipoint (MP2MP) between subsets of these endpoints, and a set of SLOs and SLEs for each endpoints sending to each connectivity construct. A connection construct is the basic connectivity unit of a network slice, and a slice service may consist of one or more connection constructs. The endpoints are conceptual points that could map to a device, application or a network function. And the specific service requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics, such as security, 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 function). An example of an IETF network slice containing multiple connectivity constructs is shown in Figure 2.
As shown in the example, an IETF network slice may have multiple NSEs. The NSEs are the ingress/egress points where traffic enters/exits the IETF network slice. As the edge of the IETF network slice, the NSEs also delimit a topological network portion within which the committed SLOs apply.

When an NSC receives a message via its customer-facing interface for creation/modification of an IETF network slice, it uses the provided NSEs to retrieve the corresponding service demarcation link or slice provider edge node" (e.g., PE). The NSC further maps them to the appropriate service/tunnel/path endpoints in the underlying network. It then uses services/tunnels/paths to realize the IETF network slice.
The 'ietf-network-slice' module uses two main data 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 a provider 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. And list "ns-connection" is used to abstract connections or connectivity constructs between NSEs.

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 network-slices
  +--rw ns-slo-sle-templates
    +--rw ns-slo-sle-template* [id]
      +--rw id string
      +--rw template-description? string
  +--rw network-slice* [ns-id]
    +--rw ns-id string
    +--rw ns-description? string
    +--rw ns-tags
      +--rw ns-tag* [index]
        +--rw index uint32
        +--rw ns-tag-type? identityref
        +--rw ns-tag-value? string
    +--:(standard)
      +--rw slo-sle-template? leafref
    +--:(custom)
      +--rw slo-sle-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 bound? uint64
        +--rw security* identityref
        +--rw isolation? identityref
        +--rw max-occupancy-level? uint8
        +--rw mtu uint16
        +--rw steering-constraints
          +--rw path-constraints
---rw service-function

++-rw status
    +++-rw admin-enabled?  boolean
    +++-ro oper-status?    operational-type

++-rw ns-endpoints
    +++-rw ns-endpoint* [ep-id]
        +++-rw ep-id               string
        +++-rw ep-description?      string

    +++-rw location
        |    +++-rw altitude?     int64
        |    +++-rw latitude?     decimal64
        |    +++-rw longitude?    decimal64

        +++-rw node-id?           string
        +++-rw ep-ip?             inet:ip-address

    +++-rw ns-match-criteria
        +++-rw ns-match-criterion* [index]
            +++-rw index           uint32
            |    +++-rw match-type?  identityref
            |    |    +++-rw values* [index]
            |    |    |    +++-rw index    uint8
            |    |    |    +++-rw value?   string
            |    |    +++-rw target-ns-connection-group-id? leafref

    +++-rw ep-peering
        +++-rw protocol* [protocol-type]
            +++-rw protocol-type  identityref

    +++-rw ep-network-access-points
        +++-rw ep-network-access-point* [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-address? inet:ip-address
            +++-rw network-access-tp-ip-prefix-length?  uint8
            +++-rw network-access-qos-policy-name? string
            +++-rw mtu                     uint16
        +++-rw network-access-tags
```yang
++-rw network-access-tag* [index]
    +--rw index          uint32
    +--rw network-access-tag-type?
        identityref
    +--rw network-access-tag-value?  string

++-rw ns-match-criteria
    +--rw ns-match-criterion* [index]
        +--rw index
        +--rw match-type?
            identityref
        +--rw values* [index]
            +--rw index     uint8
            +--rw value?    string
        +--rw target-ns-connection-group-id?  leafref

++-rw ep-peering
    +--rw protocol* [protocol-type]
        +--rw protocol-type  identityref
    +--rw attribute* [index]
        +--rw index
        +--rw attribute-description?  string
        +--rw value?  string

++-rw incoming-rate-limits
    +--rw cir?   uint64
    +--rw cbs?   uint64
    +--rw eir?   uint64
    +--rw ebs?   uint64
    +--rw pir?   uint64
    +--rw pbs?   uint64

++-rw outgoing-rate-limits
    +--rw cir?   uint64
    +--rw cbs?   uint64
    +--rw eir?   uint64
    +--rw ebs?   uint64
    +--rw pir?   uint64
    +--rw pbs?   uint64
```

### Figure 3

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 and SLE templates. When creation of IETF Network slice, the SLO and SLE policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the upper layer management system:
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": Gives some description of an IETF Network Slice service.

* "status": Is used to show the operative and administrative status of the IETF Network Slice, and can be used as indicator to detect network slice anomalies.

* "ns-tags": It is a mean to correlate the higher level "Customer higher level operation system" and IETF network slices. It might be used by IETF network slice operator to provide additional information to the IETF Network Slice Controller (NSC) during the automation of the IETF network slices. E.g. adding tag with "customer-name" when multiple actual customers use a same network slice. Another use-case for "ns-tag" might be for Operator to provide additional attributes to NSC which might be used during
the realization of IETF network slices such as type of services (e.g., L2 or L3). These additional attributes can also be used by the NSC for various use-cases such as monitoring and assurance of the IETF network slices where NSC can notify the higher system by issuing the notifications. Note that all these attributes are OPTIONAL but might be useful for some use-cases.

* "ns-slo-sle-policy": Defines SLO and SLE policies for the "ietf-network-slice". More description are provided in Section 6.2

* "ns-endpoint": 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.

* "ns-connection-groups": Abstracts the connections between NSEs.

6.1. IETF Network Slice Connectivity

Based on the customer’s traffic requirements, an IETF Network Slice connectivity type could be point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-point (MP2P), multipoint-to-multipoint (MP2MP) or a combination of these types.

[I-D.ietf-teas-ietf-network-slices] defines the basic connectivity construct for a network slice, and the connectivity construct may have different SLO and SLE requirements. "ns-connection" represents this connectivity construct, and "ns-slo-sle-policy" under it represents the per-connection SLO and SLE requirements.

Apart from the per-connection SLO and SLE, slice traffic is usually managed by combining similar types of traffic. For example, some connections for video services require high bandwidth, and some connections for voice over IP request low latency and reliability. "ns-connect-group" is thus defined to treat each type as a class with per-connection-group SLO and SLE.

6.2. IETF Network Slice SLO and SLE Policy

As defined in [I-D.ietf-teas-ietf-network-slices], the SLO and SLE policy of an IETF Network Slice defines some common attributes.

"ns-slo-sle-policy" is used to represent specific SLO and SLE policies. During the creation of an IETF Network Slice, the policy can be specified either by a standard SLO and SLO template or a customized SLO and SLE policy.
The policy can apply to per-network slice, per-connection group "ns-connection group", or per-connection "ns-connection".

The container "ns-metric-bounds" supports all the variations and combinations of NS SLOs, 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 supports the following options:

- "ns-slo-one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is unidirectional.

- "ns-slo-two-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two NSE. And the bandwidth is bidirectional.

- "network-slice-slo-one-way-latency": Indicates the maximum one-way latency between two NSE.

- "network-slice-slo-two-way-latency": Indicates the maximum round-trip latency between two NSE.

- "ns-slo-one-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one-way latency between sequential packets in a flow.

- "ns-slo-two-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

- "ns-slo-one-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

- "ns-slo-two-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

- "ns-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.

The following common SLEs are defined:
"mtu": Refers to the service MTU, which is the maximum PDU size that the customer may use.

"security": Includes the request for encryption or other security techniques to traffic flowing between the two NS endpoints.

"isolation": Specifies the isolation level that a customer expects, including dedicated, shared, or other level.

max-occupancy-level: Specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume.

"steering-constraints": Specifies the constraints how the provider routes traffic for the IETF Network Slice service.

The following shows an example where a network slice policy can be configured:

```json
{
    "ietf-network-slices": {
        "ietf-network-slice": {
            "slo-policy": {
                "policy-description": "video-service-policy",
                "ns-metric-bounds": {
                    "ns-metric-bound": [
                        {
                            "metric-type": "ns-slo-one-way-bandwidth",
                            "metric-unit": "mbps",
                            "bound": "1000"
                        },
                        {
                            "metric-type": "ns-slo-availability",
                            "bound": "99.9%"
                        }
                    ]
                }
            }
        }
    }
}
```

6.3. IETF Network Slice Endpoint (NSE)

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".
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": Indicates NSE location information that facilities NSC easy identification of a NSE.

* "node-id": The NSE node information facilities NSC with easy identification of a NSE.

* "ep-ip": The NSE IP information facilities NSC with easy identification of a NSE.

* "ns-match-criteria": Defines matching policies for network slice traffic to apply on a given NSE.

* "ep-network-access-points": Specifies the list of the interfaces attached to an edge device of the IETF Network Slice by which the customer traffic is received. This is an optional NSE attribute. When a NSE has multiple interfaces attached and the NSC needs NSE interface-specific attributes, each "ep-network-access-point" can specify attributes such as interface specific IP address, MTU, etc.

* "incoming-rate-limits" and "outgoing-rate-limits": Set the 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. If an NSE has multiple AC, the "rate limit" of "ep-network-access-point" can be set to an AC specific value, but the rate cannot exceed the "rate limit" of the NSE. If a NSE only contains a single AC, then the "rate-limit" of "ep-network-access-point" is the same with the NSE "rate-limit". The definition refers to [RFC7640].

* "ep-peering": Specifies the protocol for a NSE for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.

* "status": Enables the control of the operative and administrative status of the NSE, can be used as indicator to detect NSE anomalies.
NSE defines 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 with PE parameters example: As shown in Figure 4, customer 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-points' (Two PE interfaces) to map the interfaces and corresponding services/tunnels/paths.
Legend:
O: Representation of the IETF network slice endpoints (NSE)
+: Mapping of NES to PE or CE-PE interfaces
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

NSE1
(With PE1 parameters)
NSE2
(with PE2 parameters)

Customer
Edge 1
Provider
Edge 1

Provider
Edge 2
Customer
Edge 2

Legend:
O: Representation of the IETF network slice endpoints (NSE)
+: Mapping of NES to PE or CE-PE interfaces
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 with CE parameters example: As shown in Figure 5, customer 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 controller (NSC) uses 'node-id' (CE device ID), 'ep-ip' (CE tunnel endpoint IP), 'network-slice-match-criteria' (VLAN interface), 'ep-network-access-points' (Two nexthop interfaces) to retrieve the corresponding CEs, ACs, or PEs, and further map to services/tunnels/paths.
Internet-Draft      Network Slice Service YANG Model          March 2022

Legend:
O: Representation of the IETF network slice endpoints (NSE)
+: Mapping of NSE to CE or CE-PE interfaces
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

Note: The model needs to be optimized for better extension of other protocols or AC technologies.

7. IETF Network Slice Monitoring

An IETF Network Slice is a connectivity with specific SLO characteristics, including bandwidth, latency, etc. The connectivity is a combination of logical unidirectional connections, represented by 'ns-connection'.

This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

* Per NS connection: specified in 'ns-connection-monitoring' under the "ns-connection".
* Per NS Endpoint: specified in 'ep-monitoring' under the "ns-endpoint".
* Per NS connection group: specified in ’ns-connection-monitoring’ under the "ns-connection-group".

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-connection’, ’ns-endpoint’ or "ns-connection-group", so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the customer management system could initiate modification based on the IETF Network Slice running status.

Note: More critical events affecting service delivery need to be added.

8. IETF Network Slice Service Module

The "ietf-network-slice" module uses types defined in [RFC6991] and [RFC8776], and [RFC7640].

<CODE BEGINS> file "ietf-network-slice@2022-03-04.yang"
module ietf-network-slice {
  yang-version 1.1;
  prefix ietf-ns;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Types.";
  }
  import ietf-te-types {
    prefix te-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }
  import ietf-te-packet-types {
    prefix te-packet-types;
    reference
      "RFC 8776: Common YANG Data Types for Traffic Engineering.";
  }

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

description
"This module contains a YANG module for the IETF Network Slice.

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

Redistribution and use in source and binary forms, with or
without modification, is permitted pursuant to, and subject
to the license terms contained in, the Revised BSD License
set forth in Section 4.c of the IETF Trust’s Legal Provisions
Relating to IETF Documents

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

revision 2022-03-04 {
    description
        "initial version.";
    reference
        "RFC XXXX: A Yang Data Model for IETF Network Slice Operation";
}

/* Features */
/* Identities */

identity ns-tag-type {
    description
        "Base identity for IETF Network Slice tag type.";
}

identity ns-tag-customer {
    base ns-tag-type;
    description
        "The IETF Network Slice customer ID tag type.";
}
identity ns-tag-service {
    base ns-tag-type;
    description
        "The IETF Network Slice service tag type.";
}

identity ns-tag-opaque {
    base ns-tag-type;
    description
        "The IETF Network Slice opaque tag type.";
}

identity network-access-tag-type {
    description
        "Base identity for the network access tag type.";
}

identity network-access-tag-vlan-id {
    base network-access-tag-type;
    description
        "The network access interface VLAN ID tag type.";
}

identity network-access-tag-ip-mask {
    base network-access-tag-type;
    description
        "The network access tag IP mask.";
}

identity network-access-tag-opaque {
    base network-access-tag-type;
    description
        "The network access opaque tag type.";
}

identity ns-isolation-type {
    description
        "Base identity for IETF Network slice isolation level.";
}

identity ns-isolation-shared {
    base ns-isolation-type;
    description
        "Shared resources (e.g. queues) are associated with the Network Slice traffic. Hence, the IETF network slice traffic can be impacted by effects of other services traffic sharing";
the same resources."
}

identity ns-isolation-dedicated {
    base ns-isolation-type;
    description
        "Dedicated resources (e.g. queues) are associated with the
Network Slice traffic. Hence, the IETF network slice traffic
is isolated from other services traffic sharing the same
resources.";
}

identity ns-security-type {
    description
        "Base identity for IETF Network security level.";
}

identity ns-security-authenticate {
    base ns-security-type;
    description
        "IETF Network Slice requires authentication.";
}

identity ns-security-integrity {
    base ns-security-type;
    description
        "IETF Network Slice requires data integrity.";
}

identity ns-security-encryption {
    base ns-security-type;
    description
        "IETF Network Slice requires data encryption.";
}

identity ns-connectivity-type {
    description
        "Base identity for IETF Network Slice connectivity.";
}

identity point-to-point {
    base ns-connectivity-type;
    description
        "Identity for point-to-point IETF Network Slice connectivity.";
}

identity point-to-multipoint {
    base ns-connectivity-type;
description
 "Identity for point-to-multipoint IETF Network Slice connectivity.";
"
}

identity multipoint-to-multipoint {
 base ns-connectivity-type;
 description
 "Identity for multipoint-to-multipoint IETF Network Slice connectivity.";
"
}

identity any-to-any {
 base ns-connectivity-type;
 description
 "Identity for any-to-any IETF Network Slice connectivity.";
"
}

identity hub-spoke {
 base ns-connectivity-type;
 description
 "Identity for Hub-and-Spoke IETF Network Slice connectivity.";
"
}

identity custom {
 base ns-connectivity-type;
 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 ns-slo-metric-type {
    description
        "Base identity for IETF Network Slice SLO metric type.";
}

identity ns-slo-one-way-bandwidth {
    base ns-slo-metric-type;
    description
        "SLO bandwidth metric. Minimum guaranteed bandwidth between
         two endpoints at any time and is measured unidirectionally.";
}

identity ns-slo-two-way-bandwidth {
    base ns-slo-metric-type;
    description
        "SLO bandwidth metric. Minimum guaranteed bandwidth between
         two endpoints at any time.";
}

identity ns-slo-shared-bandwidth {
    base ns-slo-metric-type;
    description
        "The shared SLO bandwidth bound. It is the limit on the
         bandwidth that can be shared amongst a group of connections
         of an IETF Network Slice.";
}

identity ns-slo-one-way-delay {
    base ns-slo-metric-type;
    description
        "SLO one-way-delay is the upper bound of network delay when
         transmitting between two endpoints. The metric is defined in
         RFC7679.";
}

identity ns-slo-two-way-delay {
    base ns-slo-metric-type;
    description
        "SLO two-way delay is the upper bound of network delay when
         transmitting between two endpoints. The metric is defined in
         RFC2681.";
}
identity ns-slo-one-way-delay-variation {
    base ns-slo-metric-type;
    description
        "SLO one-way delay variation is defined by RFC3393, is the difference in the one-way delay between sequential packets between two endpoints.";
}

identity ns-slo-two-way-delay-variation {
    base ns-slo-metric-type;
    description
        "SLO two-way delay variation is defined by RFC5481, is the difference in the round-trip delay between sequential packets between two endpoints.";
}

identity ns-slo-one-way-packet-loss {
    base ns-slo-metric-type;
    description
        "SLO loss metric. The ratio of packets dropped to packets transmitted between two endpoints in one-way over a period of time as specified in RFC7680.";
}

identity ns-slo-two-way-packet-loss {
    base ns-slo-metric-type;
    description
        "SLO loss metric. The ratio of packets dropped to packets transmitted between two endpoints in two-way over a period of time as specified in RFC7680.";
}

identity ns-slo-availability {
    base ns-slo-metric-type;
    description
        "SLO availability level.";
}

identity ns-match-type {
    description
        "Base identity for IETF Network Slice traffic match type.";
}

identity ns-phy-interface-match {
    base ns-match-type;
    description
        "Use the physical interface as match criteria for the IETF Network Slice traffic.";
}
identity ns-vlan-match {
    base ns-match-type;
    description
        "Use the VLAN ID as match criteria for the IETF Network Slice traffic.";
}

identity ns-label-match {
    base ns-match-type;
    description
        "Use the MPLS label as match criteria for the IETF Network Slice traffic.";
}

identity peering-protocol-type {
    description
        "Base identity for NSE peering protocol type.";
}

identity peering-protocol-bgp {
    base peering-protocol-type;
    description
        "Use BGP as protocol for NSE peering with customer device.";
}

identity peering-static-routing {
    base peering-protocol-type;
    description
        "Use static routing for NSE peering with customer device.";
}

/*
 * 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 measurements monitoring type.";
    }
  }
  description "An enumerated type for monitoring on a IETF Network Slice connection.";
}

/* Groupings */
grouping status-params {
  description "A grouping used to join operational and administrative status.";
  container status {
    description "A container for the administrative and operational state.";
    leaf admin-enabled {
      type boolean;
      description "The administrative status.";
    }
    leaf oper-status {
      type operational-type;
      config false;
      description "The operational status.";
    }
  }
}

grouping ns-match-criteria {
  description "A grouping for the IETF Network Slice match definition.";
  container ns-match-criteria {
    description "Describes the IETF Network Slice match criteria.";
    list ns-match-criterion {
      key "index";
      description "List of the IETF Network Slice traffic match criteria.";
    }
  }
}
leaf index {
    type uint32;
    description
        "The entry index.";
}

leaf match-type {
    type identityref {
        base ns-match-type;
    }
    description
        "Identifies an entry in the list of the IETF Network Slice match criteria.";
}

list values {
    key "index";
    description
        "List of match criteria values.";
    leaf index {
        type uint8;
        description
            "Index of an entry in the list.";
    }
    leaf value {
        type string;
        description
            "Describes the IETF Network Slice match criteria, e.g. IP address, VLAN, etc.";
    }
}

leaf target-ns-connection-group-id {
    type leafref {
        path "/network-slices/network-slice"
        + "/ns-connection-groups/ns-connection-group"
        + "/ns-connection-group-id";
    }
    description
        "reference to a Network Slice connection group.";
}

}

grouping ns-sles {
    description
        "Indirectly Measurable Objectives of a IETF Network Slice.";
    leaf-list security {
        type identityref {
            base ns-security;
        }
        description
            "Indirectly Measurable Objectives of Security Constraints.";
    }
    leaf-list ip-traffic-shape {
        type identityref {
            base ns-traffic-shape;
        }
        description
            "Indirectly Measurable Objectives of Traffic Shape.";
    }
    leaf-list buffer-space {
        type identityref {
            base ns-buffer-space;
        }
        description
            "Indirectly Measurable Objectives of Buffer Space.";
    }
    leaf-list latency {
        type identityref {
            base ns-latency;
        }
        description
            "Indirectly Measurable Objectives of Latency.";
    }
    leaf-list capacity {
        type identityref {
            base ns-capacity;
        }
        description
            "Indirectly Measurable Objectives of Capacity.";
    }
    leaf-list availability {
        type identityref {
            base ns-availability;
        }
        description
            "Indirectly Measurable Objectives of Availability.";
    }
    leaf-list throughput {
        type identityref {
            base ns-throughput;
        }
        description
            "Indirectly Measurable Objectives of Throughput.";
    }
    leaf-list jitter {
        type identityref {
            base ns-jitter;
        }
        description
            "Indirectly Measurable Objectives of Jitter.";
    }
    leaf-list bandwidth {
        type identityref {
            base ns-bandwidth;
        }
        description
            "Indirectly Measurable Objectives of Bandwidth.";
    }
    leaf-list reliability {
        type identityref {
            base ns-reliability;
        }
        description
            "Indirectly Measurable Objectives of Reliability.";
    }
    leaf-list security {
        type identityref {
            base ns-security;
        }
        description
            "Indirectly Measurable Objectives of Security Constraints.";
    }
    leaf-list ip-traffic-shape {
        type identityref {
            base ns-traffic-shape;
        }
        description
            "Indirectly Measurable Objectives of Traffic Shape.";
    }
    leaf-list buffer-space {
        type identityref {
            base ns-buffer-space;
        }
        description
            "Indirectly Measurable Objectives of Buffer Space.";
    }
    leaf-list latency {
        type identityref {
            base ns-latency;
        }
        description
            "Indirectly Measurable Objectives of Latency.";
    }
    leaf-list capacity {
        type identityref {
            base ns-capacity;
        }
        description
            "Indirectly Measurable Objectives of Capacity.";
    }
    leaf-list availability {
        type identityref {
            base ns-availability;
        }
        description
            "Indirectly Measurable Objectives of Availability.";
    }
    leaf-list throughput {
        type identityref {
            base ns-throughput;
        }
        description
            "Indirectly Measurable Objectives of Throughput.";
    }
    leaf-list jitter {
        type identityref {
            base ns-jitter;
        }
        description
            "Indirectly Measurable Objectives of Jitter.";
    }
    leaf-list bandwidth {
        type identityref {
            base ns-bandwidth;
        }
        description
            "Indirectly Measurable Objectives of Bandwidth.";
    }
    leaf-list reliability {
        type identityref {
            base ns-reliability;
        }
        description
            "Indirectly Measurable Objectives of Reliability.";
    }
}

base ns-security-type;
}
description
 "The IETF Network Slice security SLE(s)";
}
leaf isolation {
 type identityref {
   base ns-isolation-type;
 }
default "ns-isolation-shared";
description
 "The IETF Network Slice isolation SLE requirement.";
}
leaf max-occupancy-level {
 type uint8 {
   range "1..100";
 }
description
 "The maximal occupancy level specifies the number of flows to
be admitted.";
}
leaf mtu {
 type uint16;
 units "bytes";
 mandatory true;
description
 "The MTU specifies the maximum length in octets of data
packets that can be transmitted by the NS. The value needs
to be less than or equal to the minimum MTU value of
all 'ep-network-access-points' in the NSEs of the NS.";
}
container steering-constraints {
 description
 "Container for the policy of steering constraints
applicable to IETF Network Slice.";
container path-constraints {
 description
 "Container for the policy of path constraints
applicable to IETF Network Slice.";
}
container service-function {
 description
 "Container for the policy of service function
applicable to IETF Network Slice.";
}
}
grouping ns-metric-bounds {
  description
    "IETF Network Slice metric bounds grouping."
  container ns-metric-bounds {
    description
      "IETF Network Slice metric bounds container."
    list ns-metric-bound {
      key "metric-type";
      description
        "List of IETF Network Slice metric bounds."
      leaf metric-type {
        type identityref {
          base ns-slo-metric-type;
        }
        description
          "Identifies an entry in the list of metric type
            bounds for the IETF 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 bound {
        type uint64;
        default "0";
        description
          "The Bound on the Network Slice connection metric. A
            zero indicate an unbounded upper limit for the
            specific metric-type.";
      }
    }
  }
}

grouping ep-peering {
  description
    "A grouping for the IETF Network Slice Endpoint peering."
  container ep-peering {
    description
      "Describes NSE peering attributes."
    }
  }
}
list protocol {
  key "protocol-type";
  description "List of the NSE peering protocol.";
  leaf protocol-type {
    type identityref {
      base peering-protocol-type;
    }
    description "Identifies an entry in the list of NSE peering protocol type.";
  }
  list attribute {
    key "index";
    description "List of protocol attribute.";
    leaf index {
      type uint8;
      description "Index of an entry in the list.";
    }
    leaf attribute-description {
      type string;
      description "The description of the attribute.";
    }
    leaf value {
      type string;
      description "Describes the value of protocol attribute, e.g. nexthop address, peer address, etc.";
    }
  }
}

grouping ep-network-access-points {
  description "Grouping for the endpoint network access definition.";
  container ep-network-access-points {
    description "List of network access points.";
    list ep-network-access-point {
      key "network-access-id";
      description "The IETF Network Slice network access points related parameters.";
    }
  }
}
leaf network-access-id {
  type string;
  description  
    "Uniquely identifier a network access point.";
}
leaf network-access-description {
  type string;
  description  
    "The network access point description.";
}
leaf network-access-node-id {
  type string;
  description  
    "The network access point node ID in the case of
    multi-homing.";
}
leaf network-access-tp-id {
  type string;
  description  
    "The termination port ID of the EP network access
    point.";
}
leaf network-access-tp-ip-address {
  type inet:ip-address;
  description  
    "The IP address of the EP network access point.";
}
leaf network-access-tp-ip-prefix-length {
  type uint8;
  description  
    "The subnet prefix length expressed in bits.";
}
leaf network-access-qos-policy-name {
  type string;
  description  
    "The name of the QoS policy that is applied to the
    network access point. The name can reference a QoS
    profile that is pre-provisioned on the device.";
}
leaf mtu {
  type uint16;
  units "bytes";
  mandatory true;
  description  
    "Maximum size in octets of a data packet that
    can traverse a NSE network access point.";
}
container network-access-tags {

description
 "Container for the network access tags.";

list network-access-tag {
  key "index";
  description
   "The network access point tags list.";
  leaf index {
    type uint32;
    description
     "The entry index.";
  }
  leaf network-access-tag-type {
    type identityref {
      base network-access-tag-type;
    }
    description
     "The network access point tag type.";
  }
  leaf network-access-tag-value {
    type string;
    description
     "The network access point tag value.";
  }
}

/* Per ep-network-access-point rate limits */
uses ns-match-criteria;
uses ep-peering;
uses ns-rate-limit;

grouping ep-monitoring-metrics {
  description
  "Grouping for the NS endpoint monitoring metrics.";
  container ep-monitoring {
    config false;
    description
     "Container for NS endpoint monitoring metrics.";
    leaf incoming-utilized-bandwidth {
      type te-types:te-bandwidth;
      description
       "Incoming bandwidth utilization at an endpoint.";
    }
    leaf incoming-bw-utilization {
      type decimal64 {
        fraction-digits 5;
      }
    }
  }
}

leaf outgoing-utilized-bandwidth {
  type te-types:te-bandwidth;
  description
  "Outgoing bandwidth utilization at an 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 ns-connection-monitoring-metrics {
  description
  "Grouping for NS connection monitoring metrics.";
  uses te-packet-types:one-way-performance-metrics-packet;
  uses te-packet-types:two-way-performance-metrics-packet;
}

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 bw-rate-limits {
  description
  "Bandwidth rate limits grouping.";
  reference
  "RFC 7640: Traffic Management Benchmarking";
  leaf cir {
    type uint64;
    units "bps";
    description
    "Committed Information Rate. The maximum number of bits
    that a port can receive or send during one-second over an
    interface.";
  }
  leaf cbs {
    type uint64;
    units "bytes";
    description
    "Committed Burst Size. CBS controls the bursty nature
    of the traffic. Traffic that does not use the configured
    CIR accumulates credits until the credits reach the
    configured CBS.";
  }
  leaf eir {
    type uint64;
    units "bps";
    description
    "Excess Information Rate, i.e., excess frame delivery
    allowed not subject to SLA. The traffic rate can be
limited by EIR.

leaf ebs {
  type uint64;
  units "bytes";
  description "Excess Burst Size. The bandwidth available for burst traffic from the EBS is subject to the amount of bandwidth that is accumulated during periods when traffic allocated by the EIR policy is not used."
}

leaf pir {
  type uint64;
  units "bps";
  description "Peak Information Rate, i.e., maximum frame delivery allowed. It is equal to or less than sum of CIR and EIR."
}

leaf pbs {
  type uint64;
  units "bytes";
  description "Peak Burst Size."
}

grouping ns-rate-limit {
  description "The rate limits grouping."
  container incoming-rate-limits {
    description "Container for the asymmetric traffic control."
    uses bw-rate-limits;
  }
  container outgoing-rate-limits {
    description "The rate-limit imposed on outgoing traffic."
    uses bw-rate-limits;
  }
}

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
    "Give more description of the Network Slice endpoint."
}

uses geolocation-container;
leaf node-id {
  type string;
  description
    "Uniquely identifies an edge node within the IETF slice network."
}

leaf ep-ip {
  type inet:ip-address;
  description
    "The IP address of the endpoint."
}

uses ns-match-criteria;
uses ep-peering;
uses ep-network-access-points;
uses ns-rate-limit;
/*/ Per NSE rate limits */
uses status-params;
uses ep-monitoring-metrics;
}

//ns-endpoint

grouping ns-connection {
  description
    "The network slice connection grouping.";
list ns-connection {
  key "ns-connection-id";
  description
    "List of Network Slice connections.";
leaf ns-connection-id {
  type uint32;
  description
    "The Network Slice connection identifier."
}
leaf ns-connectivity-type {
  type identityref {
    base ns-connectivity-type;
  }
  default "point-to-point";
  description

"Network Slice connection construct type.";
}
leaf-list src-nse {
  type leafref {
    path "/network-slices/network-slice" + "/ns-endpoints/ns-endpoint/ep-id";
  }
  description
    "reference to source Network Slice endpoint.";
}
leaf-list dest-nse {
  type leafref {
    path "/network-slices/network-slice" + "/ns-endpoints/ns-endpoint/ep-id";
  }
  description
    "reference to source Network Slice endpoint.";
}
uses ns-slo-sle-policy;
/* Per connection ns-slo-sle-policy overrides * the per network slice ns-slo-sle-policy. */
container ns-connection-monitoring {
  config false;
  description
    "SLO status Per NS connection.";
  uses ns-connection-monitoring-metrics;
}
}
}
//ns-connection

grouping ns-connection-group {
  description
    "The Network Slice connection group is described in this container.";
  leaf ns-connection-group-id {
    type string;
    description
      "The Network Slice connection group identifier.";
  }
  uses ns-slo-sle-policy;
  uses ns-connection;
  /* Per connection ns-slo-sle-policy overrides * the per network slice ns-slo-sle-policy. */
  container ns-connection-group-monitoring {
config false;
description
"SLO status Per NS connection."
uses ns-connection-monitoring-metrics;
}
}

//ns-connection-group

grouping slice-template {
description
"Grouping for slice-templates.";
container ns-slo-sle-templates {
description
"Contains a set of network slice templates to reference in the IETF network slice.";
list ns-slo-sle-template {
key "id";
leaf id {
type string;
description
"Identification of the Service Level Objective (SLO) and Service Level Expectation (SLE) template to be used. Local administration meaning.";
}
leaf template-description {
type string;
description
"Description of the SLO &amp; SLE policy template.";
}
description
"List for SLO and SLE template identifiers.";
}
}

/* Configuration data nodes */
grouping ns-slo-sle-policy {
description
"Network Slice policy grouping.";
choice ns-slo-sle-policy {
description
"Choice for SLO and SLE policy template. Can be standard template or customized template.";
case standard {
description
"Standard SLO template.";
}
leaf slo-sle-template {
  type leafref {
    path "/network-slices" + "/ns-slo-sle-templates/ns-slo-sle-template/id";
  }
  description
    "Standard SLO and SLE template to be used.";
}

case custom {
  description
    "Customized SLO template.";
  container slo-sle-policy {
    description
      "Contains the SLO policy.";
    leaf policy-description {
      type string;
      description
        "Description of the SLO policy.";
    }
    uses ns-metric-bounds;
    uses ns-sles;
  }
}
}

container network-slices {
  description
    "Contains a list of IETF network slice";
  uses slice-template;
  list network-slice {
    key "ns-id";
    description
      "A network-slice is identified by a ns-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.";
    }
    container ns-tags {
      description
        "Container for the list of IETF Network Slice tags.";
    }
  }
}
list ns-tag {
    key "index";
    description
        "IETF Network Slice tag list.";
    leaf index {
        type uint32;
        description
            "The entry index.";
    }
    leaf ns-tag-type {
        type identityref {
            base ns-tag-type;
        }
        description
            "The IETF Network Slice tag type.";
    }
    leaf ns-tag-value {
        type string;
        description
            "The IETF Network Slice tag value.";
    }
}

uses ns-slo-sle-policy;
uses status-params;
container ns-endpoints {
    description
        "NS Endpoints.";
    list ns-endpoint {
        key "ep-id";
        uses endpoint;
        description
            "List of endpoints in this slice.";
    }
}

container ns-connection-groups {
    description
        "Contains NS connections group.";
    list ns-connection-group {
        key "ns-connection-group-id";
        description
            "List of Network Slice connections.";
        uses ns-connection-group;
    }
}

//ietf-network-slice list
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/network-slices/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:

    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].
11. Acknowledgments

The authors wish to thank Mohamed Boucadair, John Mullooly, Kenichi Ogaki, Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, and many others for their helpful comments and suggestions.

12. Contributors

The following authors contributed significantly to this document:

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

13. References

13.1. Normative References


13.2. Informative References

[I-D.geng-teas-network-slice-mapping]

[I-D.ietf-opsawg-vpn-common]

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

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

[I-D.liu-teas-transport-network-slice-yang]

Appendix A. IETF Network Slice Service Model Usage Example

The following example describes a simplified service configuration of two IETF Network slice instances:

* IETF Network Slice 1 on PE1, PE2, and PE3, with two NS-connection-groups

```
+-----+ VLAN100
    |       +-----+       |
    |   DU1 |       |  PE1 |
    +-----+       +-----+       |
    |       +-----+       |
    | VLAN200 |       | VLAN300 |
    +-----+       +-----+       |
    |       +-----+       |
    |   DU2 |       |  PE2 |
    +-----+       +-----+       |
    |       +-----+       |
    |       +-----+       |
    |       +-----+       |
    |       +-----+       |
    |       +-----+       |
```

POST: /restconf/data/ietf-network-slice:ietf-network-slices
Host: example.com
Content-Type: application/yang-data+json

```json
{
    "ietf-network-slice:network-slices": {
        "network-slice": [
            {
                "ns-id": "NS1",
                "ns-description": "URLLC",
                "ns-tags": [
                    {
                        "ns-tag": {
                            "index": 1,
                            "ns-tag-type": "ns-tag-customer",
                            "ns-tag-value": "FOO"
                        }
                    },
                    {
                        "ns-tag": {
                            "index": 2,
                            "ns-tag-type": "ns-tag-customer",
                            "ns-tag-value": "BAR"
                        }
                    }
                ]
            }
        ]
    }
}
```
"index": 3,
"ns-tag-type": "ns-tag-service",
"ns-tag-value": "L2"
},
"status": {
  "admin-enabled": true,
  "oper-status": "up"
},
"ns-endpoints": {
  "ns-endpoint": [
    {
      "ep-id": "DU1",
      "ep-description": "DU1 at location X",
      "ep-ip": "1.1.1.1",
      "ns-match-criteria": {
        "ns-match-criterion": [
          {
            "index": 0,
            "match-type": "ns-vlan-match",
            "values": [
              {
                "index": 1,
                "value": "VLAN-100"
              }
            ],
            "target-ns-connection-group-id": "Matrix1"
          },
          {
            "index": 1,
            "match-type": "ns-vlan-match",
            "values": [
              {
                "index": 1,
                "value": "VLAN-200"
              },
              {
                "index": 2,
                "value": "VLAN-300"
              }
            ],
            "target-ns-connection-group-id": "Matrix2"
          }
        ]
      },
      "ep-network-access-points": {
        "ep-network-access-point": [

{ "network-access-id": "AC1-VRF100",
  "network-access-description": "VRF100 to PE1",
  "network-access-node-id": "PE1",
  "network-access-tp-id": "1",
  "network-access-tp-ip-address": "192.0.1.2",
  "network-access-tp-ip-prefix-length": 24,
  "network-access-qos-policy-name": "QoS-Gold",
  "network-access-tags": {
    "network-access-tag": [
      { "index": 1,
        "network-access-tag-type": "network-access-tag-vlan-id"
      },
      { "index": 2,
        "network-access-tag-type": "network-access-tag-vrf-id",
        "network-access-tag-value": "FOO"
      }
    ],
    "ep-peering": {
      "protocol": [ { "protocol-type": "peering-protocol-bgp",
                      "attribute": [ { "index": 1,
                                       "value": "COLOR:10"
                                 },
                                      { "index": 2,
                                        "value": "RT:20"
                                      },
                                      { "index": 3,
                                        "value": "RT:30"
                                      }
                                 ]
                },
          "incoming-rate-limits": { "cir": "1000000",
                                   "cbs": "1000",
                                   "pir": "5000000",
                                   "pbs": "1000" } } ]
  }
}
{  
  "network-access-id": "AC2-VRF200",
  "network-access-description": "VRF200 to PE1",
  "network-access-node-id": "PE1",
  "network-access-tp-id": "2",
  "network-access-tp-ip-address": "192.0.2.2",
  "network-access-tp-ip-prefix-length": 24,
  "network-access-qos-policy-name": "QoS-Gold",
  "network-access-tags": {
    "network-access-tag": [
      {  
        "index": 1,
        "network-access-tag-type": "network-access-tag-vlan-id",
        "network-access-tag-value": "100"
      },
      {  
        "index": 2,
        "network-access-tag-type": "network-access-tag-vrf-id",
        "network-access-tag-value": "FOO"
      }
    ]
  },
  "ep-peering": {
    "protocol": {
      {  
        "protocol-type": "peering-protocol-bgp",
        "attribute": {
          {  
            "index": 1,
            "value": "COLOR:10"
          },
          {  
            "index": 2,
            "value": "RT:20"
          },
          {  
            "index": 3,
            "value": "RT:30"
          }
        }
      }
    },
    "incoming-rate-limits": {
      "cir": "1000000",
      "cbs": "1000",
    }
  }
}
"pir": "5000000",
"pbs": "1000"
}
]
}
)
,
{
"ep-id": "DU2",
"ep-description": "DU2 at location Y",
"ep-ip": "2.2.2.2",
"ep-network-access-points": {
  "ep-network-access-point": [
    {
      "network-access-id": "AC1-VRF100",
      "network-access-description": "VRF100 to PE2",
      "network-access-node-id": "PE2",
      "network-access-tp-id": "1",
      "network-access-tp-ip-address": "192.1.1.2",
      "network-access-tp-ip-prefix-length": 24,
      "network-access-qos-policy-name": "QoS-Gold",
      "ep-peering": {
        "protocol": {
          "protocol-type": "peering-protocol-bgp",
          "attribute": [
            {
              "index": 1,
              "value": "COLOR:10"
            },
            {
              "index": 2,
              "value": "RT:20"
            },
            {
              "index": 3,
              "value": "RT:30"
            }
          ]
        }
      }
    }
  ]
}
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
},
{
    "network-access-id": "AC2-VRF200",
    "network-access-description": "VRF200 to PE1",
    "network-access-node-id": "PE2",
    "network-access-tp-id": "2",
    "network-access-tp-ip-address": "192.1.2.2",
    "network-access-tp-ip-prefix-length": 24,
    "network-access-qos-policy-name": "QoS-Gold",
    "ep-peering": [
        "protocol": [
            "protocol-type": "peering-protocol-bgp",
            "attribute": [
                {"index": 1, "value": "COLOR:10"},
                {"index": 2, "value": "RT:20"},
                {"index": 3, "value": "RT:30"}
            ]
        ]
    ],
    "incoming-rate-limits": {
        "cir": "1000000",
        "cbs": "1000",
        "pir": "5000000",
        "pbs": "1000"
    }
},
{
    "ep-id": "CU1",
    "ep-description": "CU1 at location Z",
    "ep-ip": "3.3.3.3",
    "ep-network-access-points": {
        "ep-network-access-point": [
            {"network-access-id": "AC1-VRF100"},
           ㎞
"network-access-description": "VRF100 to PE2",
"network-access-node-id": "PE3",
"network-access-tp-id": "1",
"network-access-tp-ip-address": "192.2.1.2",
"network-access-tp-ip-prefix-length": 24,
"network-access-qos-policy-name": "QoS-Gold",
"ep-peering": {
  "protocol": [
    {
      "protocol-type": "peering-protocol-bgp",
      "attribute": [
        {
          "index": 1,
          "value": "COLOR:10"
        },
        {
          "index": 2,
          "value": "RT:20"
        },
        {
          "index": 3,
          "value": "RT:30"
        }
      ]
    }
  ]
},
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
},
{
  "network-access-id": "AC2-VRF200",
  "network-access-description": "VRF200 to PE1",
  "network-access-node-id": "PE3",
  "network-access-tp-id": "2",
  "network-access-tp-ip-address": "192.2.2.2",
  "network-access-tp-ip-prefix-length": 24,
  "network-access-qos-policy-name": "QoS-Gold",
  "ep-peering": {
    "protocol": [
      {
        "protocol-type": "peering-protocol-bgp",
        "attribute": [
          {
            "index": 1,
            "value": "COLOR:10"
          },
          {
            "index": 2,
            "value": "RT:20"
          },
          {
            "index": 3,
            "value": "RT:30"
          }
        ]
      }
    ]
  }
}
"data": {
  "network-slice-profile": {
    "index": 1,
    "value": "COLOR:10"
  },
  "incoming-rate-limits": {
    "cir": "1000000",
    "cbs": "1000",
    "pir": "5000000",
    "pbs": "1000"
  }
}
"ns-connection-groups": {
  "ns-connection-group": [
    {
      "ns-connection-group-id": "Matrix1",
      "slo-sle-policy": {
        "policy-description": "URLLC-SLAs-Template1",
        "ns-metric-bounds": {
          "ns-metric-bound": [
            {
              "metric-type": "ns-slo-shared-bandwidth",
              "metric-unit": "Gbps",
              "value-description": "Shared bandwidth for Matrix1 connection",
              "bound": "15"
            },
            {
              "metric-type": "ns-slo-one-way-bandwidth",
              "metric-unit": "Gbps",
              "value-description": "One-way bandwidth for Matrix3 connections",
              "bound": "10"
            }
          ]
        }
      }
    }
  ]
}
"metric-type": "ns-slo-one-way-delay",
"metric-unit": "msec",
"value-description": "One-way delay for Matrix3 connections",
"}
{
"metric-type": "ns-slo-one-way-delay-variation",
"metric-unit": "msec",
"value-description": "One-way delay variation for Matrix3 connections",
"
"ns-connection": [
{
"ns-connection-id": 1,
"src-nse": ["DU1"],
"dest-nse": ["CU1"],
"slo-sle-policy": {
"ns-metric-bounds": {
"ns-metric-bound": [
{
"metric-type": "ns-slo-one-way-delay",
"metric-unit": "msec",
"bound": "20"
}
]
}
}
],
"ns-connection-id": 2,
"src-nse": ["DU2"],
"dest-nse": ["CU1"
]
}]
},
"ns-connection-group-id": "Matrix2",
"slo-sle-template": "URLLC-SLAs-Template2",
"ns-connection": [
Appendix B. Comparison with Other Possible Design choices for IETF Network Slice Service Interface

According to the 5.3.1 IETF Network Slice Service Interface [I-D.ietf-teas-ietf-network-slices], the Network Slice service Interface is a technology-agnostic interface, which is used for a customer to express requirements for a particular IETF Network Slice. Customers operate on abstract IETF Network Slices, with details related to their realization hidden. As classified by [RFC8309], the Network Slice service Interface is classified as Customer Service Model.
This draft analyzes the following existing IETF models to identify the gap between the IETF Network Slice service interface requirements.

B.1. ACTN VN Model Augmentation

The difference between the ACTN VN model and the IETF Network Slice service requirements is that the IETF Network Slice service interface 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 customer 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 service requirements and the IETF basic network model is that the IETF Network Slice service requests abstract customer 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:

* Network: a transport network list and an list of nodes contained in the network
* Link: "links" list and "termination points" list describe how nodes in a network are connected to each other
* 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.
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 needs to distinguish different IETF Network Slice traffic of the same gNB. Therefore, in addition to using "node-id" and "ep-ip" to identify a Network Slice Endpoint, 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
IETF Network Slice Service YANG Model
draft-ietf-teas-ietf-network-slice-nbi-yang-02

Abstract

This document defines a YANG model for the IETF Network Slice service. The model can be used by an IETF Network Slice customer to manage IETF Network Slices.

Status of This Memo

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

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

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

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

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

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

Code Components
1. Introduction

This document defines a YANG [RFC7950] data model for the IETF Network Slice service.

The YANG model discussed in this document is defined based on the description of the IETF Network Slice service in [I-D.ietf-teas-ietf-network-slices], which is used to operate IETF Network Slices during the IETF Network Slice instantiation. This YANG model supports various operations on IETF Network Slices such as creation, modification, deletion, and monitoring.

The YANG model discussed in this document describes the requirements of an IETF Network Slice service from the point of view of the customer. It is thus classified as customer service model in [RFC8309].

The IETF Network Slice operational state is included in the same tree as the configuration consistent with Network Management Datastore Architecture [RFC8342].

Editorial Note: (To be removed by RFC Editor)

This draft contains several placeholder values that need to be replaced with finalized values at the time of publication. Please apply the following replacements:

* "XXXX" --> the assigned RFC value for this draft both in this draft and in the YANG models under the revision statement.

* The "revision" date in model, in the format XXXX-XX-XX, needs to be updated with the date the draft gets approved.

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:

* client

* configuration data
* state data

This document makes use of the terms defined in [RFC7950].

The tree diagram used in this document follow the notation defined in [RFC8340].

This document also makes use of the terms introduced in the Framework for IETF Network Slices [I-D.ietf-teas-ietf-network-slices]:

* Attachment Circuit (AC): as defined in section 3.1 [I-D.ietf-teas-ietf-network-slices].

* Service Demarcation Point (SDP): The point at which an IETF Network Slice service is delivered by a service provider to a customer, as defined in section 4.2 [I-D.ietf-teas-ietf-network-slices].

* Connectivity Construct: A set of SDPs together with a communication type that defines how traffic flows between the SDPs, as defined in section 3.2 [I-D.ietf-teas-ietf-network-slices].

This document defines the following terms:

* Connection Group: Connection group is an arbitrary collection of one or more connection constructs, which can be used for the following possible purposes:

  Assign the same SLO/SLE policies to multiple connectivity constructs unless SLO/SLE policy is explicitly overridden at the individual connectivity construct level.

  Combine multiple connectivity constructs to support some well-known connectivity types, such as bidirectional unicast service, multipoint-to-point (MP2P) service, hub-and-spoke service etc.

  Share specific SLO limits within multiple connectivity constructs.

2.1. Acronyms

The following acronyms are used in the document:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>Attachment Circuit</td>
</tr>
<tr>
<td>CE</td>
<td>Customer Edge</td>
</tr>
<tr>
<td>NSC</td>
<td>Network Slice Controller</td>
</tr>
</tbody>
</table>
3. IETF Network Slice Service Overview

As defined in [I-D.ietf-teas-ietf-network-slices], an IETF Network Slice service is specified in terms of a set of SDPs, a set of one or more connectivity constructs between subsets of these SDPs, and a set of SLOs and SLEs (see Section 4) for each SDP sending to each connectivity construct. A communication type (point-to-point (P2P), point-to-multipoint (P2MP), or any-to-any (A2A)) is specified for each connectivity construct.

A slice service may include only one connectivity construct or multiple connectivity constructs that associate sets of SDPs. The SDPs serve as the IETF Network Slice ingress/egress points. An SDP is identified by a unique identifier in the context of an IETF Network Slice service.

An example of IETF network slice services containing only one connectivity construct is shown in Figure 1.
Figure 1: An Example of IETF Network Slice Services

An example of IETF network slice services containing multiple connectivity constructs is shown in Figure 2.
Figure 2: An Example of IETF Network Slice Services

As shown in the example, The IETF network slice service 4 contains two P2P connectivity constructs between various SDPs. The IETF network slice service 5 is a bidirectional P2P service on SDP14 and SDP15. The service consists of two unidirectional P2P connectivity constructs. For the traffic from each SDPs, there is only one connectivity construct. For the IETF network slice service 6, an P2P service on SDP 16 and SDP 17, this service includes two unidirectional P2P connectivity constructs with different set of SLOs between the two SDPs. Since SDP16 is the source of two connectivity
constructs, this requires matching criteria be applied to traffic of SDP16 to distinguish between flows constructed by two connections. One example could be that the SDP16 uses DSCPs in the packets to differentiate traffic and steers to different connectivity constructs to ensure different SLOs.

4. IETF Network Slice Service Model Usage

The intention of the IETF Network Slice service model is to allow the customer to manage IETF Network Slices. In particular, the model allows customers to operate in an abstract and technology-agnostic manner, with details of the IETF Network Slices realization hidden.

According to the [I-D.ietf-teas-ietf-network-slices] description, IETF Network Slices are applicable to use cases such as (but not limited to) network wholesale services, network infrastructure sharing among operators, NFV (Network Function Virtualization) connectivity, Data Center Interconnect, and 5G E2E network slice.

As shown in Figure 3, in all these use-cases, the model is used by the customer higher level operation system to communicate with NSC for life cycle management of IETF Network Slices including both enablement and monitoring. For example, in 5G E2E (End-to-end) 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.

The IETF Network Slice Controller (NSC) is a logical entity that allows customers to manage IETF network slices. The NSC receives request from its customer-facing interface (e.g., from a management system). This interface carries data objects the IETF Network Slice service customer provides, describing the needed IETF network slices service in terms of a set of SDPs, the associated connectivity constructs and the service objectives that the customer wishes to be fulfilled. These requirements are then translated into technology-specific actions that are implemented in the underlying network using a network-facing interface. The details of how the IETF Network Slices are put into effect are out of scope for this document.
5. IETF Network Slice Service Modeling Description

The 'ietf-network-slice-service' module uses two main data nodes: list 'slice-service' and container 'slo-sle-templates' (see Figure 4).

The 'slice-service' list includes the set of IETF Network Slice services managed within a provider network. 'slice-service' is the data structure that abstracts an IETF Network Slice service. Under the "slice-service", list "sdp" is used to abstract the SDPs. And list "connection-group" is used to abstract connectivity constructs between SDPs.

The 'slo-sle-templates' container is used by the NSC to maintain a set of common network slice SLO and SLE templates that apply to one or several IETF Network Slice services.

The figure below describes the overall structure of the YANG module:

```
module: ietf-network-slice-service
  +--rw network-slice-services
    +--rw slo-sle-templates
      |  +--rw slo-sle-template* [id]
      |     +--rw id                      string
      |     +--rw template-description?   string
      +--rw slice-service* [service-id]
        +--rw service-id                      string
        +--rw service-description?            string
        +--rw service-tags
          +--rw tag-type* [tag-type]
          |  +--rw tag-type identityref
          |  +--rw value* string
          +--rw tag-opaque* [tag-name]
            +--rw tag-name string
```
+++rw value*                  string
++-rw (slo-sle-policy)?
+++:(standard)
|  ++-rw slo-sle-template?                leafref
+++:(custom)
 ++-rw service-slo-sle-policy
++-rw policy-description?  string
++-rw metric-bounds
  +++rw metric-bound* [metric-type]
    +++rw metric-type     identityref
    +++rw metric-unit     string
    +++rw value-description?  string
    +++rw bound?        uint64
  +++rw security*            identityref
  +++rw isolation?            identityref
  +++rw max-occupancy-level?  uint8
  +++rw mtu?                  uint16
  +++rw steering-constraints
    +++-rw path-constraints
    +++-rw service-function
+++rw status
++-rw admin-status
  |  ++-rw status?     identityref
  |  ++-rw last-updated?   yang:date-and-time
++-ro oper-status
  +++-rw status?     identityref
  +++-ro last-updated?    yang:date-and-time
+++-rw sdps
++-rw sdp* [sdp-id]
  +++rw sdp-id                string
  +++rw sdp-description?      string
  +++rw location
    +++-rw altitude?        int64
    +++-rw latitude?        decimal64
    +++-rw longitude?       decimal64
  +++-rw node-id?            string
  +++-rw sdp-ip?              inet:ip-address
  +++-rw service-match-criteria
    +++-rw match-criterion* [index]
      +++-rw index        uint32
      |      +++-rw match-type
      |                  | identityref
      +++-rw value*      string
      +++-rw target-connection-group-id leafref
      +++-rw connection-group-sdp-role?
        | identityref
        | +++-rw target-connectivity-construct-id?  leafref
      +++-rw sdp-peering

| +--rw protocol* [protocol-type] |
|    +--rw protocol-type identityref |
|    +--rw attribute* [attribute-type] |
|        +--rw attribute-type identityref |
|        +--rw value* string |
|    +--rw opaque* [attribute-name] |
|        +--rw attribute-name string |
|        +--rw value* string |
| +--rw attachment-circuits |
|    +--rw attachment-circuit* [ac-id] |
|        +--rw ac-id string |
|        +--rw ac-description? string |
|        +--rw ac-node-id? string |
|        +--rw ac-tp-id? string |
|        +--rw ac-ip-address? inet:ip-address |
|        +--rw ac-ip-prefix-length? uint8 |
|        +--rw ac-gos-policy-name? string |
|        +--rw mtu? uint16 |
|        +--rw ac-tags |
|            +--rw ac-tags* [ac-tag-type] |
|                +--rw ac-tag-type identityref |
|                +--rw value* string |
|            +--rw ac-tag-opaque* [tag-name] |
|                +--rw tag-name string |
|                +--rw value* string |
|    +--rw service-match-criteria |
|        +--rw match-criterion* [index] |
|            +--rw index uint32 |
|            +--rw match-type identityref |
|            +--rw value* string |
|            +--rw target-connection-group-id leafref |
|            +--rw connection-group-sdp-role? identityref |
|            +--rw target-connectivity-construct-id? leafref |
| +--rw sdp-peering |
|    +--rw protocol* [protocol-type] |
|        +--rw protocol-type identityref |
|        +--rw attribute* [attribute-type] |
|            +--rw attribute-type identityref |
|            +--rw value* string |
|        +--rw opaque* [attribute-name] |
|            +--rw attribute-name string |
|            +--rw value* string |
|    +--rw incoming-rate-limits |
|        +--rw cir? uint64 |
++-rw cbs?  uint64
++-rw eir?  uint64
++-rw ebs?  uint64
++-rw pir?  uint64
++-rw pbs?  uint64
++-rw outgoing-rate-limits
  ++-rw cir?  uint64
  ++-rw cbs?  uint64
  ++-rw eir?  uint64
  ++-rw ebs?  uint64
  ++-rw pir?  uint64
  ++-rw pbs?  uint64
++-rw incoming-rate-limits
  ++-rw cir?  uint64
  ++-rw cbs?  uint64
  ++-rw eir?  uint64
  ++-rw ebs?  uint64
  ++-rw pir?  uint64
  ++-rw pbs?  uint64
++-rw status
  ++-rw admin-status
    ++-rw status?         identityref
    ++-rw last-updated?   yang:date-and-time
  ++-ro oper-status
    ++-ro status?         identityref
    ++-ro last-updated?   yang:date-and-time
++-ro sdp-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 connection-groups
  ++-rw connection-group* [connection-group-id]
    ++-rw connection-group-id     string
    ++-rw connectivity-type?      identityref
    ++-rw (slo-sle-policy)?
      +--:(standard)
        |  ++-rw slo-sle-template?      leafref
      +--:(custom)
++-rw bound?
  |  uint64
++-rw security*
  |  identityref
++-rw isolation?
  |  identityref
++-rw max-occupancy-level?
  |  uint8
++-rw mtu?
  |  uint16
+-rw steering-constraints
  +--rw path-constraints
  +--rw service-function
+-rw (slo-sle-policy)?
  +--:(standard)
    |  ++-rw slo-sle-template?  leafref
  +--:(custom)
    ++-rw service-slo-sle-policy
      ++-rw policy-description?  string
      ++-rw metric-bounds
        +--rw metric-bound*  [metric-type]
          +--rw metric-type
            |  identityref
          +--rw metric-unit  string
          +--rw value-description?  string
          +--rw bound?  uint64
    ++-rw security*
    ++-rw isolation?
    ++-rw max-occupancy-level?  uint8
    ++-rw mtu?
    ++-rw steering-constraints
      +--rw path-constraints
      +--rw service-function
    +--ro connectivity-construct-monitoring
      +--ro one-way-min-delay?  uint32
      +--ro one-way-max-delay?  uint32
      +--ro one-way-delay-variation?  uint32
      +--ro one-way-packet-loss?  decimal64
      +--ro two-way-min-delay?  uint32
      +--ro two-way-max-delay?  uint32
      +--ro two-way-delay-variation?  uint32
      +--ro two-way-packet-loss?  decimal64
    +--ro connection-group-monitoring
      +--ro one-way-min-delay?  uint32
      +--ro one-way-max-delay?  uint32
      +--ro one-way-delay-variation?  uint32
      +--ro one-way-packet-loss?  decimal64
      +--ro two-way-min-delay?  uint32
5.1. IETF Network Slice Service SLO and SLE Templates

The ‘ns-templates’ container (Figure 4) is used by the 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 Slice services. The exact definition of the templates is deployment specific to each network provider.

The model includes only the identifiers of SLO and SLE templates. When creation of IETF Network slice, the SLO and SLE policies can be easily identified.

The following shows an example where two network slice templates can be retrieved by the customer higher level operation system:

```
{
  "network-slice-services": {
    "slo-sle-templates": {
      "slo-sle-template": [
        {
          "id": "GOLD-template",
          "template-description": "Two-way bandwidth: 1 Gbps, one-way latency 100ms",
          "sle-isolation": "service-isolation-shared",
        },
        {
          "id": "PLATINUM-template",
          "template-description": "Two-way bandwidth: 1 Gbps, one-way latency 50ms",
          "sle-isolation": "service-isolation-dedicated",
        }
      ]
    }
  }
}
```

5.2. IETF Network Slice Service

The ‘slice-service’ is the data structure that abstracts an IETF Network Slice service. Each ‘slice-service’ is uniquely identified by an identifier: ‘service-id’.

An IETF Network Slice service has the following main parameters:

* "service-id": Is an identifier that is used to uniquely identify the IETF Network Slice service within NSC.

* "service-description": Gives some description of an IETF Network Slice service.

* "status": Is used to show the operative and administrative status of the IETF Network Slice service, and can be used as indicator to detect network slice anomalies.

* "service-tags": It is a means to correlate the higher level "Customer higher level operation system" and IETF network slices. It might be used by IETF network Slice service operator to provide additional information to the IETF Network Slice Controller (NSC) during the automation of the IETF network slices. E.g. adding tag with "customer-name" when multiple actual customers use a same network slice service. Another use-case for "service-tag" might be for an operator to provide additional attributes to NSC which might be used during the realization of IETF Network Slice services such as type of services (e.g., L2 or L3). These additional attributes can also be used by the NSC for various use-cases such as monitoring and assurance of the IETF Network Slice services where NSC can notify the customer system by issuing the notifications. Note that all these attributes are OPTIONAL but might be useful for some use-cases.

* "slo-sle-policy": Defines SLO and SLE policies for the "slice-service". More description are provided in Section 5.2.3

* "sdp": Represents a set of endpoints (SDPs) involved in the IETF Network Slice service with each 'sdp' belonging to a single 'slice-service'. More description are provided in Section 5.2.1.

* "connection-groups": Abstracts the connections to the set of SDPs of the IETF Network Slice service.

5.2.1. IETF Network Slice Service Demarcation Point

An SDP belong to a single IETF Network Slice service. An IETF Network Slice service involves two or more SDPs. An IETF Network Slice service can be modified by adding new "sdp" or removing existing "sdp".

Section 4.2 [I-D.ietf-teas-ietf-network-slices] describes four possible ways in which the SDP may be placed:
* Within CE
* Provider-facing ports on the CE
* Customer-facing ports on the PE
* Within PE

In the four options, the Attachment Circuit (AC) may be part of the IETF Network Slice service or may be external to it. Based on the definition of AC in section 2.1 [I-D.ietf-teas-ietf-network-slices], the customer and provider may agree on a per (IETF Network Slice service, connectivity construct, and SLOs/SLEs) basis to police or shape traffic on the AC in both the ingress (CE to PE) direction and egress (PE to CE) direction. This ensures that the traffic is within the capacity profile that is agreed in an IETF Network Slice service. Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider.

An IETF Network Slice SDP has several characteristics:

* "sdp-id": Uniquely identifies the SDP within the Network Slice Controller (NSC). The identifier is a string that allows any encoding for the local administration of the IETF Network Slice service.

* "location": Indicates SDP location information, which helps the NSC to identify an SDP.

* "node-id": The SDP node information, which helps the NSC to identify an SDP.

* "sdp-ip": The SDP IP information, which helps the NSC to identify an SDP.

* "service-match-criteria": Defines matching policies for network slice service traffic to apply on a given SDP.

* "attachment-circuit": Specifies the list of ACs by which the service traffic is received. This is an optional SDP attribute. When a SDP has multiple ACs and the AC specific attributes is needed, each "attachment-circuit" can specify attributes such as interface specific IP address, service MTU, "service-match-criteria", etc.

* "incoming-rate-limits" and "outgoing-rate-limits": Set the rate-limiting policies to apply on a given SDP, 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 SDP to the IETF scope Network that passes through the AC. When Bandwidth is applied to the outgoing direction, it is applied to the traffic from the IETF Network to the SDP of that particular slice service. If an SDP has multiple AC, the "incoming-rate-limits" and "outgoing-rate-limits" of "attachment-circuit" can be set to an AC specific value, but the rate cannot exceed the "incoming-rate-limits" and "outgoing-rate-limits" of the SDP. If a SDP only contains a single AC, then the "incoming-rate-limits" and "outgoing-rate-limits" of "attachment-circuit" is the same with the SDP "incoming-rate-limits" and "outgoing-rate-limits". The definition refers to [I-D.ietf-teas-ietf-network-slices].

* "sdp-peering": Specifies the protocol for a SDP for exchanging control-plane information, e.g. L1 signaling protocol or L3 routing protocols, etc.

* "status": Enables the control of the operative and administrative status of the SDP, can be used as indicator to detect SDP anomalies.

The customer may choose to use an explicit "service-match-criteria" to map all the SDP’s traffic or a subset of the SDP’s traffic to a specific connection-group or connectivity-construct.

If an SDP is placed at the port or AC of a CE or PE, and there is only one single connectivity construct with a source at the SDP, traffic can be implicitly mapped to this connectivity construct since the port or AC can be used to identify the traffic and the SDP is the only source of the connectivity-construct.

If an SDP is placed within CE or PE, or there are many single connectivity constructs with a source at the SDP. Traffic needs to be explicitly mapped into the IETF Network Slice’s specific connectivity construct. The policies, "service-match-criteria", are based on the values in which combination of layer 2 and layer 3 header and payload fields within a packet to identify to which (IETF Network Slice service, connectivity construct, and SLOs/SLEs) that packet is assigned.

The customer may choose to use an explicit match-type of "match-any" to map all the SDP’s traffic to the appropriate connection-group or connectivity-construct.

Similarly, if a subset of traffic is matched (ie. dscp-match) and mapped to a connectivity-construct, the customer may choose to add a subsequent "match-any" to explicitly map the remaining SDP traffic to a separate connectivity-construct. If the customer chooses to
implicitly map remaining traffic and if there is no additional connectivity constructs where the "sdp-id" source is specified, then that traffic will be dropped.

While explicit matching is optional in some use cases, explicit matching provides a more clear and readable implementation, but the choice is left to the customer.

To illustrate the use of SDP options, the below are two examples. How the NSC realize the mapping is out of scope for this document.

* SDPs at customer-facing ports on the PEs: As shown in Figure 5, customer of the IETF network slice service would like to connect two SDPs to satisfy specific service, e.g., Network wholesale services. In this case, the IETF network slice SDPs are mapped to customer-facing ports of PE nodes. The IETF network slice controller (NSC) uses ‘node-id’ (PE device ID), ‘attachment-circuit’ (ACs) to map SDPs to the customer-facing ports on the PEs.

---

**Legend:**

| O | Representation of the IETF network slice endpoints (SDP) |
| + | Mapping of SDP to customer-facing ports on the PE |
| X | Physical interfaces used for realization of IETF network slice service |
| S1 | L0/L1/L2/L3 services used for realization of IETF network slice service |
| T1 | Tunnels used for realization of IETF network slice service |

---

Figure 5
* SDPs within CEs: As shown in Figure 6, customer of the IETF network slice service would like to connect two SDPs to provide connectivity between transport portion of 5G RAN to 5G Core network functions. In this scenario, the IETF network slice controller (NSC) uses 'node-id' (CE device ID), 'sdp-ip' (CE tunnel endpoint IP), 'service-match-criteria' (VLAN tag), 'attachment-circuit' (ACs) to map SDPs to the CE tunnel endpoints. And the NSC can also retrieve the corresponding ACs, or PEs, and further map the slice service to services/tunnels/paths.

SDP3
(With CE1 parameters)
+<--------------- IETF Network Slice 2 -------------->o
+                          +
+                         +
+                        +
+                       +
+                   ++++++++ v
+               +              v
+ o        PE1|-------------------| PE2 | o
+            |       X-----------------X   |
+       +-----+                    AC
Customer       Provider           Provider       Customer
Edge 1          Edge 1             Edge 2          Edge 2

SDP4
(with CE2 parameters)

Legend:
O: Representation of the IETF network slice endpoints (SDP)
+: Mapping of SDP to CE
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 6

5.2.2. IETF Network Slice Service Connectivity Construct

Based on the customer’s service traffic requirements, an IETF Network Slice service connectivity type could be point-to-point (P2P), point-to-multipoint (P2MP), any-to-any (A2A) or a combination of these types.
[I-D.ietf-teas-ietf-network-slices] defines the basic connectivity construct for a network slice, and the connectivity construct may have different SLO and SLE requirements. "connectivity-construct" represents this connectivity construct, and "slo-sle-policy" under it represents the per-connectivity construct SLO and SLE requirements.

Apart from the per-connectivity construct SLO and SLE, slice service traffic is usually managed by combining similar types of traffic. For example, some connections for video services require high bandwidth, and some connections for voice over IP request low latency and reliability. "connection-group" is thus defined to treat each type as a class with per-connection-group SLO and SLE such that the connectivity construct can inherit the SLO/SLE from the group if not explicitly defined.

5.2.3. IETF Network Slice Service SLO and SLE Policy

As defined in section 4 [I-D.ietf-teas-ietf-network-slices], the SLO and SLE policy of an IETF Network Slice service defines some common attributes.

"slo-sle-policy" is used to represent specific SLO and SLE policies. During the creation of an IETF Network Slice service, the policy can be specified either by a standard SLO and SLO template or a customized SLO and SLE policy.

The policy can apply to per-network slice service, per-connection group "connection group", or per-connectivity construct "connectivity-construct". Since there are multiple mechanisms for assigning a policy to a single connectivity construct, an overridden precedence order among them is as follows:

* Connectivity-construct at a individual sending SDP
* Connectivity-construct
* Connection-group
* Slice-level

That is, the policy assigned through the sending SDP has highest precedence, and the policy assigned by the slice level has lowest precedence. Therefore, the policy assigned through the sending SDP takes precedence over the policy assigned through the connection-construct entry.
The container "metric-bounds" supports all the variations and combinations of SLOs, which includes a list of "metric-bound" and each "metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and supports the following options:

"service-slo-one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two SDP. And the bandwidth is unidirectional.

"service-slo-two-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two SDP. And the bandwidth is bidirectional.

"service-slo-one-way-delay": Indicates the maximum one-way latency between two SDP.

"service-slo-two-way-delay": Indicates the maximum round-trip latency between two SDP.

"service-slo-one-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the one-way latency between sequential packets in a flow.

"service-slo-two-way-delay-variation": Indicates the jitter constraint of the slice maximum permissible delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

"service-slo-one-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"service-slo-two-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two endpoints.

"service-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.

The following common SLEs are defined:

"mtu": Refers to the service MTU, which is the maximum PDU size that the customer may use.
"security": Includes the request for encryption or other security techniques to traffic flowing between the two NS endpoints.

"isolation": Specifies the isolation level that a customer expects, including dedicated, shared, or other level.

max-occupancy-level: Specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume.

"steering-constraints": Specifies the constraints how the provider routes traffic for the IETF Network Slice service.

The following shows an example where a network slice policy can be configured:

```json
{
  "slice-services": {
    "slice-service": {
      "service-id": "exp-slice",
      "service-slo-sle-policy": {
        "policy-description": "video-service-policy",
        "metric-bounds": {
          "metric-bound": [
            {
              "metric-type": "service-slo-one-way-bandwidth",
              "metric-unit": "mbps",
              "bound": "1000"
            },
            {
              "metric-type": "service-slo-availability",
              "bound": "99.9%"
            }
          ]
        }
      }
    }
  }
}
```

5.2.4. IETF Network Slice Service Monitoring

An IETF Network Slice service defines connectivity with specific SLO characteristics, including bandwidth, latency, etc. The connectivity is a combination of logical unidirectional connections, represented by ‘connectivity-construct’.
This model also describes performance status of an IETF Network Slice. The statistics are described in the following granularity:

* Per SDP: specified in 'sdp-monitoring' under the "sdp".

* Per connectivity construct: specified in 'connectivity-construct-monitoring' under the "connectivity-construct".

* Per connection group: specified in 'connection-group-monitoring' under the "connection-group".

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 "sdp", "connectivity-construct", or "connection-group", so that only interested contents will be sent. These mechanisms can be used for monitoring the IETF Network Slice performance status so that the customer management system could initiate modification based on the IETF Network Slice running status.

6. IETF Network Slice Service Module

The "ietf-network-slice" module uses types defined in [RFC6991], [RFC9181], and [RFC8776], and [RFC7640].

<CODE BEGINS> file "ietf-network-slice-service@2022-07-11.yang"
module ietf-network-slice-service {  
yang-version 1.1;  
namespace  
prefix ietf-nss;  

import ietf-inet-types {  
    prefix inet;  
    reference  
        "RFC 6991: Common YANG Types.";  
}  
import ietf-vpn-common {  
    prefix vpn-common;  
    reference  
        "RFC 9181: A Common YANG Data Model for Layer 2 and Layer 3 VPNs.";  
}  
import ietf-te-types {  
    prefix te-types;  
    reference

"RFC 8776: Common YANG Data Types for Traffic Engineering.";
}
import ietf-te-packet-types {
  prefix te-packet-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: Bo Wu
  <lana.wubo@huawei.com>
Editor: Dhruv Dhody
  <dhruv.ietf@gmail.com>
Editor: Reza Rokui
  <reza.rokui@nokia.com>
Editor: Tarek Saad
  <tsaad@juniper.net>
Editor: Reza Rokui
  <reza.rokui@nokia.com>
Author: Liuyan Han
  <hanliuyan@chinamobile.com>";
description "This module defines a model for the IETF Network Slice service.

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

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

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.";
revision 2022-07-11 {
  description "initial version.";
  reference "RFC XXXX: A Yang Data Model for IETF Network Slice service operation";
}
identity service-tag-type {
    description
        "Base identity for IETF Network Slice service tag type.";
}

identity service-tag-customer {
    base service-tag-type;
    description
        "The IETF Network Slice service customer ID tag type.";
}

identity service-tag-service {
    base service-tag-type;
    description
        "The IETF Network Slice service tag type.";
}

identity service-tag-opaque {
    base service-tag-type;
    description
        "The IETF Network Slice service opaque tag type.";
}

identity attachment-circuit-tag-type {
    description
        "Base identity for the attachment circuit tag type.";
}

identity attachment-circuit-tag-vlan-id {
    base attachment-circuit-tag-type;
    description
        "The attachment circuit VLAN ID tag type.";
}

identity attachment-circuit-tag-ip-mask {
    base attachment-circuit-tag-type;
    description
        "The attachment circuit tag IP mask.";
}

identity service-isolation-type {
    description
        "Base identity for IETF Network slice service isolation level.";
}
identity service-isolation-shared {
    base service-isolation-type;
    description
    "Shared resources (e.g. queues) are associated with the
    slice service traffic. Hence, the traffic can be impacted
    by effects of other services traffic
    sharing the same resources.";
}

identity service-isolation-dedicated {
    base service-isolation-type;
    description
    "Dedicated resources (e.g. queues) are associated with the
    Network Slice service traffic. Hence, the service traffic
    is isolated from other service traffic
    sharing the same resources.";
}

identity service-security-type {
    description
    "Base identity for slice service security level.";
}

identity service-security-authenticate {
    base service-security-type;
    description
    "Indicates the slice service requires authentication.";
}

identity service-security-integrity {
    base service-security-type;
    description
    "Indicates the slice service requires data integrity.";
}

identity service-security-encryption {
    base service-security-type;
    description
    "Indicates the slice service requires data encryption.";
}

identity point-to-point {
    base vpn-common:vpn-topology;
    description
    "Identity for point-to-point IETF Network Slice
    service connectivity.";
}
identity point-to-multipoint {
  base vpn-common:vpn-topology;
  description
    "Identity for point-to-point IETF Network Slice
     service connectivity.";
}

identity multipoint-to-multipoint {
  base vpn-common:vpn-topology;
  description
    "Identity for point-to-point IETF Network Slice
     service connectivity.";
}

identity multipoint-to-point {
  base vpn-common:vpn-topology;
  description
    "Identity for point-to-point IETF Network Slice
     service connectivity.";
}

identity sender-role {
  base vpn-common:role;
  description
    "A SDP is acting as a sender.";
}

identity receiver-role {
  base vpn-common:role;
  description
    "A SDP is acting as a receiver.";
}

identity service-slo-metric-type {
  description
    "Base identity for IETF Network Slice service SLO metric type.";
}

identity service-slo-one-way-bandwidth {
  base service-slo-metric-type;
  description
    "SLO bandwidth metric. Minimum guaranteed bandwidth between
     two SDPs at any time and is measured unidirectionally.";
}

identity service-slo-two-way-bandwidth {
  base service-slo-metric-type;
  description
    "SLO bandwidth metric. Minimum guaranteed bandwidth between
     two SDPs at any time and is measured unidirectionally.";
}
"SLO bandwidth metric. Minimum guaranteed bandwidth between two SDPs at any time."
}

identity service-slo-shared-bandwidth {
    base service-slo-metric-type;
    description
        "The shared SLO bandwidth bound. It is the limit on the bandwidth that can be shared amongst a group of connectivity constructs of a slice service.";
}

identity service-slo-one-way-delay {
    base service-slo-metric-type;
    description
        "SLO one-way-delay is the upper bound of network delay when transmitting between two SDPs. The metric is defined in RFC7679.";
}

identity service-slo-two-way-delay {
    base service-slo-metric-type;
    description
        "SLO two-way delay is the upper bound of network delay when transmitting between two SDPs. The metric is defined in RFC2681.";
}

identity service-slo-one-way-delay-variation {
    base service-slo-metric-type;
    description
        "SLO one-way delay variation is defined by RFC3393, is the difference in the one-way delay between sequential packets between two SDPs.";
}

identity service-slo-two-way-delay-variation {
    base service-slo-metric-type;
    description
        "SLO two-way delay variation is defined by RFC5481, is the difference in the round-trip delay between sequential packets between two SDPs.";
}

identity service-slo-one-way-packet-loss {
    base service-slo-metric-type;
    description
        "SLO loss metric. The ratio of packets dropped to packets
transmitted between two SDPs in one-way
over a period of time as specified in RFC7680.";
}

identity service-slo-two-way-packet-loss {
  base service-slo-metric-type;
  description
    "SLO loss metric. The ratio of packets dropped to packets
    transmitted between two SDPs in two-way
    over a period of time as specified in RFC7680.";
}

identity service-slo-availability {
  base service-slo-metric-type;
  description
    "SLO availability level.";
}

identity service-match-type {
  description
    "Base identity for IETF Network Slice service traffic
    match type.";
}

identity service-phy-interface-match {
  base service-match-type;
  description
    "Use the physical interface as match criteria for
    slice service traffic.";
}

identity service-vlan-match {
  base service-match-type;
  description
    "Use the VLAN ID as match criteria for the slice service
    traffic.";
}

identity service-label-match {
  base service-match-type;
  description
    "Use the MPLS label as match criteria for the slice service
    traffic.";
}

identity service-source-ip-match {
  base service-match-type;
  description
    "Use the source IP address as match criteria for the slice service
    traffic.";
}
"Use source ip-address in the packet header as match criteria for the slice service traffic."
}

identity service-destination-ip-match {
  base service-match-type;
  description
    "Use destination ip-address in the packet header as match criteria for the slice service traffic.";
}

identity service-dscp-match {
  base service-match-type;
  description
    "Use DSCP in the IP packet header as match criteria for the slice service traffic.";
}

identity service-any-match {
  base service-match-type;
  description
    "Match all slice service traffic.";
}

identity peering-protocol-type {
  description
    "Base identity for SDP peering protocol type.";
}

identity peering-protocol-bgp {
  base peering-protocol-type;
  description
    "Use BGP as protocol for SDP peering with customer device.";
}

identity peering-static-routing {
  base peering-protocol-type;
  description
    "Use static routing for SDP peering with customer device.";
}

identity peering-attribute-type {
  description
    "Base identity for BGP peering";
}

identity remote-as {
  base peering-attribute-type;
description
   "Identity for remote-as attribute of BGP peering.";
}

identity neighbor {
    base peering-attribute-type;
    description
       "Identity for neighbor attribute of BGP peering.";
}

identity local-as {
    base peering-attribute-type;
    description
       "Identity for local-as attribute of BGP peering.";
}

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%";
}

/* grouping */
grouping service-match-criteria {
  description
    "A grouping for the slice service match definition.";
  container service-match-criteria {
    description
      "Describes the slice service match criteria.";
    list match-criterion {
      key "index";
      description
        "List of the slice service traffic match criteria.";
      leaf index {
        type uint32;
        description
          "The entry index.";
      }
      leaf match-type {
        type identityref {
          base service-match-type;
        }
        mandatory true;
        description
          "Identifies an entry in the list of the slice service
          match criteria.";
      }
      leaf-list value {
        type string;
        description
          "Describes the slice service match criteria, e.g.
          IP address, VLAN, etc.";
      }
      leaf target-connection-group-id {
        type leafref {
          path
            "/ietf-nss:network-slice-services"
            + "/ietf-nss:slice-service"
            + "/ietf-nss:connection-groups/ietf-nss:connection-group"
            + "/ietf-nss:connection-group-id";
        }
      }
  }
}"
mandatory true;
description
"Reference to the slice service connection group.";
}
leaf connection-group-sdp-role {
type identityref {
  base vpn-common:role;
}
default "vpn-common:any-to-any-role";
description
"Indicates the role in the connection group when
a slice service has multiple multipoint-to-multipoint
connection groups, e.g., hub-spoke."
}
leaf target-connectivity-construct-id {
type leafref {
  path
  "/ietf-nss:network-slice-services/slice-service"
  + "/ietf-nss:connection-groups"
  + "/ietf-nss:connection-group[connection-group-id"
  + "=current()/../target-connection-group-id]"
  + "/ietf-nss:connectivity-construct/ietf-nss:cc-id"
}
description
"Reference to a Network Slice connection construct.";
}
}
}

grouping service-sles {
description
"Indirectly Measurable Objectives of a slice service.";
leaf-list security {
type identityref {
  base service-security-type;
}
description
"The slice service security SLE(s)";
}
leaf isolation {
type identityref {
  base service-isolation-type;
}
default "service-isolation-shared";
description
"The slice service isolation SLE requirement.";
}
leaf max-occupancy-level {
  type uint8 {
    range "1..100";
  }
  description
  "The maximal occupancy level specifies the number of flows to
   be admitted.";
}
leaf mtu {
  type uint16;
  units "bytes";
  description
  "The MTU specifies the maximum length in octets of data
   packets that can be transmitted by the slice service.
   The value needs to be less than or equal to the
   minimum MTU value of all 'attachment-circuits' in the SDPs.";
}
container steering-constraints {
  description
  "Container for the policy of steering constraints
   applicable to the slice service.";
  container path-constraints {
    description
    "Container for the policy of path constraints
     applicable to the slice service.";
  }
  container service-function {
    description
    "Container for the policy of service function
     applicable to the slice service.";
  }
}

grouping service-metric-bounds {
  description
  "Slice service metric bounds grouping.";
  container metric-bounds {
    description
    "Slice service metric bounds container.";
    list metric-bound {
      key "metric-type";
      description
      "List of slice service metric bounds.";
      leaf metric-type {
        type identityref {
          base service-slo-metric-type;
        }
      }
    }
  }
}

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 bound {
  type uint64;
  default "0";
  description
    "The Bound on the slice service connection metric. A zero indicate an unbounded upper limit for the specific metric-type.";
}
}
}
grouping sdp-peering {

description
  "A grouping for the slice service SDP peering.";
}
container sdp-peering {
  description
    "Describes SDP peering attributes.";
  list protocol {
    key "protocol-type";
    description
      "List of the SDP peering protocol.";
    leaf protocol-type {
      type identityref {
        base peering-protocol-type;
      }
      description
        "Identifies an entry in the list of SDP peering protocol type.";
    }
  list attribute {
    key "attribute-type";
description
  "list of protocol attributes";
leaf attribute-type {
  type identityref {
    base peering-attribute-type;
  }
  description
    "identifies the attribute type";
}
leaf-list value {
  type string;
  description
    "Describes the value of protocol attribute, e.g. nexthop address, peer address, etc.";
}
}
list opaque {
  key "attribute-name";
  description
    "List of protocol attributes.";
leaf attribute-name {
  type string;
  description
    "The name of the attribute.";
}
leaf-list value {
  type string;
  description
    "The value(s) of the attribute";
}
}
}
grouping sdp-attachment-circuits {
  description
    "Grouping for the SDP attachment circuit definition.";
container attachment-circuits {
  description
    "List of attachment circuit.";
list attachment-circuit {
  key "ac-id";
  description
    "The IETF Network Slice service SDP attachment circuit related parameters.";
  leaf ac-id {
    type string;
    }}
description
   "Uniquely identifier a attachment circuit.";
}
leaf ac-description {
   type string;
   description
   "The attachment circuit description.";
}
leaf ac-node-id {
   type string;
   description
   "The attachment circuit node ID in the case of
    multi-homing.";
}
leaf ac-tp-id {
   type string;
   description
   "The termination port ID of the attachment circuit.";
}
leaf ac-ip-address {
   type inet:ip-address;
   description
   "The IP address of the attachment circuit.";
}
leaf ac-ip-prefix-length {
   type uint8;
   description
   "The subnet prefix length expressed in bits.";
}
leaf ac-qos-policy-name {
   type string;
   description
   "The name of the QoS policy that is applied to the
    attachment circuit. The name can reference a QoS
    profile that is pre-provisioned on the device.";
}
leaf mtu {
   type uint16;
   units "bytes";
   description
   "Maximum size in octets of the slice service data packet
    that can traverse a SDP.";
}
container ac-tags {
   description
   "Container for the attachment circuit tags.";
   list ac-tags {
      key "ac-tag-type";
   }
}
description
"The attachment circuit tags list.";
leaf ac-tag-type {
  type identityref {
    base attachment-circuit-tag-type;
  }
  description
    "The attachment circuit tag type.";
}
leaf-list value {
  type string;
  description
    "The attachment circuit tag value.";
}
list ac-tag-opaque {
  key "tag-name";
  description
    "The attachment circuit tag opaque list.";
  leaf tag-name {
    type string;
    description
      "The opaque tags name";
  }
  leaf-list value {
    type string;
    description
      "The opaque tags value";
  }
}
/* Per ac rate limits */
uses service-match-criteria;
uses sdp-peering;
uses service-rate-limit;
}
}
grouping sdp-monitoring-metrics {
  description
    "Grouping for the SDP monitoring metrics.";
  container sdp-monitoring {
    config false;
    description
      "Container for SDP monitoring metrics.";
    leaf incoming-utilized-bandwidth {
      type te-types:te-bandwidth;
    }
  }
}
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
        "Outoing service bandwidth utilization at a SDP.";
}

leaf outgoing-bw-utilization {
    type decimal64 {
        fraction-digits 5;
        range "0..100";
    }
    units "percent";
    mandatory true;
    description
        "To be used to define the service bandwidth utilization as a percentage of the available bandwidth.";
}


grouping connectivity-construct-monitoring-metrics {
    description
        "Grouping for connectivity construct monitoring metrics.";
    uses te-packet-types:one-way-performance-metrics-packet;
    uses te-packet-types:two-way-performance-metrics-packet;
}

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 bw-rate-limits {
  description
    "Bandwidth rate limits grouping.";
  reference
    "RFC 7640: Traffic Management Benchmarking";
  leaf cir {
    type uint64;
    units "bps";
    description
      "Committed Information Rate. The maximum number of bits
      that a port can receive or send during one-second over an
      interface.";
  }
  leaf cbs {
    type uint64;
    units "bytes";
    description
      "Committed Burst Size. CBS controls the bursty nature
      of the traffic. Traffic that does not use the configured
      CIR accumulates credits until the credits reach the
      configured CBS.";
  }
}

leaf eir {
    type uint64;
    units "bps";
    description
        "Excess Information Rate, i.e., excess frame delivery
         allowed not subject to SLA. The traffic rate can be
         limited by EIR.";
}

leaf ebs {
    type uint64;
    units "bytes";
    description
        "Excess Burst Size. The bandwidth available for burst
         traffic from the EBS is subject to the amount of
         bandwidth that is accumulated during periods when
         traffic allocated by the EIR policy is not used.";
}

leaf pir {
    type uint64;
    units "bps";
    description
        "Peak Information Rate, i.e., maximum frame delivery
         allowed. It is equal to or less than sum of CIR and EIR.";
}

leaf pbs {
    type uint64;
    units "bytes";
    description
        "Peak Burst Size.";
}

grouping service-rate-limit {
    description
        "The rate limits grouping.";
    container incoming-rate-limits {
        description
            "Container for the asymmetric traffic control.";
        uses bw-rate-limits;
    }
    container outgoing-rate-limits {
        description
            "The rate-limit imposed on outgoing traffic.";
        uses bw-rate-limits;
    }
}

grouping sdp {

description
  "Slice service SDP related information";
leaf sdp-id {
  type string;
  description
    "Unique identifier for the referred slice service SDP.";
}
leaf sdp-description {
  type string;
  description
    "Give more description of the SDP.";
}
uses geolocation-container;
leaf node-id {
  type string;
  description
    "Uniquely identifies an edge node of the SDP.";
}
leaf sdp-ip {
  type inet:ip-address;
  description
    "The IP address of the SDP.";
}
uses service-match-criteria;
uses sdp-peering;
uses sdp-attachment-circuits;
uses service-rate-limit;
/* Per SDP rate limits */
uses vpn-common:service-status;
uses sdp-monitoring-metrics;
}
//service-sdp

grouping connectivity-construct {
  description
    "Grouping for slice service connectivity construct.";
list connectivity-construct {
  key "cc-id";
  description
    "List of connectivity constructs.";
  leaf cc-id {
    type uint32;
    description
      "The connectivity construct identifier.";
  }
  choice connectivity-construct-type {
    default "p2p";
  }
}
description
  "Choice for connectivity construct type.";
case p2p {
  description
    "P2P connectivity construct.";
  leaf p2p-sender-sdp {
    type leafref {
      path "../../../sdps/sdp/sdp-id";
    } 
    description
      "Reference to a sender SDP.";
  }
  leaf p2p-receiver-sdp {
    type leafref {
      path "../../../sdps/sdp/sdp-id";
    } 
    description
      "Reference to a receiver SDP.";
  }
}
case p2mp {
  description
    "P2MP connectivity construct.";
  leaf p2mp-sender-sdp {
    type leafref {
      path "../../../sdps/sdp/sdp-id";
    } 
    description
      "Reference to a sender SDP.";
  }
  leaf-list p2mp-receiver-sdp {
    type leafref {
      path "../../../sdps/sdp/sdp-id";
    } 
    description
      "Reference to a receiver SDP.";
  }
}
case a2a {
  description
    "A2A connectivity construct.";
  list a2a-sdp {
    key "sdp-id";
    description
      "List of included A2A SDPs.";
    leaf sdp-id {
      type leafref {
        path "../../../sdps/sdp/sdp-id";
      } 
    }
  }
}
} description
  "Reference to a SDP."
}
uses service-slo-sle-policy;
}
}
uses service-slo-sle-policy;
/* Per connectivity construct service-slo-sle-policy
 * overrides the per slice service-slo-sle-policy.
 */
container connectivity-construct-monitoring {
  config false;
  description
    "SLO status per connectivity construct.";
  uses connectivity-construct-monitoring-metrics;
}
}

//connectivity-construct

grouping connection-group {
  description
    "Grouping for slice service connection group.";
  leaf connection-group-id {
    type string;
    description
      "The connection group identifier.";
  }
  leaf connectivity-type {
    type identityref {
      base vpn-common:vpn-topology;
    }
    default "vpn-common:any-to-any";
    description
      "Connection group connectivity type.";
  }
  uses service-slo-sle-policy;
  uses connectivity-construct;
  /* Per connection group service-slo-sle-policy overrides
   * the per slice service-slo-sle-policy.
   */
  container connection-group-monitoring {
    config false;
    description
      "SLO status per connection group.";
uses connectivity-construct-monitoring-metrics;

//connection-group

grouping slice-service-template {
    description
    "Grouping for slice service templates.";
    container slo-sle-templates {
        description
        "Contains a set of slice service templates.";
        list slo-sle-template {
            key "id";
            leaf id {
                type string;
                description
                "Identification of the Service Level Objective (SLO) and Service Level Expectation (SLE) template to be used. Local administration meaning.";
            }
            leaf template-description {
                type string;
                description
                "Description of the SLO and SLE policy template.";
            }
            description
            "List for SLO and SLE template identifiers.";
        }
    }
}

/* Configuration data nodes */

grouping service-slo-sle-policy {
    description
    "Slice service policy grouping.";
    choice slo-sle-policy {
        description
        "Choice for SLO and SLE policy template. Can be standard template or customized template.";
        case standard {
            description
            "Standard SLO template.";
            leaf slo-sle-template {
                type leafref {
                    path "/ietf-nss:network-slice-services" + "/ietf-nss:slo-sle-templates"
case standard {
    description
    "Standard SLO and SLE template to be used.";
}
}
case custom {
    description
    "Customized SLO and SLE template.";
    container service-slo-sle-policy {
        description
        "Contains the SLO and SLE policy.";
        leaf policy-description {
            type string;
            description
            "Description of the SLO and SLE policy.";
        }
        uses service-metric-bounds;
        uses service-sles;
    }
}
}

container network-slice-services {
    description
    "Contains a list of IETF network slice services";
    uses slice-service-template;
    list slice-service {
        key "service-id";
        description
        "A slice service is identified by a service-id.";
        leaf service-id {
            type string;
            description
            "A unique slice service identifier.";
        }
        leaf service-description {
            type string;
            description
            "Textual description of the slice service.";
        }
        container service-tags {
            description
            "Container for the list of service tags.";
            list tag-type {
                key "tag-type";
                description
            }
        }
    }
}

description
"Standard SLO and SLE template to be used.";
leaf tag-type {
    type identityref {
        base service-tag-type;
    }
    description "Slice service tag type.";
}
leaf-list value {
    type string;
    description "The tag value";
}
list tag-opaque {
    key "tag-name";
    description "The service tag opaquelist.";
    leaf tag-name {
        type string;
        description "The opaque tag name";
    }
    leaf-list value {
        type string;
        description "The opaque tag value";
    }
}
}
uses service-slo-sle-policy;
uses vpn-common:service-status;
container sdps {
    description "Slice service SDPs.";
    list sdp {
        key "sdp-id";
        uses sdp;
        description "List of SDPs in this slice service.";
    }
}
container connection-groups {
    description "Contains connections group.";
    list connection-group {
        key "connection-group-id";
        description
7. 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-service/network-slice-services/slice-service

The entries in the list above include the whole network configurations corresponding with the slice service 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.

8. 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:
This document requests to register a YANG module in the YANG Module Names registry [RFC7950].

- Name: ietf-network-slice-service
- Prefix: ietf-nss
- Reference: RFC XXXX

9. Acknowledgments

The authors wish to thank Mohamed Boucadair, Kenichi Ogaki, Sergio Belotti, Qin Wu, Susan Hares, Eric Grey, Daniele Ceccarelli, and many others for their helpful comments and suggestions.

10. Contributors

The following authors contributed significantly to this document:

Luis M. Contreras
Telefonica
Spain

Email: luismiguel.contrerasmurillo@telefonica.com

John Mullooly
Cisco
Email:jmullool@cisco.com

11. References

11.1. Normative References


Wu, et al. Expires 12 January 2023
11.2. Informative References

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

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

[I-D.liu-teas-transport-network-slice-yang]


Appendix A. Examples of Slice Services with Different Connection Types

A.1. Example-1: Two Any-to-any Slice Services with different match approach

The following example describes a simplified service configuration of two IETF Network slice instances where the SDPs are the customer-facing ports on the PE:

* IETF Network Slice 1 on SDP1, SDP11a, and SDP4, with any-to-any connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with unmanaged CE elements.

* IETF Network Slice 2 on SDP2, SDP11b, with any-to-any connectivity type. This is a L3 slice service and using the uniform high-BW slo-sle-template policy between all SDPs. These SDPs will also have AC eBGP peering sessions with unmanaged CE elements.

Slice 1 uses the explicit match approach for mapping SDP traffic to a connectivity-construct, while slice 2 uses the implicit approach. Both approaches are supported.

Note: These two slices both use service-tags of "L3". This service-tag is operator defined and has no specific meaning in the YANG model other to give a "hint" to the NSC on the Service Expectation being L3 forwarding. Other examples we may choose to eliminate it. The usage of this tag is arbitrary and up to the operator and the NSC on it’s need and usage.
{
    "data": {
      "ietf-network-slice-service:network-slice-services": {
        "slo-sle-templates": {
          "slo-sle-template": [
            {
              "id": "high-BW-template",
              "template-description": "take the highest BW forwarding path"
            },
            {
              "id": "low-latency-template",
              "template-description": "lowest possible latency forwarding behavior"
            }
          ]
        },
        "slice-service": [
          {
            "service-id": "slice1",
            "service-description": "example slice1",
            "service-tags": {
              "tag-type": [
                {
                  "tag-type": "ietf-nss:service-tag-service",
                  "value": ["L3"]
                }
              ]
            },
            "slo-sle-template": "low-latency-template",
          }
        ]
      }
    }
  }
"status": {},
"sdps": {
  "sdp": [
  {
    "sdp-id": "1",
    "node-id": "PE-A",
    "attachment-circuits": {
      "attachment-circuit": [
      {
        "ac-id": "ac1",
        "ac-description": "AC1 connected to device 1",
        "ac-node-id": "PE-A",
        "ac-tp-id": "GigabitEthernet5/0/0/0.100",
        "ac-ip-address": "192.0.2.1",
        "ac-ip-prefix-length": 24,
        "ac-tags": {
          "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          },
          ]
        },
      },
      "service-match-criteria": {
        "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-network-slice-service:service-any-match",
          "target-connection-group-id": "matrix1",
          "target-connectivity-construct-id": 1
        },
        ]
      },
      "sdp-peering": {
        "protocol": [
        {
          "protocol-type": "ietf-nss:peering-protocol-bgp",
          "attribute": [
          {
            "attribute-type": "ietf-nss:neighbor",
            "value": ["192.0.2.2"]
          },
          {
            "attribute-type": "ietf-nss:remote-as",
            "value": ["64001"]
          },
          ]
        },
        ]
      },
  },
  ]
},
}]
"attribute-type": "ietf-nss:local-as",
"value": ["64000"]
}
]
]
]
}
"status": {
}
}
"sdp-id": "3a",
"node-id": "PE-B",
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "ac3a",
      "ac-description": "AC3a connected to device 3",
      "ac-node-id": "PE-B",
      "ac-tp-id": "GigabitEthernet8/0/0/4.101",
      "ac-ip-address": "192.0.3.1",
      "ac-ip-prefix-length": 24,
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["101"]
          }
        ]
      },
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-network-slice-service:service-any-match",
            "target-connection-group-id": "matrix1",
            "target-connectivity-construct-id": 1
          }
        ]
      },
      "sdp-peering": {
        "protocol": [
          {
            "protocol-type": "ietf-nss:peering-protocol-bgp",
            "attribute": ["attribute-type": "ietf-nss:local-as",
"value": ["64000"]
}]}]}}]}},
{  
  "attribute-type": "ietf-nss:neighbor",
  "value": "[192.0.3.2]"
},
{  
  "attribute-type": "ietf-nss:remote-as",
  "value": "[64002]"
},
{  
  "attribute-type": "ietf-nss:local-as",
  "value": "[64000]"
}
]
"status": {
}
},

"sdp-id": "4",
"node-id": "PE-C",
"attachment-circuits": {
  "attachment-circuit": [
  {
    "ac-id": "ac4",
    "ac-description": "AC4 connected to device 4",
    "ac-node-id": "PE-C",
    "ac-tp-id": "GigabitEthernet4/0/0/3.100",
    "ac-ip-address": "192.0.4.1",
    "ac-ip-prefix-length": 24,
    "ac-tags": {
      "ac-tags": [
      {
        "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
        "value": "[100]"
      }
      ],
    },
    "service-match-criteria": {
      "match-criterion": [
      {
        "index": 1,
        "match-type": "ietf-network-slice-service:service-any-match",
        "target-connection-group-id": "matrix1",
      }
      ]
    }
  }
  ]
}
"target-connectivity-construct-id": 1
}
},
"sdp-peering": {
  "protocol": [
    {
      "protocol-type": "ietf-nss:peering-protocol-bgp",
      "attribute": [
        {
          "attribute-type": "ietf-nss:neighbor",
          "value": ["192.0.4.2"]
        },
        {
          "attribute-type": "ietf-nss:remote-as",
          "value": ["64004"]
        },
        {
          "attribute-type": "ietf-nss:local-as",
          "value": ["64000"]
        }
      ]
    }
  ]
},
"status": {
}
},
"connection-groups": {
  "connection-group": [
    {
      "connection-group-id": "matrix1",
      "connectivity-type": "ietf-vpn-common:any-to-any",
      "connectivity-construct": [
        {
          "cc-id": 1,
          "a2a-sdp": [
            {
              "sdp-id": "1"
            },
            {
              "sdp-id": "3a"
            }
          ]
        }
      ]
    }
  ]
},
{  "service-id": "slice2",
   "service-description": "example slice2",
   "service-tags": {  
   "tag-type": [  
   {  
   "tag-type": "ietf-nss:service-tag-service",
   "value": ["L3"]  
   }  
   ]  
   },  
   "slo-sle-template": "high-BW-template",
   "status": {  
   },  
   "sdps": {  
   "sdp": [  
   {  
   "sdp-id": "2",
   "node-id": "PE-A",
   "attachment-circuits": {  
   "attachment-circuit": [  
   {  
   "ac-id": "ac2",
   "ac-description": "AC2 connected to device 2",
   "ac-node-id": "PE-A",
   "ac-tp-id": "GigabitEthernet7/0/0/3.200",
   "ac-ip-address": "198.51.100.1",
   "ac-ip-prefix-length": 24,
   "ac-tags": {  
   "ac-tags": [  
   {  
   "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
   "value": ["100"]  
   }  
   ]  
   },  
   "sdp-peering": {  
   "protocol": [  
   
{  
  "protocol-type": "ietf-nss:peering-protocol-bgp",
  "attribute": [
    
    {
      "attribute-type": "ietf-nss:neighbor",
      "value": ["192.51.100.2"]
    },
    
    {
      "attribute-type": "ietf-nss:remote-as",
      "value": ["64031"]
    },
    
    {
      "attribute-type": "ietf-nss:local-as",
      "value": ["64000"]
    }
  ]
}

"status": {
}

},

{"sdp-id": "3b",
 "node-id": "PE-B",
 "attachment-circuits": {
  "attachment-circuit": [
    
    {
      "ac-id": "ac3b",
      "ac-description": "AC3b connected to device 3",
      "ac-node-id": "PE-B",
      "ac-tp-id": "GigabitEthernet8/0/0/4.201",
      "ac-ip-address": "198.51.101.1",
      "ac-ip-prefix-length": 24,
      "ac-tags": {
        "ac-tags": [  
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["201"]
          }
        ]
      }
    },
    
    "sdp-peering": {
      "protocol": [
        
      ]
    }
  ]
}

"protocol-type": "ietf-nss:peering-protocol-bgp",
"attribute": [
{
"attribute-type": "ietf-nss:neighbor",
"value": ["192.51.101.2"]
},
{
"attribute-type": "ietf-nss:remote-as",
"value": ["64032"]
},
{
"attribute-type": "ietf-nss:local-as",
"value": ["64000"]
}
]
]
],
"status": {
}
}
},
"connection-groups": {
"connection-group": [
{
"connection-group-id": "matrix2",
"connectivity-type": "ietf-vpn-common:any-to-any",
"connectivity-construct": [
{
"cc-id": 1,
"a2a-sdp": [
{
"sdp-id": "2"
},
{
"sdp-id": "3b"
}
]
}
]
}
]
A.2. Example-2: Two P2P slice services with different match approaches

The following example describes a simplified service configuration of two IETF Network slice instances where the SDPs are the customer-facing ports on the PE:

* IETF Network Slice 3 on SDP5 and SDP7a with pt-to-pt connectivity type. This is a L2 slice service and using the uniform low-latency slo-sle-template policies between the SDPs.

* IETF Network Slice 4 on SDP6 and SDP7b, with pt-to-pt connectivity type. This is a L2 slice service and using the a high-BW slo-sle-template policies between the SDPs. Traffic from SDP6 and SDP7b is requesting a BW of 1000Mbps, while in the reverse direction from SDP7b to SDP6, 5000Mbps is being requested.

Slice 3 uses the explicit match approach for mapping SDP traffic to a Connectivity-group, while slice 2 uses the implicit approach. Both approaches are supported.

Note: These two slices both use service-tags of "L2". This service-tag is operator defined and has no specific meaning in the YANG model other to give a "hint" to the NSC on the Service Expectation being L2 forwarding. Other examples we may choose to eliminate it. The usage of this tag is arbitrary and up to the operator and the NSC on it’s need and usage.
{ 
    "data": {
        "ietf-network-slice-service:network-slice-services": {
            "slo-sle-templates": {
                "slo-sle-template": [
                    {
                        "id": "high-BW-template",
                        "template-description": "take the highest BW forwarding path"
                    },
                    {
                        "id": "low-latency-template",
                        "template-description": "lowest possible latency forwarding behavior"
                    }
                ]
            },
            "slice-service": [
                {
                    "service-id": "slice3",
                    "service-description": "example slice3",
                    "slo-sle-template": "low-latency-template",
                    "status": {
                    },
                    "sdps": {
                        "sdp": [
                            {
                                "sdp-id": "5",
                                "node-id": "PE-A",
                                "attachment-circuits": {
                                    "attachment-circuit": [
                                    ]
                                }
                            }
                        ]
                    }
                }
            ]
        }
    }
}
"ac-id": "ac5",
"ac-description": "AC5 connected to device 5",
"ac-node-id": "PE-A",
"ac-tp-id": "GigabitEthernet5/0/0/1",
"ac-tags": {
  "ac-tags": [
    {
      "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
      "value": ["100"]
    }
  ],
},
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-any-match",
      "target-connection-group-id": "matrix3"
    }
  ]
},
"status": {
},
"sdp-id": "7a",
"node-id": "PE-B",
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "ac7a",
      "ac-description": "AC7a connected to device 7",
      "ac-node-id": "PE-B",
      "ac-tp-id": "GigabitEthernet8/0/0/5",
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["200"]
          }
        ]
      },
      "service-match-criteria": {
        "match-criterion": [
          {
        
```
"index": 1,
"match-type": "ietf-network-slice-service:service-any-match",
"target-connection-group-id": "matrix3"
],
"status": {
}
},
"connection-groups": {
"connection-group": [
{
"connection-group-id": "matrix3",
"connectivity-type": "ietf-network-slice-service:point-to-point",
"connectivity-construct": [
{
"cc-id": 1,
"p2p-sender-sdp": "5",
"p2p-receiver-sdp": "7a"
},
{
"cc-id": 2,
"p2p-sender-sdp": "7a",
"p2p-receiver-sdp": "5"
}
]
},
"service-id": "slice4",
"service-description": "example slice4",
"slo-sle-template": "high-BW-template",
"status": {
},
"sdps": {
"sdp": [
{ "sdp-id": "6",
"node-id": "PE-A",
"attachment-circuits": {
"attachment-circuit": [
}]]
]}}
}
{
    "ac-id": "ac6",
    "ac-description": "AC6 connected to device 6",
    "ac-node-id": "PE-A",
    "ac-tp-id": "GigabitEthernet7/0/0/4",
    "ac-tags": {
        "ac-tags": [
            {
                "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
                "value": ["101"]
            }
        ]
    }
},
"status": {
},
"sdp-id": "7b",
"node-id": "PE-B",
"attachment-circuits": {
    "attachment-circuit": [
        {
            "ac-id": "ac7b",
            "ac-description": "AC7b connected to device 7",
            "ac-node-id": "PE-B",
            "ac-tp-id": "GigabitEthernet8/0/0/5",
            "ac-tags": {
                "ac-tags": [
                    {
                        "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
                        "value": ["201"]
                    }
                ]
            }
        }
    ]
},
"status": {
},
"connection-groups": {
    "connection-group": [
    
"connection-group-id": "matrix4",
"connectivity-type": "ietf-network-slice-service:point-to-point",
"connectivity-construct": [
  {
    "cc-id": 1,
    "p2p-sender-sdp": "6",
    "p2p-receiver-sdp": "7b",
    "service-slo-sle-policy": {
      "metric-bounds": {
        "metric-bound": [
          {
            "metric-type": "ietf-nss:service-slo-one-way-bandwidth",
            "metric-unit": "Mbps",
            "value-description": "1000"
          }
        ]
      }
    }
  },
  {
    "cc-id": 2,
    "p2p-sender-sdp": "7b",
    "p2p-receiver-sdp": "6",
    "service-slo-sle-policy": {
      "metric-bounds": {
        "metric-bound": [
          {
            "metric-type": "ietf-nss:service-slo-one-way-bandwidth",
            "metric-unit": "Mbps",
            "value-description": "5000"
          }
        ]
      }
    }
  }
]
A.3. Example-3: A Hub-and-spoke Slice Service with a P2MP connectivity construct

The following example describes a simplified service configuration of one IETF Network slice instance where the SDPs are the customer-facing ports on the PE:

IETF Network Slice 5 is a hub-spoke slice with SDP14 as the hub and SDP11, SDP12, SDP13a, SDP13b as spokes. This is a L3 slice service and using the uniform low-latency slo-sle-template policies between all spokes and the hub SDP, but using an explicit set of slo policies with a latency metric of 10ms for hub to spoke traffic. There is no peering protocol’s configured in this example.

```
+--------+ 196.0.2.1
|Device11o------/  VLAN100
+--------+      |  SDP11--------+
+--------+      +------o|  A +----------------+
|Device12o------/-------o|      |               |
+--------+         SDP12--------+
                         196.51.100.1
                      VLAN200       ++---+ SDP14 +--------+
                          196.0.4.1
                          VLAN100
                          |   C o-----/-----oDevice14|
+--------+        ++---+ SDP13a+-+--+               |
|        o------/ VLAN101
|        |      | SDP13b++------+
Device13 |      +------o|  B +---------------|
o--------/-/------o
+--------+        196.51.101.1
                            VLAN201

{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latency forwarding behavior"
          }
        ]
      }
    }
  }
}
```

"slice-service": [  {
    "service-id": "slice5",
    "service-description": "example slice5",
    "service-tags": {
      "tag-type": [  {
        "tag-type": "ietf-nss:service-tag-service",
        "value": ["L3"]
      }],
    },
    "slo-sle-template": "low-latency-template",
    "status": {
    },
    "sdps": {
      "sdp": [  {
        "sdp-id": "11",
        "node-id": "PE-A",
        "attachment-circuits": {
          "attachment-circuit": [  {
            "ac-id": "ac11",
            "ac-description": "AC11 connected to device 11",
            "ac-node-id": "PE-A",
            "ac-tp-id": "GigabitEthernet5/0/0/2",
            "ac-ip-address": "196.0.2.1",
            "ac-ip-prefix-length": 24,
            "ac-tags": [  {
              "ac-tags": [  {
                "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-
                        id",
                "value": ["100"]
              }
            ],
            "service-match-criteria": [  {
              "match-criterion": [  {
                "index": 1,
                "match-type": "ietf-network-slice-service:service-any
                        -match",
                "target-connection-group-id": "matrix5",
                "connection-group-sdp-role": "ietf-vpn-common:spoke-r
                        ole"
              }
            ]
          }
        ]
      }
    }]
  }]}
{ "ac-id": "ac13a", "ac-description": "AC13a connected to device 13", "ac-node-id": "PE-B", "ac-tp-id": "GigabitEthernet8/0/0/6", "ac-ip-address": "196.0.3.1", "ac-ip-prefix-length": 24, "ac-tags": { "ac-tags": [ { "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id", "value": ["101"] } ] }, "service-match-criteria": { "match-criterion": [ { "index": 1, "match-type": "ietf-network-slice-service:service-any-match", "target-connection-group-id": "matrix5", "connection-group-sdp-role": "ietf-vpn-common:spoke-role" } ] } }, "status": { }, "sdp-id": "13b", "node-id": "PE-B", "attachment-circuits": { "attachment-circuit": [ { "ac-id": "ac13b", "ac-description": "AC3b connected to device 13", "ac-node-id": "PE-B", "ac-tp-id": "GigabitEthernet8/0/0/4", "ac-ip-address": "196.51.101.1", "ac-ip-prefix-length": 24, "ac-tags": { "ac-tags": [ { "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id", "value": ["201"] } ] } ] } ]}
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-any-match",
      "target-connection-group-id": "matrix5",
      "connection-group-sdp-role": "ietf-vpn-common:spoke-role"
    }
  ]
}
"status": {
}
"sdp-id": "14",
"node-id": "PE-C",
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "ac14",
      "ac-description": "AC14 connected to device 14",
      "ac-node-id": "PE-C",
      "ac-tp-id": "GigabitEthernet4/0/0/3",
      "ac-ip-address": "196.0.4.1",
      "ac-ip-prefix-length": 24,
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          }
        ]
      }
    },
    "service-match-criteria": {
      "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-network-slice-service:service-any-match",
          "target-connection-group-id": "matrix5",
          "connection-group-sdp-role": "ietf-vpn-common:hub-role"
        }
      ]
    }
  ]
}
"status": {
},
"connection-groups": {
"connection-group": [
{
"connection-group-id": "matrix5",
"connectivity-type": "ietf-vpn-common:hub-spoke",
"connectivity-construct": [
{
"cc-id": 1,
"p2mp-sender-sdp": "14",
"p2mp-receiver-sdp": ["11", "12", "13a", "13b"],
"service-slo-sle-policy": {
"metric-bounds": {
"metric-bound": [
{
"metric-type": "ietf-nss:service-slo-one-way-delay",
"metric-unit": "milliseconds",
"value-description": "10"
}
]
}
},
{
"cc-id": 2,
"p2p-sender-sdp": "11",
"p2p-receiver-sdp": "14"
},
{
"cc-id": 3,
"p2p-sender-sdp": "12",
"p2p-receiver-sdp": "14"
},
{
"cc-id": 4,
"p2p-sender-sdp": "13a",
"p2p-receiver-sdp": "14"
},
{
"cc-id": 5,
A.4. Example 4: An Any-to-any Slice service with multiple SLOs and DSCP Matching

The following example describes a simplified service configuration of an IETF Network slice instance where the SDPs are the customer-facing ports on the PE:

IETF Network Slice 6 on SDP21, SDP23a, and SDP24, with any-to-any connectivity type. This is a L3 slice service and using the uniform standard slo-sle-template policies between all SDPs. For traffic matching the dscp of EF, a slo-sle-template policy of low-latency will be used. The slice uses the explicit match approach for mapping SDP traffic to a connectivity-construct.
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "template-description": "lowest possible latency forwarding behavior"
          },
          {
            "id": "standard-template",
            "template-description": "take the standard forwarding path"
          }
        ]
      },
      "slice-service": [
        {
          "service-id": "slice6",
          "service-description": "example slice6",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          }
        },
        {
          "slo-sle-template": "standard-template",
          "status": {
            "sdps": {
              "sdp": [
                {
                  "sdp-id": "21",
                  "node-id": "PE-A",
                  "attachment-circuits": {
                    "attachment-circuit": [
                      {
                        "ac-id": "ac21",
                        "ac-description": "AC21 connected to device 21",
                        "ac-node-id": "PE-A",
                        "ac-tp-id": "GigabitEthernet5/0/0/0",
                        "ac-ip-address": "194.0.2.1",
                        "ac-ip-prefix-length": 24,
                        "ac-node-id": "PE-A",
                        "ac-tp-id": "GigabitEthernet5/0/0/0",
                        "ac-ip-address": "194.0.2.1",
                        "ac-ip-prefix-length": 24,
                      }
                    ]
                  }
                }
              ]
            }
          }
        }
      ]
    }
  }
}
"ac-tags": {
  "ac-tags": [
    {
      "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
      "value": ["100"]
    }
  ]
},
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-dscp-match",
      "value": ["EF"],
      "target-connection-group-id": "matrix6",
      "target-connectivity-construct-id": 2
    },
    {
      "index": 2,
      "match-type": "ietf-network-slice-service:service-any-match",
      "target-connection-group-id": "matrix6",
      "target-connectivity-construct-id": 1
    }
  ]
},
"status": {
},
"sdp-id": "23a",
"node-id": "PE-B",
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "ac23a",
      "ac-description": "AC23a connected to device 23",
      "ac-node-id": "PE-B",
      "ac-tp-id": "GigabitEthernet8/0/0/4",
      "ac-ip-address": "194.0.3.1",
      "ac-ip-prefix-length": 24,
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["101"]
          }
        ]
      }
    }
  ]
}
"service-match-criteria": [  
  {  
    "index": 1,
    "match-type": "ietf-network-slice-service:service-dscp-p-match",
    "value": ["EF"],
    "target-connection-group-id": "matrix6",
    "target-connectivity-construct-id": 2
  },  
  {  
    "index": 2,
    "match-type": "ietf-network-slice-service:service-any-match",
    "target-connection-group-id": "matrix6",
    "target-connectivity-construct-id": 1
  }
]  
},  
"status": {  
},  
"sdp-id": "24",
"node-id": "PE-C",
"attachment-circuits": {  
  "attachment-circuit": [  
    {  
      "ac-id": "ac24",
      "ac-description": "AC24 connected to device 24",
      "ac-node-id": "PE-C",
      "ac-tp-id": "GigabitEthernet4/0/0/3",
      "ac-ip-address": "194.0.4.1",
      "ac-ip-prefix-length": 24,
      "ac-tags": {  
        "ac-tags": [  
          {  
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          }
        ]
      }
    }
  ]
},
"service-match-criteria": {  
  "match-criterion": [

{  
  "index": 1,  
  "match-type": "ietf-network-slice-service:service-dscp-match",
  "value": ["EF"],
  "target-connection-group-id": "matrix6",
  "target-connectivity-construct-id": 2
},
{  
  "index": 2,  
  "match-type": "ietf-network-slice-service:service-any-match",
  "target-connection-group-id": "matrix6",
  "target-connectivity-construct-id": 1
}
},
"status": {
}
},
"connection-groups": {
  "connection-group": [
  {
    "connection-group-id": "matrix6",
    "connectivity-type": "ietf-vpn-common:any-to-any",
    "connectivity-construct": [
    {
      "cc-id": 1,
      "a2a-sdp": [
      {
        "sdp-id": "21"
      },
      {
        "sdp-id": "23a"
      },
      {
        "sdp-id": "24"
      }
    }
  },
  {
    "cc-id": 2,
    "a2a-sdp": [
    {
      "sdp-id": "21"
    }
  }
  ]
}
A.5. Example-5: Two methods for an any-to-any Network Slice Service with multiple SLOs

The following examples describes a simplified service configuration of an IETF Network slice instance ‘NS1’ with four SDPs: SDP1, SDP2, SDP3 and SDP4 with any-to-any connectivity type. All SDPs are designated as customer-facing ports on the PE.
A.5.1. The method of a Single A2A connectivity construct

The service is realized using a single any-to-any connectivity construct, and a uniform low-bandwidth slo-sle-template policy applied to SDP2, SDP3, while a high-bandwidth slo-sle-template policy applied to SDP1 and SDP2.

```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "template-description": "take the highest BW forwarding path"
          },
          {
            "id": "low-BW-template",
            "template-description": "lowest BW forwarding behavior"
          }
        ]
      }
    }
  },
  "slice-service": [
    {
      "service-id": "NS1",
      "service-description": "URLLC",
      "service-tags": {
        "tag-type": [
          {
            "tag-type": "ietf-nss:service-tag-customer",
            "value": ["Customer-FOO"]
          },
          {
            "tag-type": "ietf-nss:service-tag-service",
            "value": ["L3"]
          }
        ]
      }
    },
    "status": {
    },
    "sdps": {
      "sdp": [
        {
          "sdp-id": "SDP1",
          "sdp-description": "Central Office 1 at location PE-A",
          "node-id": "PE-A",
          "sdp-ip": "192.0.1.1",
          "service-match-criteria": {
```
"match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-vlan-match",
      "value": ["100"],
      "target-connection-group-id": "matrix1"
    }
  ],
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "AC-SDP1",
      "ac-description": "Device 1 to PE-A",
      "ac-node-id": "PE-A",
      "ac-tp-id": "GigabitEthernet1/0/0/0",
      "ac-ip-address": "192.0.1.1",
      "ac-ip-prefix-length": 24,
      "ac-qos-policy-name": "QoS-Gold",
      "ac-tags": {
        "ac-tags": [ {
          "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
          "value": ["100"]
        }
      ],
      "ac-tag-opaque": [ {
        "tag-name": "VRF",
        "value": ["FOO"]
      }
      ],
      "sdp-peering": {
        "protocol": [ {
          "protocol-type": "ietf-nss:peering-protocol-bgp",
          "attribute": [ {
            "attribute-type": "ietf-nss:neighbor",
            "value": ["192.0.1.2"]
          }
          ]
        }
        ],
        "opaque": [ {
          "attribute-name": "color",
          "value": ["FOO"]
        }
        ]
      }
    }
  ]
}
"value": ["10"]
]
},
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
]
},
"status": {
  
},
{"sdp-id": "SDP2",
"sdp-description": "Central Office 2 at location PE-B",
"node-id": "PE-B",
"sdp-ip": "192.0.2.1",
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-vlan-match",
      "value": ["100"],
      "target-connection-group-id": "matrix1"
```
"ac-tag-opaque": [
  {
    "tag-name": "VRF",
    "value": ["FOO"]
  }
],
"sdp-peering": {
  "protocol": [
    {
      "protocol-type": "ietf-nss:peering-protocol-bgp",
      "attribute": [
        {
          "attribute-type": "ietf-nss:neighbor",
          "value": ["192.0.2.2"]
        }
      ]
    }
  ],
  "opaque": [
    {
      "attribute-name": "color",
      "value": ["10"]
    }
  ],
  "incoming-rate-limits": {
    "cir": "1000000",
    "cbs": "1000",
    "pir": "5000000",
    "pbs": "1000"
  }
},
"status": {
},
"sdp-id": "SDP3",
"sdp-description": "Remote Office 1 at location PE-C",
"node-id": "PE-C",
"sdp-ip": "192.0.3.1",
"service-match-criteria": {
  "match-criterion": {
    "index": 1,
    "match-type": "ietf-network-slice-service:service-vlan-match"...
  }
"value": ["100"],
"target-connection-group-id": "matrix1"
}
},
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "AC-SDP3",
      "ac-description": "Device 3 to PE-C",
      "ac-node-id": "PE-C",
      "ac-tp-id": "GigabitEthernet3/0/0/0",
      "ac-ip-address": "192.0.3.1",
      "ac-ip-prefix-length": 24,
      "ac-qos-policy-name": "QoS-Gold",
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          }
        ],
        "ac-tag-opaque": [
          {
            "tag-name": "VRF",
            "value": ["FOO"]
          }
        ]
      },
      "sdp-peering": {
        "protocol": [
          {
            "protocol-type": "ietf-nss:peering-protocol-bgp",
            "attribute": [
              {
                "attribute-type": "ietf-nss:neighbor",
                "value": ["192.0.3.2"]
              }
            ]
          }
        ],
        "opaque": [
          {
            "attribute-name": "color",
            "value": ["10"]
          }
        ]
      }
    }
  ]
}
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
},

"status": {
}
},

"sdp-id": "SDP4",
"sdp-description": "Remote Office 2 at location PE-D",
"node-id": "PE-D",
"sdp-ip": "192.0.4.1",
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-vlan-matching",
      "value": ["100"],
      "target-connection-group-id": "matrix1"
    }
  ]
},

"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "AC-SDP4",
      "ac-description": "Device 4 to PE-D",
      "ac-node-id": "PE-A",
      "ac-tp-id": "GigabitEthernet4/0/0/0",
      "ac-ip-address": "192.0.4.1",
      "ac-ip-prefix-length": 24,
      "ac-qos-policy-name": "QoS-Gold",
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          }
        ],
        "ac-tag-opaque": [
          {
            "tag-name": "VRF",
            "value": ["FOO"]
          }
        ]
      }
    }
  ]
}
A.5.2. The method of Two A2A connectivity construct

The service is realized using two any-to-any connectivity constructs:

* any-to-any cc-id=1 enables SDP1 and SDP2 to communicate using the high-bandwidth slo-sle-template policy between SDPs.

* any-to-any cc-id=2 enables all SDPs to communicate using the low-bandwidth slo-sle-template policy between SDPs.

```json
{
   "data": {
      "ietf-network-slice-service:network-slice-services": {
         "slo-sle-templates": {
            "slo-sle-template": [
               {
                  "id": "high-BW-template",
                  "template-description": "take the highest BW forwarding path"
               },
               {
                  "id": "low-BW-template",
                  "template-description": "lowest BW forwarding behavior"
               }
            ]
         }
      }
   }
}
```
"slice-service": [  
  {  
    "service-id": "NS1",  
    "service-description": "URLLC",  
    "service-tags": {  
      "tag-type": [  
        {  
          "tag-type": "ietf-nss:service-tag-customer",  
          "value": ["Customer-FOO"]  
        },  
        {  
          "tag-type": "ietf-nss:service-tag-service",  
          "value": ["L3"]  
        }  
      ]  
    },  
    "status": {  
    },  
    "sdps": {  
      "sdp": [  
        {  
          "sdp-id": "SDP1",  
          "sdp-description": "Central Office 1 at location PE-A",  
          "node-id": "PE-A",  
          "sdp-ip": "192.0.1.1",  
          "service-match-criteria": {  
            "match-criterion": [  
              {  
                "index": 1,  
                "match-type": "ietf-network-slice-service:service-vlan-match",  
                "value": ["100"],  
                "target-connection-group-id": "matrix1"  
              }  
            ]  
          },  
          "attachment-circuits": {  
            "attachment-circuit": [  
              {  
                "ac-id": "AC-SDP1",  
                "ac-description": "Device 1 to PE-A",  
                "ac-node-id": "PE-A",  
                "ac-tp-id": "GigabitEthernet1/0/0/0",  
                "ac-ip-address": "192.0.1.1",  
                "ac-ip-prefix-length": 24,  
                "ac-qos-policy-name": "QoS-Gold",  
                "ac-tags": {  
                  "ac-tags": [  
                }  
              }  
            ]  
          }  
        }  
      ]  
    }  
  ]
"ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
    "value": ["100"]
  ],
"ac-tag-opaque": [ ...
"tag-name": "VRF",
    "value": ["FOO"]
  ]
},
"sdp-peering": {
  "protocol": [ ...
    "protocol-type": "ietf-nss:peering-protocol-bgp",
    "attribute": [ ...
      "attribute-type": "ietf-nss:neighbor",
      "value": ["192.0.1.2"]
    ]
  ],
"opaque": [ ...
    "attribute-name": "color",
    "value": ["10"]
  ]
},
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
}
,"status": {
  ...}
},
"sdp-id": "SDP2",
"sdp-description": "Central Office 2 at location PE-B",
"node-id": "PE-B",
"sdp-ip": "192.0.2.1",
"service-match-criteria": {

"match-criterion": [ 
  
  "index": 1, 
  "match-type": "ietf-network-slice-service:service-vlan-match", 
  "value": ["100"], 
  "target-connection-group-id": "matrix1"
  
]
],
"attachment-circuits": { 
  "attachment-circuit": [ 
    
    "ac-id": "AC-SDP2",
    "ac-description": "Device 2 to PE-B",
    "ac-node-id": "PE-B",
    "ac-tp-id": "GigabitEthernet2/0/0/0",
    "ac-ip-address": "192.0.2.1",
    "ac-ip-prefix-length": 24,
    "ac-qos-policy-name": "QoS-Gold",
    "ac-tags": { 
      "ac-tags": [ 
        
        "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
        "value": ["100"]
      ]
    }
  ],
  "ac-tag-opaque": [ 
    
    "tag-name": "VRF",
    "value": ["FOO"]
  ]
],
"sdp-peering": { 
  "protocol": [ 
    
    "protocol-type": "ietf-nss:peering-protocol-bgp",
    "attribute": [ 
      
      "attribute-type": "ietf-nss:neighbor",
      "value": ["192.0.2.2"]
    ]
  ]
],
"opaque": [ 
  
  "attribute-name": "color",
  "attribute-type": "opaque",
  "value": ["FOO"]
]
"value": ["10"]
}
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
"status": {
}
"sdp-id": "SDP3",
"sdp-description": "Remote Office 1 at location PE-C",
"node-id": "PE-C",
"sdp-ip": "192.0.3.1",
"service-match-criteria": {
  "match-criterion": [
    {
      "index": 1,
      "match-type": "ietf-network-slice-service:service-vlan-match",
      "value": ["100"],
      "target-connection-group-id": "matrix1"
    }
  ]
},
"attachment-circuits": {
  "attachment-circuit": [
    {
      "ac-id": "AC-SDP3",
      "ac-description": "Device 3 to PE-C",
      "ac-node-id": "PE-C",
      "ac-tp-id": "GigabitEthernet3/0/0/0",
      "ac-ip-address": "192.0.3.1",
      "ac-ip-prefix-length": 24,
      "ac-qos-policy-name": "QoS-Gold",
      "ac-tags": {
        "ac-tags": [
          {
            "ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
            "value": ["100"]
          }
        ]
      }
    }
  ]
}
"ac-tag-opaque": [
   {
      "tag-name": "VRF",
      "value": ["FOO"]
   }
],
"sdp-peering": {
   "protocol": [
      {
         "protocol-type": "ietf-nss:peering-protocol-bgp",
         "attribute": [
            {
               "attribute-type": "ietf-nss:neighbor",
               "value": ["192.0.3.2"]
            }
         ]
      }
   ],
   "opaque": [
      {
         "attribute-name": "color",
         "value": ["10"]
      }
   ],
   "incoming-rate-limits": {
      "cir": "1000000",
      "cbs": "1000",
      "pir": "5000000",
      "pbs": "1000"
   }
],
"status": { }
},
"sdp-id": "SDP4",
"sdp-description": "Remote Office 2 at location PE-D",
"node-id": "PE-D",
"sdp-ip": "192.0.4.1",
"service-match-criteria": {
   "match-criterion": [
      {
         "index": 1,
         "match-type": "ietf-network-slice-service:service-vlan-match"
"value": ["100"],
"target-connection-group-id": "matrix1"
}
},
"attachment-circuits": {
"attachment-circuit": [
{
"ac-id": "AC-SDP4",
"ac-description": "Device 4 to PE-D",
"ac-node-id": "PE-A",
"ac-tp-id": "GigabitEthernet4/0/0/0",
"ac-ip-address": "192.0.4.1",
"ac-ip-prefix-length": 24,
"ac-qos-policy-name": "QoS-Gold",
"ac-tags": {
"ac-tags": [
{
"ac-tag-type": "ietf-nss:attachment-circuit-tag-vlan-id",
"value": ["100"]
}
],
"ac-tag-opaque": [
{
"tag-name": "VRF",
"value": ["FOO"]
}
]
},
"sdp-peering": {
"protocol": [
{
"protocol-type": "ietf-nss:peering-protocol-bgp",
"attribute": [
{
"attribute-type": "ietf-nss:neighbor",
"value": ["192.0.4.2"]
}
]
}
],
"opaque": [
{
"attribute-name": "color",
"value": ["10"]
}
]
},
}
"incoming-rate-limits": {
  "cir": "1000000",
  "cbs": "1000",
  "pir": "5000000",
  "pbs": "1000"
}
},
"status": {
}
}
},
"connection-groups": {
"connection-group": [
{
  "connection-group-id": "matrix1",
  "connectivity-construct": [
  {
    "cc-id": 1,
    "a2a-sdp": [
    {
      "sdp-id": "SDP1"
    },
    {
      "sdp-id": "SDP2"
    }
  ],
  "slo-sle-template": "high-BW-template"
 },
{
  "cc-id": 2,
  "a2a-sdp": [
  {
    "sdp-id": "SDP1"
  },
  {
    "sdp-id": "SDP2"
  },
  {
    "sdp-id": "SDP3"
  },
  {
    "sdp-id": "SDP4"
  }
  ],
  "slo-sle-template": "low-BW-template"
Appendix B.  Comparison with Other Possible Design choices for IETF Network Slice Service Interface

According to the 5.3.1 IETF Network Slice Service Interface [I-D.ietf-teas-ietf-network-slices], the Network Slice service Interface is a technology-agnostic interface, which is used for a customer to express requirements for a particular IETF Network Slice. Customers operate on abstract IETF Network Slices, with details related to their realization hidden. As classified by [RFC8309], the Network Slice service Interface is classified as Customer Service Model.

This draft analyzes the following existing IETF models to identify the gap between the IETF Network Slice service Interface requirements.

B.1.  ACTN VN Model Augmentation

The difference between the ACTN VN model and the IETF Network Slice service requirements is that the IETF Network Slice service interface 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 customer 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 SDP 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 "service-match-criteria" to specify a specific SDP belonging to an IETF Network Slice service.

B.2. RFC8345 Augmentation Model

The difference between the IETF Network Slice service requirements and the IETF basic network model is that the IETF Network Slice service requests abstract customer 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:

* Network: a transport network list and an list of nodes contained in the network

* Link: "links" list and "termination points" list describe how nodes in a network are connected to each other

* 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.
Framework for IETF Network Slices
draft-ietf-teas-ietf-network-slices-08

Abstract

This document describes network slicing in the context of networks built from IETF technologies. It defines the term "IETF Network Slice" and establishes the general principles of network slicing in the IETF context.

The document discusses the general framework for requesting and operating IETF Network Slices, the characteristics of an IETF Network Slice, the necessary system components and interfaces, and how abstract requests can be mapped to more specific technologies. The document also discusses related considerations with monitoring and security.

This document also provides definitions of related terms to enable consistent usage in other IETF documents that describe or use aspects of IETF Network Slices.

Status of This Memo

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

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

1. Introduction .................................................. 3
  1.1. Background ................................................ 4
2. Terms and Abbreviations ....................................... 5
  2.1. Core Terminology ........................................... 5
3. IETF Network Slice Objectives .................................. 7
  3.1. Definition and Scope of IETF Network Slice ............... 7
  3.2. IETF Network Slice Service ............................... 8
    3.2.1. Ancillary SDPs ......................................... 11
4. IETF Network Slice System Characteristics .................... 12
  4.1. Objectives for IETF Network Slices ....................... 12
    4.1.1. Service Level Objectives ............................. 13
    4.1.2. Service Level Expectations ........................... 14
  4.2. IETF Network Slice Service Demarcation Points .......... 16
  4.3. IETF Network Slice Decomposition ......................... 19
5. Framework .................................................... 20
  5.1. IETF Network Slice Stakeholders ........................ 20
  5.2. Expressing Connectivity Intents .......................... 20
  5.3. IETF Network Slice Controller (NSC) ...................... 22
    5.3.1. IETF Network Slice Controller Interfaces ............ 24
    5.3.2. Management Architecture .............................. 25
  5.4. IETF Network Slice Structure ............................. 26
6. Realizing IETF Network Slices .................................. 27
  6.1. Architecture to Realize IETF Network Slices ............. 28
  6.2. Procedures to Realize IETF Network Slices ............... 31
  6.3. Applicability of ACTN to IETF Network Slices ............ 32
1. Introduction

A number of use cases benefit from network connections that, along with connectivity, provide assurance of meeting a specific set of objectives with respect to network resources use. This connectivity and resource commitment is referred to as a network slice and is expressed in terms of connectivity constructs (see Section 3) and service objectives (see Section 4). 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. 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 (known as Service Demarcation Points (SDPs) - see Section 2.1 and Section 4.2) over a shared underlay network. The term IETF Network Slice service is also introduced to describe the service requested by and provided to the service provider’s customer.

Services that might benefit from IETF Network Slices include, but are not limited to:

* 5G services (e.g. eMBB, URLLC, mMTC)(See [TS23501])
* Network wholesale services
* Network infrastructure sharing among operators
* NFV connectivity and Data Center Interconnect

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 with different requirements to coexist over a shared underlay network. A request for an IETF Network Slice service is agnostic to the technology in
the underlying network so as to allow a customer to describe their network connectivity objectives in a common format, independent of the underlying technologies used.

This document also provides a framework for discussing IETF Network Slices. The framework is intended as a structure for discussing interfaces and technologies. It is not intended to specify a new set of concrete interfaces or technologies.

For example, virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to support the VPNs is often referred to as an underlay network, and the VPN is often called an overlay network. An overlay network may, in turn, serve as an underlay network to support another overlay network.

Note that it is conceivable that extensions to IETF technologies are needed in order to fully support all the ideas that can be implemented with network slices. Evaluation of existing technologies, proposed extensions to existing protocols and interfaces, and the creation of new protocols or interfaces is outside the scope of this document.

1.1. Background

The concept of network slicing has gained traction driven largely by needs surfacing from 5G ([NGMN-NS-Concept], [TS23501], and [TS28530]). In [TS23501], a Network Slice is defined as "a logical network that provides specific network capabilities and network characteristics", and a Network Slice Instance is defined as "A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice." According to [TS28530], an end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). An IETF Network Slice provides the required connectivity between different entities in RAN and CN segments of an end-to-end network slice, with a specific performance commitment. For each end-to-end network slice, the topology and performance requirement on a customer's use of an IETF Network Slice can be very different, which requires the underlay network to have the capability of supporting multiple different IETF Network Slices.

While network slices are commonly discussed in the context of 5G, it is important to note that IETF Network Slices are a narrower concept with a broader usage profile, and focus primarily on particular network connectivity aspects. Other systems, including 5G
deployments, may use IETF Network Slices as a component to create entire systems and concatenated constructs that match their needs, including end-to-end connectivity.

An IETF Network Slice could span multiple technologies and multiple administrative domains. Depending on the IETF Network Slice customer’s requirements, an IETF Network Slice could be isolated from other, often concurrent IETF Network Slices in terms of data, control and management planes.

The customer expresses requirements for a particular IETF Network Slice service by specifying what is required rather than how the requirement is to be fulfilled. That is, the IETF Network Slice customer’s view of an IETF Network Slice is an abstract one.

Thus, there is a need to create logical network structures with required characteristics. The customer of such a logical network can require a degree of isolation and performance that previously might not have been satisfied by overlay VPNs. Additionally, the IETF Network Slice customer might ask for some level of control of their virtual networks, e.g., to customize the service paths in a network slice.

This document specifies definitions and a framework for the provision of an IETF Network Slice service. Section 6 briefly indicates some candidate technologies for realizing IETF Network Slices.

2. Terms and Abbreviations

The following abbreviations are used in this document.

* NSC: Network Slice Controller
* SLA: Service Level Agreement
* SLI: Service Level Indicator
* SLO: Service Level Objective

The meaning of these abbreviations is defined in greater details in the remainder of this document.

2.1. Core Terminology

The following terms are presented here to give context. Other terminology is defined in the remainder of this document.

Customer: A customer is the requester of an IETF Network Slice
service. Customers may request monitoring of SLOs. A customer may be an entity such as an enterprise network or a network operator, an individual working at such an entity, a private individual contracting for a service, or an application or software component. A customer may be an external party (classically a paying customer) or a division of a network operator that uses the service provided by another division of the same operator. Other terms that have been applied to the customer role are "client" and "consumer".

Provider: A provider is the organization that delivers an IETF Network Slice service. A provider is the network operator that controls the network resources used to construct the network slice (that is, the network that is sliced). The provider’s network maybe a physical network or may be a virtual network supplied by another service provider.

Customer Edge (CE): The customer device that provides connectivity to a service provider. Examples include routers, Ethernet switches, firewalls, 4G/5G RAN or Core nodes, application accelerators, server load balancers, HTTP header enrichment functions, and PEPs (Performance Enhancing Proxy). In some circumstances CEs are provided to the customer and managed by the provider.

Provider Edge (PE): The device within the provider network to which a CE is attached. A CE may be attached to multiple PEs, and multiple CEs may be attached to a given PE.

Attachment Circuit (AC): A channel connecting a CE and a PE over which packets that belong to an IETF Network Slice service are exchanged. An AC is, by definition, technology specific: that is, the AC defines how customer traffic is presented to the provider network. The customer and provider agree (through configuration) on which values in which combination of layer 2 and layer 3 header and payload fields within a packet identify to which (IETF Network Slice service, connectivity construct, and SLOs/SLEs) that packet is assigned. The customer and provider may agree on a per (IETF Network Slice service, connectivity construct, and SLOs/SLEs) basis to police or shape traffic on the AC in both the ingress (CE to PE) direction and egress (PE to CE) direction. This ensures that the traffic is within the capacity profile that is agreed in an IETF Network Slice service. Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider. As described in Section 4.2 the AC may be part of the IETF Network Slice service or may be external to it.
Service Demarcation Point (SDP): The point at which an IETF Network Slice service is delivered by a service provider to a customer. Depending on the service delivery model (see Section 4.2) this may be a CE or a PE, and could be a device, a software component, or in the case of network functions virtualization (for example), be an abstract function supported within the provider’s network. Each SDP must have a unique identifier (e.g., an IP address or MAC address) within a given IETF Network Slice service and may use the same identifier in multiple IETF Network Slice services.

An SDP may be abstracted as a Service Attachment Point (SAP) [I-D.ietf-opsawg-sap] for the purpose generalizing the concept across multiple service types and representing it in management and configuration systems.

Connectivity Construct: A set of SDPs together with a communication type that defines how traffic flows between the SDPs. An IETF Network Slice service is specified in terms of a set of SDPs, the associated connectivity constructs and the service objectives that the customer wishes to see fulfilled.

3. IETF Network Slice Objectives

IETF Network Slices are created to meet specific requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics. Creation of an IETF Network Slice is initiated by a management system or other application used to specify network-related conditions for particular traffic flows in response to an actual or logical IETF Network Slice service request.

Once created, these slices can be monitored, modified, deleted, and otherwise managed.

Applications and components will be able to use these IETF Network Slices to move packets between the specified end-points of the service in accordance with specified characteristics.

3.1. Definition and Scope of IETF Network Slice

An IETF Network Slice service enables connectivity between a set of Service Demarcation Points (SDPs) with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) over a common underlay network.

An IETF Network Slice combines the connectivity resource requirements and associated network capabilities such as bandwidth, latency, jitter, and network functions with other resource behaviors such as
compute and storage availability. The definition of an IETF Network Slice is independent of the connectivity and technologies used in the underlay network. This allows an IETF Network Slice service customer 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 [TS23501].

An IETF Network Slice service is agnostic to the technology of the underlay network, and its realization may be selected based upon multiple considerations including its service requirements and the capabilities of the underlay network.

The term "Slice" refers to a set of characteristics and behaviors that differentiate one type of user-traffic from another. An IETF Network Slice assumes that an underlay 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 the SLOs/SLEs to specific flows or to all of the traffic in the slice.

3.2. IETF Network Slice Service

A service provider delivers an IETF Network Slice service for a customer. The IETF Network Slice service is specified in terms of a set of SDPs, a set of one or more connectivity constructs between subsets of these SDPs, and a set of SLOs and SLEs for each SDP sending to each connectivity construct. A communication type (point-to-point (P2P), point-to-multipoint (P2MP), or any-to-any (A2A)) is specified for each connectivity construct. That is, in a given IETF Network Slice service there may be one or more connectivity constructs of the same or different type, each connectivity construct may be between a different subset of SDPs, for a given connectivity construct each sending SDP has its own set of SLOs and SLEs, and the SLOs and SLEs in each set may be different. Note that a service provider may decide how many connectivity constructs per IETF Network Slice service it wishes to support such that an IETF Network Slice service may be limited to one connectivity construct or may support many.

This approach results in the following possible connectivity constructs:
* For a P2P connectivity construct, there is one sending SDP and one receiving SDP. This construct is like a private wire or a tunnel. All traffic injected at the sending SDP is intended to be received by the receiving SDP. The SLOs and SLEs apply at the sender (and implicitly at the receiver).

* For a P2MP connectivity construct, there is only one sending SDP and more than one receiving SDP. This is like a P2MP tunnel or multi-access VLAN segment. All traffic from the sending SDP is intended to be received by all the receiving SDPs. There is one set of SLOs and SLEs that applies at the sending SDP (and implicitly at all receiving SDPs).

* With an A2A connectivity construct, any sending SDP may send to any one receiving SDP or any set of receiving SDPs in the construct. There is an implicit level of routing in this connectivity construct that is not present in the other connectivity constructs because the provider’s network must determine to which receiving SDPs to deliver each packet. This construct may be used to support P2P traffic between any pair of SDPs, or to support multicast or broadcast traffic from one SDP to a set of other SDPs. In the latter case, whether the service is delivered using multicast within the provider’s network or using "ingress replication" or some other means is out of scope of the specification of the service. A service provider may choose to support A2A constructs, but to limit the traffic to unicast.

The SLOs/SLEs in an A2A connectivity construct apply to individual sending SDPs regardless of the receiving SDPs, and there is no linkage between sender and receiver in the specification of the connectivity construct. A sending SDP may be "disappointed" if the receiver is over-subscribed. If a customer wants to be more specific about different behaviors from one SDP to another SDP, they should use P2P connectivity constructs.

A customer traffic flow may be unicast or multicast, and various network realizations are possible:

* Unicast traffic may be mapped to a P2P connectivity construct for direct delivery, or to an A2A connectivity construct for the service provider to perform routing to the destination SDP. It would be unusual to use a P2MP connectivity construct to deliver unicast traffic because all receiving SDPs would get a copy, but this can still be done if the receivers are capable of dropping the unwanted traffic.
* A bidirectional unicast service can be constructed by specifying two P2P connectivity constructs. An additional SLE may specify fate-sharing in this case.

* Multicast traffic may be mapped to a set of P2P connectivity constructs, a single P2MP connectivity construct, or a mixture of P2P and P2MP connectivity constructs. Multicast may also be supported by an A2A connectivity construct. The choice clearly influences how and where traffic is replicated in the network. With a P2MP or A2A connectivity construct, it is the operator’s choice whether to realize the construct with ingress replication, multicast in the core, P2MP tunnels, or hub-and-spoke. This choice should not change how the customer perceives the service.

* The concept of a multipoint-to-point (MP2P) service can be realized with multiple P2P connectivity constructs. Note that, in this case, the egress may simultaneously receive traffic from all ingresses. The SLOs at the sending SDPs must be set with this in mind because the provider’s network is not capable of coordinating the policing of traffic across multiple distinct source SDPs. It is assumed that the customer, requesting SLOs for the various P2P connectivity constructs, is aware of the capabilities of the receiving SDP. If the receiver receives more traffic than it can handle, it may drop some and introduce queuing delays.

* The concept of a multipoint-to-multipoint (MP2MP) service can best be realized using a set of P2MP connectivity constructs, but could be delivered over an A2A connectivity construct if each sender is using multicast. As with MP2P, the customer is assumed to be familiar with the capabilities of all receivers. A customer may wish to achieve an MP2MP service using a hub-and-spoke architecture where they control the hub: that is, the hub may be an SDP or an ancillary SDP (see Section 3.2.1) and the service may be achieved by using a set of P2P connectivity constructs to the hub, and a single P2MP connectivity construct from the hub.

From the above, it can be seen that the SLOs of the senders define the SLOs for the receivers on any connectivity construct. That is, and in particular, the network may be expected to handle the traffic volume from a sender to all destinations. This extends to all connectivity constructs in an IETF Network Slice service.

Note that the realization of an IETF Network Slice service does not need to map the connectivity constructs one-to-one onto underlying network constructs (such as tunnels, etc.). The service provided to the customer is distinct from how the provider decides to deliver that service.
If a CE has multiple attachment circuits to a PE within a given IETF Network Slice service and they are operating in single-active mode, then all traffic between the CE and its attached PEs transits a single attachment circuit; if they are operating in all-active mode, then traffic between the CE and its attached PEs is distributed across all of the active attachment circuits.

A given sending SDP may be part of multiple connectivity constructs within a single IETF Network Slice service, and the SDP may have different SLOs and SLEs for each connectivity construct to which it is sending. Note that a given sending SDP’s SLOs and SLEs for a given connectivity construct apply between it and each of the receiving SDPs for that connectivity construct.

An IETF Network Slice service provider may freely make a deployment choice as to whether to offer a 1:1 relationship between IETF Network Slice service and connectivity construct, or to support multiple connectivity constructs in a single IETF Network Slice service. In the former case, the provider might need to deliver multiple IETF Network Slice services to achieve the function of the second case.

It should be noted that per Section 9 of [RFC4364] an IETF Network Slice service customer may actually provide IETF Network Slice services to other customers in a mode sometimes referred to as "carrier’s carrier". In this case, the underlying IETF Network Slice service provider may be owned and operated by the same or a different provider network. As noted in Section 4.3, network slices may be composed hierarchically or serially.

Section 4.2 provides a description of endpoints in the context of IETF network slicing. These are known as Service Demarcation Points (SDPs). For a given IETF Network Slice service, the customer and provider agree, on a per-SDP basis which end of the attachment circuit provides the SDP (i.e., whether the attachment circuit is inside or outside the IETF Network Slice service). This determines whether the attachment circuit is subject to the set of SLOs and SLEs at the specific SDP.

3.2.1. Ancillary SDPs

It may be the case that the set of SDPs needs to be supplemented with additional senders or receivers. An additional sender could be, for example, an IPTV or DNS server either within the provider’s network or attached to it, while an extra receiver could be, for example, a node reachable via the Internet. This is modelled as a set of ancillary SDPs which supplement the other SDPs in one or more connectivity constructs, or which have their own connectivity constructs. Note that an ancillary SDP can either have a resolvable
address, e.g., an IP address or MAC address, or the SDP may be a placeholder, e.g., IPTV or DNS server, which is resolved within the provider’s network when the IETF Network Slice service is instantiated.

4. IETF Network Slice System Characteristics

The following subsections describe the characteristics of IETF Network Slices in addition to the list of SDPs, the connectivity constructs, and the technology of the ACs.

4.1. Objectives for IETF Network Slices

An IETF Network Slice service is defined in terms of quantifiable characteristics known as Service Level Objectives (SLOs) and unquantifiable characteristics known as Service Level Expectations (SLEs). SLOs are expressed in terms Service Level Indicators (SLIs), and together with the SLEs form the contractual agreement between service customer and service provider known as a Service Level Agreement (SLA).

The terms are defined as follows:

* 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 customer can determine whether the provider is meeting the SLOs by performing measurements on the traffic.

* A Service Level Expectation (SLE) is an expression of an unmeasurable service-related request that a customer of an IETF Network Slice makes of the provider. An SLE is distinct from an SLO because the customer may have little or no way of determining whether the SLE is being met, but they still contract with the provider for a service that meets the expectation.
A Service Level Agreement (SLA) is an explicit or implicit contract between the customer of an IETF Network Slice service and the provider of the slice. The SLA is expressed in terms of a set of SLOs and SLEs that are to be applied for a given connectivity construct between a sending SDP and the set of receiving SDPs, and may describe the extent to which divergence from individual SLOs and SLEs can be tolerated, and commercial terms as well as any consequences for violating these SLOs and SLEs.

4.1.1. Service Level Objectives

SLOs define a set of measurable network attributes and characteristics that describe an IETF Network Slice service. SLOs do not describe how an IETF Network Slice service is 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 service may include multiple connection constructs that associate sets of endpoints (SDPs). SLOs apply to a given connectivity construct and apply to a specific direction of traffic flow. That is, they apply to a specific sending SDP and the connection to specific receiving SDPs.

The SLOs are combined with Service Level Expectations in an SLA.

4.1.1.1. Some Common SLOs

SLOs can be described as ‘Directly Measurable Objectives’: they are always measurable. See Section 4.1.2 for the description of Service Level Expectations which are unmeasurable service-related requests sometimes known as ‘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’. Future specifications (such as IETF Network Slice service YANG models) may precisely define these SLOs, and other SLOs may be introduced as described in Section 4.1.1.2.

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). [RFC7679] discusses one-way metrics.

Maximum Permissible Delay Variation: Packet delay variation (PDV) as defined by [RFC3393], is 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 connectivity construct is available in accordance with all of the SLOs associated with it.

4.1.1.2. Other Service Level Objectives

Additional SLOs may be defined to provide additional description of the IETF Network Slice service that a customer requests. These would be specified in further documents.

If the IETF Network Slice 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).

4.1.2. Service Level Expectations

SLEs define a set of network attributes and characteristics that describe an IETF Network Slice service, but which are not directly measurable by the customer. Even though the delivery of an SLE cannot usually be determined by the customer, the SLEs form an important part of the contract between customer and provider.

Quite often, an SLE will imply some details of how an IETF Network Slice service is realized by the provider, although most aspects of the implementation in the underlying network layers remain a free choice for the provider.

SLEs may be seen as aspirational on the part of the customer, and they are expressed as behaviors that the provider is expected to apply to the network resources used to deliver the IETF Network Slice service. The SLEs are combined with SLOs in an SLA.
An IETF Network Slice service may include multiple connection constructs that associate sets of endpoints (SDPs). SLEs apply to a given connectivity construct and apply to specific directions of traffic flow. That is, they apply to a specific sending SDP and the connection to specific receiving SDPs. However, being more general in nature than SLOs, SLEs may commonly be applied to all connection constructs in an IETF Network Slice service.

4.1.2.1. Some Common SLEs

SLEs can be described as ‘Indirectly Measurable Objectives’: they are not generally directly measurable by the customer.

Security, geographic restrictions, maximum occupancy level, and isolation are example SLEs as follows.

Security: A customer may request that the provider applies encryption or other security techniques to traffic flowing between SDPs of a connectivity construct within an IETF Network Slice service. For example, the customer could request that only network links that have MACsec [MACsec] enabled are used to realize the connectivity construct.

This SLE may include a request for encryption (e.g., [RFC4303]) between the two SDPs explicitly to meet the architectural recommendations in [TS33.210] or for compliance with [HIPAA] or [PCI].

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Please see further discussion on security in Section 9.

Geographic Restrictions: A customer may request that certain geographic limits are applied to how the provider routes traffic for the IETF Network Slice service. For example, the customer may have a preference that its traffic does not pass through a particular country for political or security reasons.

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Maximal Occupancy Level: The maximal occupancy level specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume. Since an IETF Network Slice
service may include multiple connection constructs, this SLE should also say whether it applies for the entire IETF Network Slice service, for group of connections, or on a per connection basis.

Again, a customer may not be able to fully determine whether this SLE is being met by the provider.

Isolation: As described in Section 7, a customer may request that its traffic within its IETF Network Slice service is isolated from the effects of other network services supported by the same provider. That is, if another service exceeds capacity or has a burst of traffic, the customer’s IETF Network Slice service should remain unaffected and there should be no noticeable change to the quality of traffic delivered.

In general, a customer cannot tell whether a service provider is meeting this SLE. They cannot tell whether the variation of an SLI is because of changes in the underlying network or because of interference from other services carried by the network. If the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

Diversity: A customer may request that traffic on the connection between one set of SDPs should use different network resources from the traffic between another set of SDPs. This might be done to enhance the availability of the connectivity constructs within an IETF Network Slice service.

While availability is a measurable objective (see Section 4.1.1.1) this SLE requests a finer grade of control and is not directly measurable (although the customer might become suspicious if two connections fail at the same time).

4.2. IETF Network Slice Service Demarcation Points

As noted in Section 3.1, an IETF Network Slice is a logical network topology connecting a number of endpoints. Section 3.2 goes on to describe how the IETF Network Slice service is composed of a set of one or more connectivity constructs that describe connectivity between the Service Demarcation Points (SDPs) across the underlying network.

The characteristics of IETF Network Slice (SDPs are as follows.

* SDPs are conceptual points of connection to an IETF Network Slice.
  As such, they serve as the IETF Network Slice ingress/egress points.
* Each SDP maps to a device, application, or a network function, such as (but not limited to) routers, switches, interfaces/ports, firewalls, WAN, 4G/5G RAN nodes, 4G/5G Core nodes, application accelerators, server load balancers, NAT44 [RFC3022], NAT64 [RFC6146], HTTP header enrichment functions, and Performance Enhancing Proxies (PEPs) [RFC3135].

* An SDP is identified by a unique identifier in the context of an IETF Network Slice customer.

* Each SDP is associated with a set of provider-scope identifiers such as IP addresses, encapsulation-specific identifiers (e.g., VLAN tag, MPLS Label), interface/port numbers, node ID, etc.

* SDPs are mapped to endpoints of services/tunnels/paths within the IETF Network Slice during its initialization and realization.

  - A combination of the SDP identifier and SDP provider-network-scope identifiers define an SDP in the context of the Network Slice Controller (NSC) (see Section 5.3).

  - The NSC will use the SDP provider-network-scope identifiers as part of the process of realizing the IETF Network Slice.

For a given IETF Network Slice service, the IETF Network Slice customer and provider agree where the endpoint (i.e., the service demarcation point) is located. This determines what resources at the edge of the network form part of the IETF Network Slice and are subject to the set of SLOs and SLEs for a specific endpoint.

Figure 1 shows different potential scopes of an IETF Network Slice that are consistent with the different SDP locations. For the purpose of this discussion and without loss of generality, the figure shows customer edge (CE) and provider edge (PE) nodes connected by attachment circuits (ACs). Notes after the figure give some explanations.
Figure 1: Positioning IETF Service Demarcation Points

Explanatory notes for Figure 1 are as follows:

1. If the CE is operated by the IETF Network Slice service provider, then the edge of the IETF Network Slice may be within the CE. In this case the slicing process may utilize resources from within the CE such as buffers and queues on the outgoing interfaces.

2. The IETF Network Slice may be extended as far as the CE, to include the AC, but not to include any part of the CE. In this case, the CE may be operated by the customer or the provider. Slicing the resources on the AC may require the use of traffic tagging (such as through Ethernet VLAN tags) or may require traffic policing at the AC link ends.

3. In another model, the SDPs of the IETF Network Slice are the customer-facing ports on the PEs. This case can be managed in a way that is similar to a port-based VPN: each port (AC) or virtual port (e.g., VLAN tag) identifies the IETF Network Slice and maps to an IETF Network Slice SDP.
4. Finally, the SDP may be within the PE. In this mode, the PE classifies the traffic coming from the AC according to information (such as the source and destination IP addresses, payload protocol and port numbers, etc.) in order to place it onto an IETF Network Slice.

The choice of which of these options to apply is entirely up to the network operator. It may limit or enable the provisioning of particular managed services and the operator will want to consider how they want to manage CEs and what control they wish to offer the customer over AC resources.

Note that Figure 1 shows a symmetrical positioning of SDPs, but this decision can be taken on a per-SDP basis through agreement between the customer and provider.

In practice, it may be necessary to map traffic not only onto an IETF Network Slice, but also onto a specific connectivity construct if the IETF Network Slice supports more than one with a source at the specific SDP. The mechanism used will be one of the mechanisms described above, dependent on how the SDP is realized.

Finally, note (as described in Section 2.1) that an SDP is an abstract endpoint of an IETF Network Slice service and as such may be a device, interface, or software component and may, in the case of network functions virtualization (for example), be an abstract function supported within the provider’s network.

4.3. IETF Network Slice Decomposition

Operationally, an IETF Network Slice may be composed of two or more IETF Network Slices as specified below. Decomposed network slices are 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. Framework

A number of IETF Network Slice services will typically be provided over a shared underlying network infrastructure. Each IETF Network Slice consists of both the overlay connectivity and a specific set of dedicated network resources and/or functions allocated in a shared underlay network to satisfy the needs of the IETF Network Slice customer. In at least some examples of underlying network technologies, the integration between the overlay and various underlay resources is needed to ensure the guaranteed performance requested for different IETF Network Slices.

5.1. IETF Network Slice Stakeholders

An IETF Network Slice and its realization involves the following stakeholders. The IETF Network Slice customer and IETF Network Slice provider (see Section 2.1) are also stakeholders.

Orchestrator: An orchestrator is an entity that composes different services, resource, and network requirements. It interfaces with the IETF NSC when composing a complex service such as an end-to-end network slice.

IETF Network Slice Controller (NSC): The NSC realizes an IETF Network Slice in the underlying network, and maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed to support interworking between different NSC implementations and different orchestrator implementations.

Network Controller: The Network Controller is a form of network infrastructure controller that offers network resources to the NSC to realize a particular network slice. This may be an existing network controller associated with one or more specific technologies that may be adapted to the function of realizing IETF Network Slices in a network.

5.2. Expressing Connectivity Intents

An IETF Network Slice customer communicates with the NSC using the IETF Network Slice Service Interface.

An IETF Network Slice customer may be a network operator who, in turn, use the IETF Network Slice to provide a service for another IETF Network Slice customer.
Using the IETF Network Slice Service Interface, a customer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved. That is, the customer's view of a slice is an abstract one. Customers normally have limited (or no) visibility into the provider network's actual topology and resource availability information.

This should be true even if both the customer and provider are associated with a single administrative domain, in order to reduce the potential for adverse interactions between IETF Network Slice customers and other users of the underlay network infrastructure.

The benefits of this model can include the following.

* Security: The underlay network components are less exposed to attack because the underlay network (or network operator) does not need to expose network details (topology, capacity, etc.) to the IETF Network Slice customers.

* Layered Implementation: The underlay network comprises network elements that belong to a different layer network than customer applications. Network information (advertisements, protocols, etc.) that a customer cannot interpret or respond to is not exposed to the customer. (Note - a customer should not use network information not exposed via the IETF Network Slice Service Interface, even if that information is available.)

* Scalability: Customers do not need to know any information beyond that which is exposed via the IETF Network Slice Service Interface.

The general issues of abstraction in a TE network are described more fully in [RFC7926].

This framework document does not assume any particular technology layer at which IETF Network Slices operate. A number of layers (including virtual L2, Ethernet or, IP connectivity) could be employed.

Data models and interfaces are needed to set up IETF Network Slices, and specific interfaces may have capabilities that allow creation of slices within specific technology layers.

Layered virtual connections are comprehensively discussed in other IETF documents. See, for instance, GMPLS-based networks [RFC5212] and [RFC4397], or Abstraction and Control of TE Networks (ACTN) [RFC8453] and [RFC8454]. The principles and mechanisms associated with layered networking are applicable to IETF Network Slices.
There are several IETF-defined mechanisms for expressing the need for a desired logical network. The IETF Network Slice Service Interface carries data either in a protocol-defined format, or in a formalism associated with a modeling language.

For instance:

* The Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] and GMPLS User-Network Interface (UNI) using RSVP-TE [RFC4208] use a TLV-based binary encoding to transmit data.


* gRPC/GNMI [I-D.openconfig-rtgwg-gnmi-spec] uses a binary encoded programmable interface. ProtoBufs can be used to model gRPC and GNMI data.

* For data modeling, YANG ([RFC6020] and [RFC7950]) may be used to model configuration and other data for NETCONF, RESTCONF, and GNMI, among others.

While several generic formats and data models for specific purposes exist, it is expected that IETF Network Slice management may require enhancement or augmentation of existing data models. Further, it is possible that mechanisms will be needed to determine the feasibility of service requests before they are actually made.

5.3. IETF Network Slice Controller (NSC)

The IETF NSC takes abstract requests for IETF Network Slices and implements them using a suitable underlying technology. An IETF NSC is the key component for control and management of the IETF Network Slice. It provides the creation/modification/deletion, monitoring and optimization of IETF Network Slices in a multi-domain, a multi-technology and multi-vendor environment.

The main task of the IETF NSC is to map abstract IETF Network Slice requirements to concrete technologies and establish required connectivity ensuring that resources are allocated to the IETF Network Slice as necessary.

The IETF Network Slice Service Interface is used for communicating details of an IETF Network Slice (configuration, selected policies, operational state, etc.), as well as information about status and performance of the IETF Network Slice. The details for this IETF Network Slice Service Interface are not in scope for this document.
The controller provides the following functions.

* Provides an IETF Network Slice Service Interface for creation/modification/deletion of the IETF Network Slices that is agnostic to the technology of the underlying network. The API exposed by this interface communicates the Service Demarcation Points of the IETF Network Slice, IETF Network Slice SLO/SLE parameters (and possibly monitoring thresholds), applicable input selection (filtering) and various policies, and provides a way to monitor the slice.

* Determines an abstract topology connecting the SDPs of the IETF Network Slice that meets criteria specified via the IETF Network Slice Service Interface. The NSC also retains information about the mapping of this abstract topology to underlying components of the IETF Network Slice as necessary to monitor IETF Network Slice status and performance.

* Provides "Mapping Functions" for the realization of IETF Network Slices. In other words, it will use the mapping functions that:
  - map IETF Network Slice Service Interface requests that are agnostic to the technology of the underlying network to technology-specific network configuration interfaces.
  - map filtering/selection information as necessary to entities in the underlay network.

* The controller collects telemetry data (e.g., OAM results, statistics, states, etc.) via a network configuration interface for all elements in the abstract topology used to realize the IETF Network Slice.

* Evaluates the current performance against IETF Network Slice SLO parameters using the telemetry data from the underlying realization of an IETF Network Slice (i.e., services/paths/tunnels). Exposes this performance to the IETF Network Slice customer via the IETF Network Slice Service Interface. The IETF Network Slice Service Interface may also include the capability to provide notifications if the IETF Network Slice performance reaches threshold values defined by the IETF Network Slice customer.
5.3.1. 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 2).

IETF Network Slice Service Interface: The IETF Network Slice Service Interface is an interface between a customer’s higher level operation system (e.g., a network slice orchestrator or a customer network management system) and the NSC. It is agnostic to the technology of the underlying network. The customer can use this interface to communicate the requested characteristics and other requirements for the IETF Network Slice, and the NSC can use the interface to report the operational state of an IETF Network Slice to the customer.

Network Configuration Interface: The Network Configuration Interface is an interface between the NSC and network controllers. It is technology-specific and may be built around the many network models already defined within the IETF.

These interfaces can be considered in the context of the Service Model and Network Model described in [RFC8309] and, together with the Device Configuration Interface used by the Network Controllers, provides a consistent view of service delivery and realization.

Figure 2: Interfaces of the IETF Network Slice Controller
5.3.1.1. IETF Network Slice Service Interface

The IETF Network Slice Controller provides an IETF Network Slice Service Interface that allows customers to request and monitor IETF Network Slices. Customers operate on abstract IETF Network Slices, with details related to their realization hidden.

The IETF Network Slice Service Interface is also independent of the type of network functions or services that need to be connected, i.e., it is independent of any specific storage, software, protocol, or platform used to realize physical or virtual network connectivity or functions in support of IETF Network Slices.

The IETF Network Slice Service Interface uses protocol mechanisms and information passed over those mechanisms to convey desired attributes for IETF Network Slices and their status. The information is expected to be represented as a well-defined data model, and should include at least SDP and connectivity information, SLO/SLE specification, and status information.

5.3.2. Management Architecture

The management architecture described in Figure 2 may be further decomposed as shown in Figure 3. This should also be seen in the context of the component architecture shown in Figure 5 and corresponds to the architecture in [RFC8309].
5.4. IETF Network Slice Structure

An IETF Network Slice is a set of connection constructs between various SDPs to form a logical network that meets the SLOs agreed upon.

Figure 3: Interface of IETF Network Slice Management Architecture
Figure 4: IETF Network Slice

Figure 4 illustrates this by showing a case where an IETF Network Slice provides connectivity between a set of SDP pairs (SDP1 to SDP2, ..., SDPm to SDPn) in a set of P2P connectivity constructs each 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 service/tunnel/path Endpoints (EP1 to EPn) in the underlay network. Also, the SDPs in the same IETF Network Slice may belong to the same or different address spaces.

6. Realizing IETF Network Slices

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 Network Configuration Interface. However, this section provides an overview of the components and processes involved in realizing an IETF Network Slice.

The realization can be achieved in a form of either physical or logical connectivity using VPNs, virtual networks (VNs), or a variety of tunneling technologies such as Segment Routing, MPLS, etc. Accordingly, SDPs may be realized as physical or logical service or network functions.

Legend
SDP: IETF Network Slice Service Demarcation Point
EP: Service/tunnel/path Endpoint used to realize the IETF Network Slice
6.1. Architecture to Realize IETF Network Slices

The architecture described in this section is deliberately at a high level. It is not intended to be prescriptive: implementations and technical solutions may vary freely. However, this approach provides a common framework that other documents may reference in order to facilitate a shared understanding of the work.

Figure 5 shows the architectural components of a network managed to provide IETF Network Slices. The customer’s view is of individual IETF Network Slices with their SDPs, and connectivity constructs. Requests for IETF Network Slices are delivered to the NSC.

The figure shows, without loss of generality, the CEs, ACs, and PEs, that exist in the network. The SDPs are not shown and can be placed in any of the ways described in Section 4.2.
The diagram illustrates the architecture of an IETF Network Slice, showing the relationship between CE, PE, and NSC devices and various network service request interactions. The vertical arrows represent请求 and provider views, with grouping and mapping functions. The NSC is involved in resource partitioning and topology filtering, with Program the Network at the bottom. The diagram includes textual annotations for clearer understanding.

Figure 5: Architecture of an IETF Network Slice
The network itself (at the bottom of the figure) comprises an underlay network. This could be a physical network, but may be a virtual network. The underlay network is provisioned through network controllers that may utilize device controllers [RFC8309].

The underlay network may optionally be filtered or customized by the network operator to produce a number of network topologies that we call Filter Topologies. Customization is just a way of selecting specific resources (e.g., nodes and links) from the underlay network according to their capabilities and connectivity in the underlay network. These actions are configuration options or operator policies. The resulting topologies can be used as candidates to host IETF Network Slices and provide a useful way for the network operator to know in advance that all of the resources they are using to plan an IETF Network Slice would be able to meet specific SLOs and SLEs. The creation of a Filter Topology could be an offline planning activity or could be performed dynamically as new demands arise. The use of Filter Topologies is entirely optional in the architecture, and IETF Network Slices could be hosted directly on the underlay network.

Recall that an IETF Network Slice is a service requested by / provided for the customer. The IETF Network Slice service is expressed in terms of one or more connectivity constructs. An implementation or operator is free to limit the number of connectivity constructs in a slice to exactly one. Each connectivity construct is associated within the IETF Network Slice service request with a set of SLOs and SLEs. The set of SLOs and SLEs does not need to be the same for every connectivity construct in the slice, but an implementation or operator is free to require that all connectivity constructs in a slice have the same set of SLOs and SLEs.

One or more connectivity constructs from one or more slices are mapped to a set of network resources called a Network Resource Partition (NRP). A single connectivity construct is mapped to only one NRP (that is, the relationship is many to one). An NRP may be chosen to support a specific connectivity construct because of its ability to support a specific set of SLOs and SLEs, or its ability to support particular connectivity types, or for any administrative or operational reason. An implementation or operator is free to map each connectivity construct to a separate NRP, although there may be scaling implications depending on the solution implemented. Thus, the connectivity constructs from one slice may be mapped to one or more NRPs. By implication from the above, an implementation or operator is free to map all the connectivity constructs in a slice to a single NRP, and to not share that NRP with connectivity constructs from another slice.
An NRP is simply a collection of resources identified in the underlay network. Thus, the NRP is a scoped view of a topology and may be considered as a topology in its own right. The process of determining the NRP may be made easier if the underlay network topology is first filtered into a Filter Topology in order to be aware of the subset of network resources that are suitable for specific NRPs, but this is optional.

The steps described here can be applied in a variety of orders according to implementation and deployment preferences. Furthermore, the steps may be iterative so that the components are continually refined and modified as network conditions change and as service requests are received or relinquished, and even the underlay network could be extended if necessary to meet the customers’ demands.

6.2. Procedures to Realize IETF Network Slices

There are a number of different technologies that can be used in the underlay, including physical connections, MPLS, time-sensitive networking (TSN), Flex-E, etc.

An IETF Network Slice can be realized in a network, using specific underlying technology or technologies. The creation of a new IETF Network Slice will be realized with following steps:

* The NSC exposes the network slicing capabilities that it offers for the network it manages.

* The customer may issue a request to determine whether a specific IETF Network Slice could be supported by the network. The NSC may respond indicating a simple yes or no, and may supplement a negative response with information about what it could support were the customer to change some requirements.

* The customer requests an IETF Network Slice. The NSC may respond that the slice has or has not been created, and may supplement a negative response with information about what it could support were the customer to change some requirements.

* When processing a customer request for an IETF Network Slice, the NSC maps the request to the network capabilities and applies provider policies before creating or supplementing the NRP.
Regardless of how IETF Network Slice is realized in the network (i.e., using tunnels of different types), the definition of the IETF Network Slice service does not change at all. The only difference is how the slice is realized. The following sections briefly introduce how some existing architectural approaches can be applied to realize IETF Network Slices.

6.3. Applicability of ACTN to IETF Network Slices

Abstraction and Control of TE Networks (ACTN - [RFC8453]) is a management architecture and toolkit used to create virtual networks (VNs) on top of a TE underlay network. The VNs can be presented to customers for them to operate as private networks.

In many ways, the function of ACTN is similar to IETF network slicing. Customer requests for connectivity-based overlay services are mapped to dedicated or shared resources in the underlay network in a way that meets customer guarantees for service level objectives and for separation from other customers' traffic. [RFC8453] describes the function of ACTN as collecting resources to establish a logically dedicated virtual network over one or more TE networks. Thus, in the case of a TE-enabled underlay network, the ACTN VN can be used as a basis to realize IETF network slicing.

While the ACTN framework is a generic VN framework that can be used for VN services beyond the IETF Network Slice, it also a suitable basis for delivering and realizing IETF Network Slices.

Further discussion of the applicability of ACTN to IETF Network Slices including a discussion of the relevant YANG models can be found in [I-D.ietf-teas-applicability-actn-slicing].

6.4. Applicability of Enhanced VPNs to IETF Network Slices

An enhanced VPN (VPN+) is designed to support the needs of new applications, particularly applications that are associated with 5G services, by utilizing an approach that is based on existing VPN and TE technologies and adds characteristics that specific services require over and above VPNs as they have previously been specified.

An enhanced VPN can be used to provide enhanced connectivity services between customer sites and can be used to create the infrastructure to underpin a network slicing service.

It is envisaged that enhanced VPNs will be delivered using a combination of existing, modified, and new networking technologies.
[I-D.ietf-teas-enhanced-vpn] describes the framework for Enhanced Virtual Private Network (VPN+) services.

6.5.  Network Slicing and Aggregation in IP/MPLS Networks

Network slicing provides the ability to partition a physical network into multiple isolated logical networks of varying sizes, structures, and functions so that each slice can be dedicated to specific services or customers.

Many approaches are currently being worked on to support IETF Network Slices in IP and MPLS networks with or without the use of Segment Routing. Most of these approaches utilize a way of marking packets so that network nodes can apply specific routing and forwarding behaviors to packets that belong to different IETF Network Slices. Different mechanisms for marking packets have been proposed (including using MPLS labels and Segment Routing segment IDs) and those mechanisms are agnostic to the path control technology used within the underlay network.

These approaches are also sensitive to the scaling concerns of supporting a large number of IETF Network Slices within a single IP or MPLS network, and so offer ways to aggregate the connectivity constructs of slices (or whole slices) so that the packet markings indicate an aggregate or grouping where all of the packets are subject to the same routing and forwarding behavior.

At this stage, it is inappropriate to mention any of these proposed solutions that are currently work in progress and not yet adopted as IETF work.

7.  Isolation in IETF Network Slices

7.1.  Isolation as a Service Requirement

An IETF Network Slice customer may request that the IETF Network Slice delivered to them is such that changes to other IETF Network Slices or to other services do not have any negative impact on the delivery of the IETF Network Slice. The IETF Network Slice customer may specify the degree to which their IETF Network Slice is unaffected by changes in the provider network or by the behavior of other IETF Network Slice customers. The customer may express this via an SLE it agrees with the provider. This concept is termed ‘isolation’.
In general, a customer cannot tell whether a service provider is meeting an isolation SLE. If the service varies such that an SLO is breached then the customer will become aware of the problem, and if the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

7.2. Isolation in IETF Network Slice Realization

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 of 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 IETF Network Slices.

8. Management Considerations

IETF Network Slice realization needs to be instrumented in order to track how it is working, and it might be necessary to modify the IETF Network Slice as requirements change. Dynamic reconfiguration might be needed.

The various management interfaces and components are discussed in Section 5.

9. 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 customer-defined IETF Network Slices will be mapped to their realization in the underlay networks. Underlay networks will require capabilities to conform to customer’s requests as some aspects of security may be expressed in SLEs.

IETF NSC authentication: Underlying networks need to be protected
against the attacks from an adversary NSC as this could destabilize overall network operations. An IETF Network Slice may span across different networks, therefore, the NSC should have strong authentication with each of these networks. Furthermore, both the IETF Network Slice Service Interface and the Network Configuration Interface 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 customer 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 customer is responsible for end-to-end encryption of its own traffic.

Note: See [NGMN_SEC] on 5G network slice security for discussion relevant to this section.

IETF Network Slices might use underlying virtualized networking. All types of virtual networking require special consideration to be given to the separation of traffic between distinct virtual networks, as well as some degree of protection from effects of traffic use of underlying network (and other) resources from other virtual networks sharing those resources.

For example, if a service requires a specific upper bound of latency, then that service can be degraded by added delay in transmission of service packets caused by the activities of another service or application using the same resources.

Similarly, in a network with virtual functions, noticeably impeding access to a function used by another IETF Network Slice (for instance, compute resources) can be just as service-degrading as delaying physical transmission of associated packet in the network.

While an IETF Network Slice might include encryption and other security features as part of the service, customers might be well advised to take responsibility for their own security needs, possibly by encrypting traffic before hand-off to a service provider.
10. Privacy Considerations

Privacy of IETF Network Slice service customers must be preserved. It should not be possible for one IETF Network Slice customer to discover the presence of other customers, nor should sites that are members of one IETF Network Slice be visible outside the context of that IETF Network Slice.

In this sense, it is of paramount importance that the system use the privacy protection mechanism defined for the specific underlying technologies that support the slice, including in particular those mechanisms designed to preclude acquiring identifying information associated with any IETF Network Slice customer.

11. IANA Considerations

This document makes no requests for IANA action.

12. Informative References


[I-D.ietf-opsawg-sap]

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

[I-D.openconfig-rtgw-gnmi-spec]


Farrel, et al. Expires 7 September 2022 [Page 38]


Acknowledgments

The entire TEAS Network Slicing design team and everyone participating in related discussions has contributed to this document. Some text fragments in the document have been copied from the [I-D.ietf-teas-enhanced-vpn], for which we are grateful.

Significant contributions to this document were gratefully received from the contributing authors listed in the "Contributors" section. In addition we would like to also thank those others who have attended one or more of the design team meetings, including the following people not listed elsewhere:

* Aihua Guo
* Bo Wu
* Greg Mirsky
* Lou Berger
* Rakesh Gandhi
* Ran Chen
* Sergio Belotti
* Stewart Bryant
* Tomonobu Niwa
* Xuesong Geng

Further useful comments were received from Daniele Ceccarelli, Uma Chunduri, Pavan Beeram, Tarek Saad, Kenichi Ogaki, Oscar Gonzalez de Dios, Xiaobing Niu, Dan Voyer, Igor Bryskin, Luay Jalil, Joel Halpern, John Scudder, and John Mollooly.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).
Contributors

The following authors contributed significantly to this document:

Eric Gray (The original editor of the foundation documents)
Independent
Email: ewgray@graiymage.com

Jari Arkko
Ericsson
Email: jari.arkko@piuha.net

Mohamed Boucadair
Orange
Email: mohamed.boucadair@orange.com

Dhruv Dhody
Huawei, India
Email: dhruv.ietf@gmail.com

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

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com

Authors’ Addresses

Adrian Farrel (editor)
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

John Drake (editor)
Juniper Networks
United States of America
Email: jdrake@juniper.net

Reza Rokui
Ciena
Email: rrokui@ciena.com
Abstract

This document describes network slicing in the context of networks built from IETF technologies. It defines the term "IETF Network Slice" and establishes the general principles of network slicing in the IETF context.

The document discusses the general framework for requesting and operating IETF Network Slices, the characteristics of an IETF Network Slice, the necessary system components and interfaces, and how abstract requests can be mapped to more specific technologies. The document also discusses related considerations with monitoring and security.

This document also provides definitions of related terms to enable consistent usage in other IETF documents that describe or use aspects of IETF Network Slices.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 1 January 2023.

Copyright Notice

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

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

Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Terms and Abbreviations</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Core Terminology</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>IETF Network Slice</td>
<td>7</td>
</tr>
<tr>
<td>3.1</td>
<td>Definition and Scope of IETF Network Slice</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>IETF Network Slice Service</td>
<td>8</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Ancillary CEs</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>IETF Network Slice System Characteristics</td>
<td>13</td>
</tr>
<tr>
<td>4.1</td>
<td>Objectives for IETF Network Slices</td>
<td>13</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Service Level Objectives</td>
<td>14</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Service Level Expectations</td>
<td>15</td>
</tr>
<tr>
<td>4.2</td>
<td>IETF Network Slice Service Demarcation Points</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>IETF Network Slice Composition</td>
<td>20</td>
</tr>
<tr>
<td>5.</td>
<td>Framework</td>
<td>21</td>
</tr>
<tr>
<td>5.1</td>
<td>IETF Network Slice Stakeholders</td>
<td>21</td>
</tr>
<tr>
<td>5.2</td>
<td>Expressing Connectivity Intents</td>
<td>21</td>
</tr>
<tr>
<td>5.3</td>
<td>IETF Network Slice Controller (NSC)</td>
<td>23</td>
</tr>
<tr>
<td>5.3.1</td>
<td>IETF Network Slice Controller Interfaces</td>
<td>25</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Management Architecture</td>
<td>26</td>
</tr>
<tr>
<td>6.</td>
<td>Realizing IETF Network Slices</td>
<td>27</td>
</tr>
<tr>
<td>6.1</td>
<td>Architecture to Realize IETF Network Slices</td>
<td>28</td>
</tr>
<tr>
<td>6.2</td>
<td>Procedures to Realize IETF Network Slices</td>
<td>31</td>
</tr>
<tr>
<td>6.3</td>
<td>Applicability of ACTN to IETF Network Slices</td>
<td>32</td>
</tr>
<tr>
<td>6.4</td>
<td>Applicability of Enhanced VPNs to IETF Network Slices</td>
<td>33</td>
</tr>
</tbody>
</table>
A number of use cases would benefit from a network service that supplements connectivity, such as that offered by a VPN service, with an assurance of meeting a set of specific network performance objectives. This connectivity and resource commitment is referred to as a network slice and is expressed in terms of connectivity constructs (see Section 3) and service objectives (see Section 4). 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. 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 (known as Service Demarcation Points (SDPs) - see Section 2.1 and Section 4.2) over a shared underlay network. The term IETF Network Slice service is also introduced to describe the service requested by and provided to the service provider's customer.

Services that might benefit from IETF Network Slices include, but are not limited to:

* 5G services (e.g. eMBB, URLLC, mMTC) (See [TS23501])

* Network wholesale services

* Network infrastructure sharing among operators

* NFV connectivity and Data Center Interconnect
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 with different requirements to coexist over a shared underlay network. A request for an IETF Network Slice service is agnostic to the technology in the underlay network so as to allow a customer to describe their network connectivity objectives in a common format, independent of the underlay technologies used.

This document also provides a framework for discussing IETF Network Slices. The framework is intended as a structure for discussing interfaces and technologies.

For example, virtual private networks (VPNs) have served the industry well as a means of providing different groups of users with logically isolated access to a common network. The common or base network that is used to support the VPNs is often referred to as an underlay network, and the VPN is often called an overlay network. An overlay network may, in turn, serve as an underlay network to support another overlay network.

Note that it is conceivable that extensions to IETF technologies are needed in order to fully support all the capabilities that can be implemented with network slices. Evaluation of existing technologies, proposed extensions to existing protocols and interfaces, and the creation of new protocols or interfaces are outside the scope of this document.

1.1. Background

The concept of network slicing has gained traction driven largely by needs surfacing from 5G ([NGMN-NS-Concept], [TS23501], and [TS28530]). In [TS23501], a Network Slice is defined as "a logical network that provides specific network capabilities and network characteristics", and a Network Slice Instance is defined as "A set of Network Function instances and the required resources (e.g. compute, storage and networking resources) which form a deployed Network Slice." According to [TS28530], an end-to-end network slice consists of three major types of network segments: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). An IETF Network Slice provides the required connectivity between different entities in RAN and CN segments of an end-to-end network slice, with a specific performance commitment (for example, serving as a TN slice). For each end-to-end network slice, the topology and performance requirement on a customer’s use of an IETF Network Slice can be very different, which requires the underlay network to have the capability of supporting multiple different IETF Network Slices.
While network slices are commonly discussed in the context of 5G, it is important to note that IETF Network Slices are a narrower concept with a broader usage profile, and focus primarily on particular network connectivity aspects. Other systems, including 5G deployments, may use IETF Network Slices as a component to create entire systems and concatenated constructs that match their needs, including end-to-end connectivity.

An IETF Network Slice could span multiple technologies and multiple administrative domains. Depending on the IETF Network Slice customer’s requirements, an IETF Network Slice could be isolated from other, often concurrent IETF Network Slices in terms of data, control and management planes.

The customer expresses requirements for a particular IETF Network Slice service by specifying what is required rather than how the requirement is to be fulfilled. That is, the IETF Network Slice customer’s view of an IETF Network Slice is an abstract one.

Thus, there is a need to create logical network structures with required characteristics. The customer of such a logical network can require a degree of isolation and performance that previously might not have been satisfied by overlay VPNs. Additionally, the IETF Network Slice customer might ask for some level of control of their virtual networks, e.g., to customize the service paths in a network slice.

This document specifies definitions and a framework for the provision of an IETF Network Slice service. Section 6 briefly indicates some candidate technologies for realizing IETF Network Slices.

2. Terms and Abbreviations

The following abbreviations are used in this document.

* NSC: Network Slice Controller
* SDP: Service Demarcation Point
* SLA: Service Level Agreement
* SLE: Service Level Expectation
* SLI: Service Level Indicator
* SLO: Service Level Objective
The meaning of these abbreviations is defined in greater details in the remainder of this document.

2.1. Core Terminology

The following terms are presented here to give context. Other terminology is defined in the remainder of this document.

Customer: A customer is the requester of an IETF Network Slice service. Customers may request monitoring of SLOs. A customer may be an entity such as an enterprise network or a network operator, an individual working at such an entity, a private individual contracting for a service, or an application or software component. A customer may be an external party (classically a paying customer) or a division of a network operator that uses the service provided by another division of the same operator. Other terms that have been applied to the customer role are "client" and "consumer".

Provider: A provider is the organization that delivers an IETF Network Slice service. A provider is the network operator that controls the network resources used to construct the network slice (that is, the network that is sliced). The provider’s network maybe a physical network or may be a virtual network supplied by another service provider.

Customer Edge (CE): The customer device that provides connectivity to a service provider. Examples include routers, Ethernet switches, firewalls, 4G/5G RAN or Core nodes, application accelerators, server load balancers, HTTP header enrichment functions, and PEPs (Performance Enhancing Proxy). In some circumstances CEs are provided to the customer and managed by the provider.

Provider Edge (PE): The device within the provider network to which a CE is attached. A CE may be attached to multiple PEs, and multiple CEs may be attached to a given PE.

Attachment Circuit (AC): A channel connecting a CE and a PE over which packets that belong to an IETF Network Slice service are exchanged. An AC is, by definition, technology specific; that is, the AC defines how customer traffic is presented to the provider network. The customer and provider agree (through configuration) on which values in which combination of layer 2 and layer 3 header and payload fields within a packet identify to which (IETF Network Slice service, connectivity construct, and SLOs/SLEs) that packet is assigned. The customer and provider may agree on a per (IETF Network Slice service, connectivity construct, and SLOs/SLEs)
basis to police or shape traffic on the AC in both the ingress (CE to PE) direction and egress (PE to CE) direction. This ensures that the traffic is within the capacity profile that is agreed in an IETF Network Slice service. Excess traffic is dropped by default, unless specific out-of-profile policies are agreed between the customer and the provider. As described in Section 4.2 the AC may be part of the IETF Network Slice service or may be external to it. Because SLOs and SLEs characterise the performance of the underlay network between a sending SDP and a set of receiving SDPs, the traffic policers and traffic shapers apply to a specific connectivity construct on an AC.

Service Demarcation Point (SDP): The point at which an IETF Network Slice service is delivered by a service provider to a customer. Depending on the service delivery model (see Section 4.2) this may be a CE or a PE, and could be a device, a software component, or an abstract virtual function supported within the provider’s network. Each SDP must have a unique identifier (e.g., an IP address or MAC address) within a given IETF Network Slice service and may use the same identifier in multiple IETF Network Slice services.

An SDP may be abstracted as a Service Attachment Point (SAP) [I-D.ietf-opsawg-sap] for the purpose of generalizing the concept across multiple service types and representing it in management and configuration systems.

Connectivity Construct: A set of SDPs together with a communication type that defines how traffic flows between the SDPs. An IETF Network Slice service is specified in terms of a set of SDPs, the associated connectivity constructs and the service objectives that the customer wishes to see fulfilled.

3. IETF Network Slice

IETF Network Slices are created to meet specific requirements, typically expressed as bandwidth, latency, latency variation, and other desired or required characteristics. Creation of an IETF Network Slice is initiated by a management system or other application used to specify network-related conditions for particular traffic flows in response to an actual or logical IETF Network Slice service request.

Once created, these slices can be monitored, modified, deleted, and otherwise managed.
Applications and components will be able to use these IETF Network Slices to move packets between the specified end-points of the service in accordance with specified characteristics.

A clear distinction should be made between the "IETF Network Slice service" which is the function delivered to the customer (see Section 3.2) and which is agnostic to the technologies and mechanisms used by the service provider, and the "IETF Network Slice" which is the realization of the service in the provider’s network achieved by partitioning network resources and by applying certain tools and techniques within the network (see Section 3.1 and Section 6).

3.1. Definition and Scope of IETF Network Slice

The term "Slice" refers to a set of characteristics and behaviors that differentiate one type of user-traffic from another within a network. An IETF Network Slice is a logical partition of a network that uses IETF technology. An IETF Network Slice assumes that an underlay network is capable of changing the configurations of the network devices on demand, through in-band signaling, or via controllers.

An IETF Network Slice enables connectivity between a set of Service Demarcation Points (SDPs) with specific Service Level Objectives (SLOs) and Service Level Expectations (SLEs) (see Section 4) over a common underlay network. The SLOs and SLEs characterize the performance of the underlay network between a sending SDP and a set of receiving SDPs. Thus, an IETF Network Slice delivers a service to a customer by meeting connectivity resource requirements and associated network capabilities such as bandwidth, latency, jitter, and network functions with other resource behaviors such as compute and storage availability.

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 [TS23501].

3.2. IETF Network Slice Service

A service provider delivers an IETF Network Slice service for a customer by realizing an IETF Network Slice. The IETF Network Slice service is agnostic to the technology of the underlay network, and its realization may be selected based upon multiple considerations including its service requirements and the capabilities of the underlay network. This allows an IETF Network Slice service customer
to describe their network connectivity and relevant objectives in a common format, independent of the underlay technologies used.

The IETF Network Slice service is specified in terms of a set of SDPs, a set of one or more connectivity constructs between subsets of these SDPs, and a set of SLOs and SLEs (see Section 4) for each SDP sending to each connectivity construct. A communication type (point-to-point (P2P), point-to-multipoint (P2MP), or any-to-any (A2A)) is specified for each connectivity construct. That is, in a given IETF Network Slice service there may be one or more connectivity constructs of the same or different type, each connectivity construct may be between a different subset of SDPs, for a given connectivity construct each sending SDP has its own set of SLOs and SLEs, and the SLOs and SLEs in each set may be different. Note that different connectivity constructs can be specified in the service request but the service provider may decide how many connectivity constructs per IETF Network Slice service it wishes to support such that an IETF Network Slice service may be limited to one connectivity construct or may support many.

This approach results in the following possible connectivity constructs:

* For a P2P connectivity construct, there is one sending SDP and one receiving SDP. This construct is like a private wire or a tunnel. All traffic injected at the sending SDP is intended to be received by the receiving SDP. The SLOs and SLEs apply at the sender (and implicitly at the receiver).

* For a P2MP connectivity construct, there is only one sending SDP and more than one receiving SDP. This is like a P2MP tunnel or multi-access VLAN segment. All traffic from the sending SDP is intended to be received by all the receiving SDPs. There is one set of SLOs and SLEs that applies at the sending SDP (and implicitly at all receiving SDPs).
* With an A2A connectivity construct, any sending SDP may send to any one receiving SDP or any set of receiving SDPs in the construct. There is an implicit level of routing in this connectivity construct that is not present in the other connectivity constructs because the provider’s network must determine to which receiving SDPs to deliver each packet. This construct may be used to support P2P traffic between any pair of SDPs, or to support multicast or broadcast traffic from one SDP to a set of other SDPs. In the latter case, whether the service is delivered using multicast within the provider’s network or using "ingress replication" or some other means is out of scope of the specification of the service. A service provider may choose to support A2A constructs, but to limit the traffic to unicast.

The SLOs/SLEs in an A2A connectivity construct apply to individual sending SDPs regardless of the receiving SDPs, and there is no linkage between sender and receiver in the specification of the connectivity construct. A sending SDP may be "disappointed" if the receiver is over-subscribed. If a customer wants to be more specific about different behaviors from one SDP to another SDP, they should use P2P connectivity constructs.

A customer traffic flow may be unicast or multicast, and various network realizations are possible:

* Unicast traffic may be mapped to a P2P connectivity construct for direct delivery, or to an A2A connectivity construct for the service provider to perform routing to the destination SDP. It would be unusual to use a P2MP connectivity construct to deliver unicast traffic because all receiving SDPs would get a copy, but this can still be done if the receivers are capable of dropping the unwanted traffic.

* A bidirectional unicast service can be constructed by specifying two P2P connectivity constructs. An additional SLE may specify fate-sharing in this case.

* Multicast traffic may be mapped to a set of P2P connectivity constructs, a single P2MP connectivity construct, or a mixture of P2P and P2MP connectivity constructs. Multicast may also be supported by an A2A connectivity construct. The choice clearly influences how and where traffic is replicated in the network. With a P2MP or A2A connectivity construct, it is the operator’s choice whether to realize the construct with ingress replication, multicast in the core, P2MP tunnels, or hub-and-spoke. This choice should not change how the customer perceives the service.
* The concept of a multipoint-to-point (MP2P) service can be realized with multiple P2P connectivity constructs. Note that, in this case, the egress may simultaneously receive traffic from all ingresses. The SLOs at the sending SDPs must be set with this in mind because the provider’s network is not capable of coordinating the policing of traffic across multiple distinct source SDPs. It is assumed that the customer, requesting SLOs for the various P2P connectivity constructs, is aware of the capabilities of the receiving SDP. If the receiver receives more traffic than it can handle, it may drop some and introduce queuing delays.

* The concept of a multipoint-to-multipoint (MP2MP) service can best be realized using a set of P2MP connectivity constructs, but could be delivered over an A2A connectivity construct if each sender is using multicast. As with MP2P, the customer is assumed to be familiar with the capabilities of all receivers. A customer may wish to achieve an MP2MP service using a hub-and-spoke architecture where they control the hub: that is, the hub may be an SDP or an ancillary CE (see Section 3.2.1) and the service may be achieved by using a set of P2P connectivity constructs to the hub, and a single P2MP connectivity construct from the hub.

From the above, it can be seen that the SLOs of the senders define the SLOs for the receivers on any connectivity construct. That is, and in particular, the network may be expected to handle the traffic volume from a sender to all destinations. This extends to all connectivity constructs in an IETF Network Slice service.

Note that the realization of an IETF Network Slice service does not need to map the connectivity constructs one-to-one onto underlying network constructs (such as tunnels, etc.). The service provided to the customer is distinct from how the provider decides to deliver that service.

If a CE has multiple attachment circuits to a PE within a given IETF Network Slice service and they are operating in single-active mode, then all traffic between the CE and its attached PEs transits a single attachment circuit; if they are operating in all-active mode, then traffic between the CE and its attached PEs is distributed across all of the active attachment circuits.

A given sending SDP may be part of multiple connectivity constructs within a single IETF Network Slice service, and the SDP may have different SLOs and SLEs for each connectivity construct to which it is sending. Note that a given sending SDP’s SLOs and SLEs for a given connectivity construct apply between it and each of the receiving SDPs for that connectivity construct.
An IETF Network Slice service provider may freely make a deployment choice as to whether to offer a 1:1 relationship between IETF Network Slice service and connectivity construct, or to support multiple connectivity constructs in a single IETF Network Slice service. In the former case, the provider might need to deliver multiple IETF Network Slice services to achieve the function of the second case.

It should be noted that per Section 9 of [RFC4364] an IETF Network Slice service customer may actually provide IETF Network Slice services to other customers in a mode sometimes referred to as "carrier’s carrier". In this case, the underlying IETF Network Slice service provider may be owned and operated by the same or a different provider network. As noted in Section 4.3, network slices may be composed hierarchically or serially.

Section 4.2 provides a description of endpoints in the context of IETF network slicing. These are known as Service Demarcation Points (SDPs). For a given IETF Network Slice service, the customer and provider agree, on a per-SDP basis which end of the attachment circuit provides the SDP (i.e., whether the attachment circuit is inside or outside the IETF Network Slice service). This determines whether the attachment circuit is subject to the set of SLOs and SLEs at the specific SDP.

3.2.1. Ancillary CEs

It may be the case that the set of SDPs that delimits an IETF Network Slice Service needs to be supplemented with additional senders or receivers. An additional sender could be, for example, an IPTV or DNS server either within the provider’s network or attached to it, while an extra receiver could be, for example, a node reachable via the Internet. This is modelled as a set of ancillary CEs which supplement the other SDPs in one or more connectivity constructs, or which have their own connectivity constructs. Note that an ancillary CE can either have a resolvable address, e.g., an IP address or MAC address, or it may be a placeholder, e.g., IPTV or DNS server, which is resolved within the provider’s network when the IETF Network Slice service is instantiated.

Thus, an ancillary CE may be a node within the provider network (i.e., not a CE). An example is a node that provides a service function. Another example is a node that acts as a hub. There will be times when the customer wishes to explicitly select one of these. Alternatively, an ancillary CE may be a service function at an unknown point in the provider’s network. In this case, the function may be a placeholder that has its addressed resolved as part of the realization of the slice service.
4. IETF Network Slice System Characteristics

The following subsections describe the characteristics of IETF Network Slices in addition to the list of SDPs, the connectivity constructs, and the technology of the ACs.

4.1. Objectives for IETF Network Slices

An IETF Network Slice service is defined in terms of quantifiable characteristics known as Service Level Objectives (SLOs) and unquantifiable characteristics known as Service Level Expectations (SLEs). SLOs are expressed in terms Service Level Indicators (SLIs), and together with the SLEs form the contractual agreement between service customer and service provider known as a Service Level Agreement (SLA).

The terms are defined as follows:

* 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 customer can determine whether the provider is meeting the SLOs by performing measurements on the traffic.

* A Service Level Expectation (SLE) is an expression of an unmeasurable service-related request that a customer of an IETF Network Slice makes of the provider. An SLE is distinct from an SLO because the customer may have little or no way of determining whether the SLE is being met, but they still contract with the provider for a service that meets the expectation.

* A Service Level Agreement (SLA) is an explicit or implicit contract between the customer of an IETF Network Slice service and the provider of the slice. The SLA is expressed in terms of a set of SLOs and SLEs that are to be applied for a given connectivity construct between a sending SDP and the set of receiving SDPs, and may describe the extent to which divergence from individual SLOs and SLEs can be tolerated, and commercial terms as well as any consequences for violating these SLOs and SLEs.
4.1.1. Service Level Objectives

SLOs define a set of measurable network attributes and characteristics that describe an IETF Network Slice service. SLOs do not describe how an IETF Network Slice service is 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 service may include multiple connectivity constructs that associate sets of endpoints (SDPs). SLOs apply to a given connectivity construct and apply to a specific direction of traffic flow. That is, they apply to a specific sending SDP and the set of receiving SDPs.

4.1.1.1. Some Common SLOs

SLOs can be described as ‘Directly Measurable Objectives’: they are always measurable. See Section 4.1.2 for the description of Service Level Expectations which are unmeasurable service-related requests sometimes known as ‘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’. Future specifications (such as IETF Network Slice service YANG models) may precisely define these SLOs, and other SLOs may be introduced as described in Section 4.1.1.2.

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). [RFC7679] discusses one-way metrics.

Maximum Permissible Delay Variation: Packet delay variation (PDV) as defined by [RFC3393], is 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 connectivity construct is available in accordance with all of the SLOs associated with it. Availability will often be expressed along with the time period over which the availability is measured, and specifying the maximum allowed single period of downtime.

4.1.1.2. Other Service Level Objectives

Additional SLOs may be defined to provide additional description of the IETF Network Slice service that a customer requests. These would be specified in further documents.

If the IETF Network Slice 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).

4.1.2. Service Level Expectations

SLEs define a set of network attributes and characteristics that describe an IETF Network Slice service, but which are not directly measurable by the customer (e.g. diversity, isolation, and geographical restrictions). Even though the delivery of an SLE cannot usually be determined by the customer, the SLEs form an important part of the contract between customer and provider.

Quite often, an SLE will imply some details of how an IETF Network Slice service is realized by the provider, although most aspects of the implementation in the underlying network layers remain a free choice for the provider. For example, activating unicast or multicast capabilities to deliver an IETF Network Slice service could be explicitly requested by a customer or could be left as an engineering decision for the service provider based on capabilities of the network and operational choices.

SLEs may be seen as aspirational on the part of the customer, and they are expressed as behaviors that the provider is expected to apply to the network resources used to deliver the IETF Network Slice service. Of course, over time, it is possible that mechanisms will be developed that enable a customer to verify the provision of an SLE, at which point it effectively becomes an SLO.

An IETF Network Slice service may include multiple connectivity constructs that associate sets of endpoints (SDPs). SLEs apply to a given connectivity construct and apply to specific directions of traffic flow. That is, they apply to a specific sending SDP and the
set of receiving SDPs. However, being more general in nature than SLOs, SLEs may commonly be applied to all connectivity constructs in an IETF Network Slice service.

4.1.2.1. Some Common SLEs

SLEs can be described as ‘Indirectly Measurable Objectives’: they are not generally directly measurable by the customer.

Security, geographic restrictions, maximum occupancy level, and isolation are example SLEs as follows.

Security: A customer may request that the provider applies encryption or other security techniques to traffic flowing between SDPs of a connectivity construct within an IETF Network Slice service. For example, the customer could request that only network links that have MACsec [MACsec] enabled are used to realize the connectivity construct.

This SLE may include a request for encryption (e.g., [RFC4303]) between the two SDPs explicitly to meet the architectural recommendations in [TS33.210] or for compliance with [HIPAA] or [PCI].

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Please see further discussion on security in Section 9.

Geographic Restrictions: A customer may request that certain geographic limits are applied to how the provider routes traffic for the IETF Network Slice service. For example, the customer may have a preference that its traffic does not pass through a particular country for political or security reasons.

Whether or not the provider has met this SLE is generally not directly observable by the customer and cannot be measured as a quantifiable metric.

Maximal Occupancy Level: The maximal occupancy level specifies the number of flows to be admitted and optionally a maximum number of countable resource units (e.g., IP or MAC addresses) an IETF Network Slice service can consume. Because an IETF Network Slice service may include multiple connectivity constructs, this SLE should state whether it applies to all connectivity constructs, a specified subset of them, or an individual connectivity construct.
Again, a customer may not be able to fully determine whether this SLE is being met by the provider.

Isolation: As described in Section 7, a customer may request that its traffic within its IETF Network Slice service is isolated from the effects of other network services supported by the same provider. That is, if another service exceeds capacity or has a burst of traffic, the customer’s IETF Network Slice service should remain unaffected and there should be no noticeable change to the quality of traffic delivered.

In general, a customer cannot tell whether a service provider is meeting this SLE. They cannot tell whether the variation of an SLI is because of changes in the underlay network or because of interference from other services carried by the network. If the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

Diversity: A customer may request that different connectivity constructs use different underlay network resources. This might be done to enhance the availability of the connectivity constructs within an IETF Network Slice service.

While availability is a measurable objective (see Section 4.1.1.1) this SLE requests a finer grade of control and is not directly measurable (although the customer might become suspicious if two connectivity constructs fail at the same time).

4.2. IETF Network Slice Service Demarcation Points

As noted in Section 3.1, an IETF Network Slice provides connectivity between sets of SDPs with specific SLOs and SLEs. Section 3.2 goes on to describe how the IETF Network Slice service is composed of a set of one or more connectivity constructs that describe connectivity between the Service Demarcation Points (SDPs) across the underlay network.

The characteristics of IETF Network Slice SDPs are as follows.

* An SDP is the point of attachment to an IETF Network Slice. As such, SDPs serve as the IETF Network Slice ingress/egress points.

* An SDP is identified by a unique identifier in the context of an IETF Network Slice customer.
* The provider associates each SDP with a set of provider-scope identifiers such as IP addresses, encapsulation-specific identifiers (e.g., VLAN tag, MPLS Label), interface/port numbers, node ID, etc.

* SDPs are mapped to endpoints of services/tunnels/paths within the IETF Network Slice during its initialization and realization.

  - A combination of the SDP identifier and SDP provider-network-scope identifiers define an SDP in the context of the Network Slice Controller (NSC) (see Section 5.3).

  - The NSC will use the SDP provider-network-scope identifiers as part of the process of realizing the IETF Network Slice.

Note that an ancillary CE (see Section 3.2.1) is the endpoint of a connectivity construct and so has an SDP in this discussion.

For a given IETF Network Slice service, the IETF Network Slice customer and provider agree where the SDP is located. This determines what resources at the edge of the network form part of the IETF Network Slice and are subject to the set of SLOs and SLEs for a specific SDP.

Figure 1 shows different potential scopes of an IETF Network Slice that are consistent with the different SDP locations. For the purpose of this discussion and without loss of generality, the figure shows customer edge (CE) and provider edge (PE) nodes connected by attachment circuits (ACs). Notes after the figure give some explanations.
Explanatory notes for Figure 1 are as follows:

1. If the CE is operated by the IETF Network Slice service provider, then the edge of the IETF Network Slice may be within the CE. In this case the slicing process may utilize resources from within the CE such as buffers and queues on the outgoing interfaces.

2. The IETF Network Slice may be extended as far as the CE, to include the AC, but not to include any part of the CE. In this case, the CE may be operated by the customer or the provider. Slicing the resources on the AC may require the use of traffic tagging (such as through Ethernet VLAN tags) or may require traffic policing at the AC link ends.

3. The SDPs of the IETF Network Slice are the customer-facing ports on the PEs. This case can be managed in a way that is similar to a port-based VPN: each port (AC) or virtual port (e.g., VLAN tag) identifies the IETF Network Slice and maps to an IETF Network Slice SDP.
4. Finally, the SDP may be within the PE. In this mode, the PE classifies the traffic coming from the AC according to information (such as the source and destination IP addresses, payload protocol and port numbers, etc.) in order to place it onto an IETF Network Slice.

The choice of which of these options to apply is entirely up to the network operator. It may limit or enable the provisioning of particular managed services and the operator will want to consider how they want to manage CEs and what control they wish to offer the customer over AC resources.

Note that Figure 1 shows a symmetrical positioning of SDPs, but this decision can be taken on a per-SDP basis through agreement between the customer and provider.

In practice, it may be necessary to map traffic not only onto an IETF Network Slice, but also onto a specific connectivity construct if the IETF Network Slice supports more than one with a source at the specific SDP. The mechanism used will be one of the mechanisms described above, dependent on how the SDP is realized.

Finally, note (as described in Section 2.1) that an SDP is an abstract endpoint of an IETF Network Slice service and as such may be a device, interface, or software component. An ancillary CE (Section 3.2.1) should also be thought of as an SDP.

4.3. IETF Network Slice Composition

Operationally, an IETF Network Slice may be composed of two or more IETF Network Slices as specified below. Decomposed network slices are 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. Framework

A number of IETF Network Slice services will typically be provided over a shared underlay network infrastructure. Each IETF Network Slice consists of both the overlay connectivity and a specific set of dedicated network resources and/or functions allocated in a shared underlay network to satisfy the needs of the IETF Network Slice customer. In at least some examples of underlay network technologies, the integration between the overlay and various underlay resources is needed to ensure the guaranteed performance requested for different IETF Network Slices.

5.1. IETF Network Slice Stakeholders

An IETF Network Slice and its realization involves the following stakeholders. The IETF Network Slice customer and IETF Network Slice provider (see Section 2.1) are also stakeholders.

Orchestrator: An orchestrator is an entity that composes different services, resource, and network requirements. It interfaces with the IETF NSC when composing a complex service such as an end-to-end network slice.

IETF Network Slice Controller (NSC): The NSC realizes an IETF Network Slice in the underlay network, and maintains and monitors the run-time state of resources and topologies associated with it. A well-defined interface is needed to support interworking between different NSC implementations and different orchestrator implementations.

Network Controller: The Network Controller is a form of network infrastructure controller that offers network resources to the NSC to realize a particular network slice. This may be an existing network controller associated with one or more specific technologies that may be adapted to the function of realizing IETF Network Slices in a network.

5.2. Expressing Connectivity Intents

An IETF Network Slice customer communicates with the NSC using the IETF Network Slice Service Interface.

An IETF Network Slice customer may be a network operator who, in turn, uses the IETF Network Slice to provide a service for another IETF Network Slice customer.
Using the IETF Network Slice Service Interface, a customer expresses requirements for a particular slice by specifying what is required rather than how that is to be achieved. That is, the customer’s view of a slice is an abstract one. Customers normally have limited (or no) visibility into the provider network’s actual topology and resource availability information.

This should be true even if both the customer and provider are associated with a single administrative domain, in order to reduce the potential for adverse interactions between IETF Network Slice customers and other users of the underlay network infrastructure.

The benefits of this model can include the following.

* Security: The underlay network components are less exposed to attack because the underlay network (or network operator) does not need to expose network details (topology, capacity, etc.) to the IETF Network Slice customers.

* Layered Implementation: The underlay network comprises network elements that belong to a different layer network than customer applications. Network information (advertisements, protocols, etc.) that a customer cannot interpret or respond to is not exposed to the customer. (Note – a customer should not use network information not exposed via the IETF Network Slice Service Interface, even if that information is available.)

* Scalability: Customers do not need to know any information concerning Network topology, capabilities, or state beyond that which is exposed via the IETF Network Slice Service Interface.

The general issues of abstraction in a TE network are described more fully in [RFC7926].

This framework document does not assume any particular technology layer at which IETF Network Slices operate. A number of layers (including virtual L2, Ethernet or, IP connectivity) could be employed.

Data models and interfaces are needed to set up IETF Network Slices, and specific interfaces may have capabilities that allow creation of slices within specific technology layers.

Layered virtual connections are comprehensively discussed in other IETF documents. See, for instance, GMPLS-based networks [RFC5212] and [RFC4397], or Abstraction and Control of TE Networks (ACTN) [RFC8453] and [RFC8454]. The principles and mechanisms associated with layered networking are applicable to IETF Network Slices.
There are several IETF-defined mechanisms for expressing the need for a desired logical network. The IETF Network Slice Service Interface carries data either in a protocol-defined format, or in a formalism associated with a modeling language.

For instance:

* The Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440] and GMPLS User-Network Interface (UNI) using RSVP-TE [RFC4208] use a TLV-based binary encoding to transmit data.


* gRPC/GNMI [I-D.openconfig-rtgwg-gnmi-spec] uses a binary encoded programmable interface. ProtoBufs can be used to model gRPC and GNMI data.

* For data modeling, YANG ([RFC6020] and [RFC7950]) may be used to model configuration and other data for NETCONF, RESTCONF, and GNMI, among others.

While several generic formats and data models for specific purposes exist, it is expected that IETF Network Slice management may require enhancement or augmentation of existing data models. Further, it is possible that mechanisms will be needed to determine the feasibility of service requests before they are actually made.

5.3. IETF Network Slice Controller (NSC)

The IETF NSC takes abstract requests for IETF Network Slices and implements them using a suitable underlay technology. An IETF NSC is the key component for control and management of the IETF Network Slice. It provides the creation/modification/deletion, monitoring and optimization of IETF Network Slices in a multi-domain, a multi-technology and multi-vendor environment.

The main task of the IETF NSC is to map abstract IETF Network Slice requirements to concrete technologies and establish required connectivity ensuring that resources are allocated to the IETF Network Slice as necessary.

The IETF Network Slice Service Interface is used for communicating details of an IETF Network Slice (configuration, selected policies, operational state, etc.), as well as information about status and performance of the IETF Network Slice. The details for this IETF Network Slice Service Interface are not in scope for this document.
The controller provides the following functions.

* Provides an IETF Network Slice Service Interface for creation/modification/deletion of the IETF Network Slices that is agnostic to the technology of the underlay network. The API exposed by this interface communicates the Service Demarcation Points of the IETF Network Slice, IETF Network Slice SLO/SLE parameters (and possibly monitoring thresholds), applicable input selection (filtering) and various policies, and provides a way to monitor the slice.

* Determines an abstract topology connecting the SDPs of the IETF Network Slice that meets criteria specified via the IETF Network Slice Service Interface. The NSC also retains information about the mapping of this abstract topology to underlay components of the IETF Network Slice as necessary to monitor IETF Network Slice status and performance.

* Provides "Mapping Functions" for the realization of IETF Network Slices. In other words, it will use the mapping functions that:

  - map IETF Network Slice Service Interface requests that are agnostic to the technology of the underlay network to technology-specific network configuration interfaces.

  - map filtering/selection information as necessary to entities in the underlay network so that those entities are able to identify what traffic is associated with which connectivity construct and IETF network slice and necessary according to the realization solution, and how traffic should be treated to meet the SLOs and SLEs of the connectivity construct.

* The controller collects telemetry data (e.g., OAM results, statistics, states, etc.) via a network configuration interface for all elements in the abstract topology used to realize the IETF Network Slice.

* Evaluates the current performance against IETF Network Slice SLO parameters using the telemetry data from the underlying realization of an IETF Network Slice (i.e., services/paths/tunnels). Exposes this performance to the IETF Network Slice customer via the IETF Network Slice Service Interface. The IETF Network Slice Service Interface may also include the capability to provide notifications if the IETF Network Slice performance reaches threshold values defined by the IETF Network Slice customer.
5.3.1. 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 2).

IETF Network Slice Service Interface: The IETF Network Slice Service Interface is an interface between a customer's higher level operation system (e.g., a network slice orchestrator or a customer network management system) and the NSC. It is agnostic to the technology of the underlay network. The customer can use this interface to communicate the requested characteristics and other requirements for the IETF Network Slice, and the NSC can use the interface to report the operational state of an IETF Network Slice to the customer.

Network Configuration Interface: The Network Configuration Interface is an interface between the NSC and network controllers. It is technology-specific and may be built around the many network models already defined within the IETF.

These interfaces can be considered in the context of the Service Model and Network Model described in [RFC8309] and, together with the Device Configuration Interface used by the Network Controllers, provides a consistent view of service delivery and realization.

---

Figure 2: Interfaces of the IETF Network Slice Controller
5.3.1.1. IETF Network Slice Service Interface

The IETF Network Slice Controller provides an IETF Network Slice Service Interface that allows customers to request and monitor IETF Network Slices. Customers operate on abstract IETF Network Slices, with details related to their realization hidden.

The IETF Network Slice Service Interface is also independent of the type of network functions or services that need to be connected, i.e., it is independent of any specific storage, software, protocol, or platform used to realize physical or virtual network connectivity or functions in support of IETF Network Slices.

The IETF Network Slice Service Interface uses protocol mechanisms and information passed over those mechanisms to convey desired attributes for IETF Network Slices and their status. The information is expected to be represented as a well-defined data model, and should include at least SDP and connectivity information, SLO/SLE specification, and status information.

5.3.2. Management Architecture

The management architecture described in Figure 2 may be further decomposed as shown in Figure 3. This should also be seen in the context of the component architecture shown in Figure 4 and corresponds to the architecture in [RFC8309].

Note that the customer higher level operation system of Figure 2 and the Network Slice Orchestrator of Figure 3 may be considered equivalent to the Service Management & Orchestration (SMO) of [ORAN].
6. Realizing IETF Network Slices

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 Network Configuration Interface. However, this section provides an overview of the components and processes involved in realizing an IETF Network Slice.

Figure 3: Interface of IETF Network Slice Management Architecture
6.1. Architecture to Realize IETF Network Slices

The architecture described in this section is deliberately at a high level. It is not intended to be prescriptive: implementations and technical solutions may vary freely. However, this approach provides a common framework that other documents may reference in order to facilitate a shared understanding of the work.

Figure 4 shows the architectural components of a network managed to provide IETF Network Slices. The customer’s view is of individual IETF Network Slices with their SDPs, and connectivity constructs. Requests for IETF Network Slices are delivered to the NSC.

The figure shows, without loss of generality, the CEs, ACs, and PEs, that exist in the network. The SDPs are not shown and can be placed in any of the ways described in Section 4.2.
Figure 4: Architecture of an IETF Network Slice
The network itself (at the bottom of the figure) comprises an underlay network. This could be a physical network, but may be a virtual network. The underlay network is provisioned through network controllers that may utilize device controllers [RFC8309].

The underlay network may optionally be filtered or customized by the network operator to produce a number of network topologies that we call Filter Topologies. Customization is just a way of selecting specific resources (e.g., nodes and links) from the underlay network according to their capabilities and connectivity in the underlay network. These actions are configuration options or operator policies. The resulting topologies can be used as candidates to host IETF Network Slices and provide a useful way for the network operator to know in advance that all of the resources they are using to plan an IETF Network Slice would be able to meet specific SLOs and SLEs. The creation of a Filter Topology could be an offline planning activity or could be performed dynamically as new demands arise. The use of Filter Topologies is entirely optional in the architecture, and IETF Network Slices could be hosted directly on the underlay network.

Recall that an IETF Network Slice is a service requested by / provided for the customer. The IETF Network Slice service is expressed in terms of one or more connectivity constructs. An implementation or operator is free to limit the number of connectivity constructs in an IETF Network Slice to exactly one. Each connectivity construct is associated within the IETF Network Slice service request with a set of SLOs and SLEs. The set of SLOs and SLEs does not need to be the same for every connectivity construct in the IETF Network Slice, but an implementation or operator is free to require that all connectivity constructs in an IETF Network Slice have the same set of SLOs and SLEs.

A Network Resource Partition (NRP) is a collection of resources (bufferage, queuing, scheduling, etc.) in the underlay network. The amount and granularity of resources allocated in an NRP is flexible and depends on the operator’s policy. Some NRP realizations may build NRPs with dedicated topologies, while some other realizations may use a shared topology for multiple NRPs; one possible realization is of a single NRP using all of the resources of the entire underlay network topology. Thus, an NRP consists of a subset of the buffer/queuing/scheduling resources on each of a connected set of links in the underlay network. The connected set of links can be all of the links in the underlay network and in this case there can be a single NRP and it has all of the buffer/queuing/scheduling resources for each of the links in the underlay network.
One or more connectivity constructs from one or more IETF Network Slices are mapped to an NRP. A single connectivity construct is mapped to only one NRP (that is, the relationship is many to one). Thus, all traffic flows in a connectivity construct assigned to an NRP are assigned to that NRP. Further, all PEs connected by a connectivity construct must be present in the NRP to which that connectivity construct is assigned.

An NRP may be chosen to support a specific connectivity construct because of its ability to support a specific set of SLOs and SLEs, or its ability to support particular connectivity types, or for any administrative or operational reason. An implementation or operator is free to map each connectivity construct to a separate NRP, although there may be scaling implications depending on the solution implemented. Thus, the connectivity constructs from one slice may be mapped to one or more NRPs. By implication from the above, an implementation or operator is free to map all the connectivity constructs in a slice to a single NRP, and to not share that NRP with connectivity constructs from another slice. By default, NRPs are work conserving.

The process of determining the NRP may be made easier if the underlay network topology is first filtered into a Filter Topology in order to be aware of the subset of network resources that are suitable for specific NRPs. In this case, each Filter Topology is treated as an underlay network on which NRPs can be constructed. The stage of generating Filter Topologies is optional within this framework.

The steps described here can be applied in a variety of orders according to implementation and deployment preferences. Furthermore, the steps may be iterative so that the components are continually refined and modified as network conditions change and as service requests are received or relinquished, and even the underlay network could be extended if necessary to meet the customers’ demands.

6.2. Procedures to Realize IETF Network Slices

There are a number of different technologies that can be used in the underlay, including physical connections, MPLS, time-sensitive networking (TSN), Flex-E, etc.

An IETF Network Slice can be realized in a network, using specific underlay technology or technologies. The creation of a new IETF Network Slice will be realized with following steps:

* The NSC exposes the network slicing capabilities that it offers for the network it manages so that the customer can determine whether to request services and what features are in scope.
* The customer may issue a request to determine whether a specific IETF Network Slice could be supported by the network. The NSC may respond indicating a simple yes or no, and may supplement a negative response with information about what it could support were the customer to change some requirements.

* The customer requests an IETF Network Slice. The NSC may respond that the slice has or has not been created, and may supplement a negative response with information about what it could support were the customer to change some requirements.

* When processing a customer request for an IETF Network Slice, the NSC maps the request to the network capabilities and applies provider policies before creating or supplementing the NRP.

Regardless of how IETF Network Slice is realized in the network (i.e., using tunnels of different types), the definition of the IETF Network Slice service does not change at all. The only difference is how the slice is realized. The following sections briefly introduce how some existing architectural approaches can be applied to realize IETF Network Slices.

6.3. Applicability of ACTN to IETF Network Slices

Abstraction and Control of TE Networks (ACTN - [RFC8453]) is a management architecture and toolkit used to create virtual networks (VNs) on top of a TE underlay network. The VNs can be presented to customers for them to operate as private networks.

In many ways, the function of ACTN is similar to IETF network slicing. Customer requests for connectivity-based overlay services are mapped to dedicated or shared resources in the underlay network in a way that meets customer guarantees for service level objectives and for separation from other customers' traffic. [RFC8453] describes the function of ACTN as collecting resources to establish a logically dedicated virtual network over one or more TE networks. Thus, in the case of a TE-enabled underlay network, the ACTN VN can be used as a basis to realize IETF network slicing.

While the ACTN framework is a generic VN framework that can be used for VN services beyond the IETF Network Slice, it also a suitable basis for delivering and realizing IETF Network Slices.

Further discussion of the applicability of ACTN to IETF Network Slices including a discussion of the relevant YANG models can be found in [I-D.ietf-teas-applicability-actn-slicing].
6.4. Applicability of Enhanced VPNs to IETF Network Slices

An enhanced VPN (VPN+) is designed to support the needs of new applications, particularly applications that are associated with 5G services, by utilizing an approach that is based on existing VPN and TE technologies and adds characteristics that specific services require over and above VPNs as they have previously been specified.

An enhanced VPN can be used to provide enhanced connectivity services between customer sites and can be used to create the infrastructure to underpin an IETF Network Slice service.

It is envisaged that enhanced VPNs will be delivered using a combination of existing, modified, and new networking technologies.

[I-D.ietf-teas-enhanced-vpn] describes the framework for Enhanced Virtual Private Network (VPN+) services.

6.5. Network Slicing and Aggregation in IP/MPLS Networks

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. The support of resource preemption between IETF network slices is deployment specific.

Many approaches are currently being worked on to support IETF Network Slices in IP and MPLS networks with or without the use of Segment Routing. Most of these approaches utilize a way of marking packets so that network nodes can apply specific routing and forwarding behaviors to packets that belong to different IETF Network Slices. Different mechanisms for marking packets have been proposed (including using MPLS labels and Segment Routing segment IDs) and those mechanisms are agnostic to the path control technology used within the underlay network.

These approaches are also sensitive to the scaling concerns of supporting a large number of IETF Network Slices within a single IP or MPLS network, and so offer ways to aggregate the connectivity constructs of slices (or whole slices) so that the packet markings indicate an aggregate or grouping where all of the packets are subject to the same routing and forwarding behavior.

At this stage, it is inappropriate to mention any of these proposed solutions that are currently work in progress and not yet adopted as IETF work.
6.6. Network Slicing and Service Function Chaining (SFC)

A customer may request an IETF Network Slice service that involves a set of service functions (SFs) together with the order in which these SFs are invoked. Also, the customer can specify the service objectives to be met by the underlying network (e.g., one-way delay to cross a service function path, one-way delay to reach a specific SF). These SFs are considered as ancillary CEs and are possibly placeholders (i.e., the SFs are identified, but not their locators).

Service Function Chaining (SFC) [RFC7665] techniques can be used by a provider to instantiate such an IETF Network Service Slice. The NSC may proceed as follows.

* Expose a set of ancillary CEs that are hosted in the underlay network.
* Capture the SFC requirements (including, traffic performance metrics) from the customer. One or more service chains may be associated with the same IETF Network Slice service as connectivity constructs.
* Execute an SF placement algorithm to decide where to locate the ancillary CEs in order to fulfill the service objectives.
* Generate SFC classification rules to identify (part of) the slice traffic that will be bound to an SFC. These classification rules may be the same as or distinct from the identification rules used to bind incoming traffic to the associated IETF Network Slice.

The NSC also generates a set of SFC forwarding policies that govern how the traffic will be forwarded along a service function path (SFP).

* Identify the appropriate Classifiers in the underlay network and provision them with the classification rules. Likewise, the NSC communicates the SFC forwarding policies to the appropriate Service Function Forwarders (SFF).

The provider can enable an SFC data plane mechanism, such as [RFC8300], [RFC8596], or [I-D.ietf-spring-nsh-sr].

7. Isolation in IETF Network Slices
7.1. Isolation as a Service Requirement

An IETF Network Slice customer may request that the IETF Network Slice delivered to them is such that changes to other IETF Network Slices or to other services do not have any negative impact on the delivery of the IETF Network Slice. The IETF Network Slice customer may specify the degree to which their IETF Network Slice is unaffected by changes in the provider network or by the behavior of other IETF Network Slice customers. The customer may express this via an SLE it agrees with the provider. This concept is termed ‘isolation’.

In general, a customer cannot tell whether a service provider is meeting an isolation SLE. If the service varies such that an SLO is breached then the customer will become aware of the problem, and if the service varies within the allowed bounds of the SLOs, there may be no noticeable indication that this SLE has been violated.

7.2. Isolation in IETF Network Slice Realization

Isolation may be achieved in the underlay network by various forms of resource partitioning ranging from dedicated allocation of resources for a specific IETF Network Slice, to sharing of 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 IETF Network Slices.

8. Management Considerations

IETF Network Slice realization needs to be instrumented in order to track how it is working, and it might be necessary to modify the IETF Network Slice as requirements change. Dynamic reconfiguration might be needed.

The various management interfaces and components are discussed in Section 5.

9. 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 customer-defined IETF Network Slices will be mapped to their realization in the underlay networks. Underlay networks will require capabilities to conform to customer’s requests as some aspects of security may be expressed in SLEs.

IETF NSC authentication: Underlay networks need to be protected against the attacks from an adversary NSC as this could destabilize overall network operations. An IETF Network Slice may span across different networks, therefore, the NSC should have strong authentication with each of these networks. Furthermore, both the IETF Network Slice Service Interface and the Network Configuration Interface 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 customer 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 customer is responsible for end-to-end encryption of its own traffic. While an IETF Network Slice might include encryption and other security features as part of the service (for example as SLEs), customers might be well advised to take responsibility for their own security needs.

Note: See [NGMN_SEC] on 5G network slice security for discussion relevant to this section.

IETF Network Slices might use underlying virtualized networking. All types of virtual networking require special consideration to be given to the separation of traffic between distinct virtual networks, as well as some degree of protection from effects of traffic use of underlay network (and other) resources from other virtual networks sharing those resources.

For example, if a service requires a specific upper bound of latency, then that service can be degraded by added delay in transmission of service packets caused by the activities of another service or application using the same resources.
Similarly, in a network with virtual functions, noticeably impeding access to a function used by another IETF Network Slice (for instance, compute resources) can be just as service-degrading as delaying physical transmission of associated packet in the network.

10. Privacy Considerations

Privacy of IETF Network Slice service customers must be preserved. It should not be possible for one IETF Network Slice customer to discover the presence of other customers, nor should sites that are members of one IETF Network Slice be visible outside the context of that IETF Network Slice.

In this sense, it is of paramount importance that the system use the privacy protection mechanism defined for the specific underlay technologies that support the slice, including in particular those mechanisms designed to preclude acquiring identifying information associated with any IETF Network Slice customer.

11. IANA Considerations

This document makes no requests for IANA action.

Acknowledgments

The entire TEAS Network Slicing design team and everyone participating in related discussions has contributed to this document. Some text fragments in the document have been copied from the [I-D.ietf-teas-enhanced-vpn], for which we are grateful.

Significant contributions to this document were gratefully received from the contributing authors listed in the "Contributors" section. In addition we would like to also thank those others who have attended one or more of the design team meetings, including the following people not listed elsewhere:

* Aihua Guo
* Bo Wu
* Greg Mirsky
* Lou Berger
* Rakesh Gandhi
* Ran Chen
Further useful comments were received from Daniele Ceccarelli, Uma Chunduri, Pavan Beeram, Tarek Saad, Kenichi Ogaki, Oscar Gonzalez de Dios, Xiaobing Niu, Dan Voyer, Igor Bryskin, Luay Jalil, Joel Halpern, John Scudder, John Mullooly, and Krzysztof Szarkowicz.

This work is partially supported by the European Commission under Horizon 2020 grant agreement number 101015857 Secured autonomic traffic management for a Tera of SDN flows (Teraflow).

Contributors

The following authors contributed significantly to this document:

Eric Gray
(The original editor of the foundation documents)
Independent
Email: ewgray@graiymage.com

Jari Arkko
Ericsson
Email: jari.arkko@piuha.net

Mohamed Boucadair
Orange
Email: mohamed.boucadair@orange.com

Dhruv Dhody
Huawei, India
Email: dhruv.ietf@gmail.com

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

Xufeng Liu
Volta Networks
Email: xufeng.liu.ietf@gmail.com
Informative References


Appendix A. Examples

This appendix contains realisation examples. This is not intended to be a complete set of possible deployments. Nor does it provide definitive ways to realise these deployments.

Farrel, et al. Expires 1 January 2023
The examples shown here must not be considered to be normative. The descriptions of terms and concepts in the body of the document take precedence. The examples

A.1. Multi-Point to Point Service

As described in Section 3.2 an MP2P service can be realized with multiple P2P connectivity constructs. Figure 5 shows a simple MP2P service where traffic is sent from any of CE1, CE2, and CE3, to the receiver which is CE4. The service comprises three P2P connectivity constructs CE1-CE4, CE2-CE4, and CE3-CE4.

![Figure 5: Example MP2P Service with P2P Connections](image)

A.2. Service Function Chaining and Ancillary CEs

Section 3.2.1 introduces the concept of ancillary CEs. Figure 6 shows a simple example of IETF Network Slices with connectivity constructs that are used to deliver traffic from CE1 to CE3 taking in a service function along the path.
A customer may want to provide a service where traffic is delivered from CE1 to CE3 including a service function sited within the customer’s network at CE2. To achieve this, the customer may request an IETF Network Slice Service comprising two P2P connectivity constructs (CE1-CE2 and CE2-CE3 represented as *** in the figure).

Alternatively, the service function for the same CE1 to CE3 flow may be hosted at a node within the network operator’s. This is an ancillary CE in the IETF Network Slice Service that the customer requests. This service contains two P2P connectivity constructs (CE1-ACE1 and ACE1-CE3 represented as ooo in the figure). How the customer know of the existence of the ancillary CE and the service functions it offers is a matter for agreement between the customer and the network operator.

Finally, it may be that the customer knows that the network operator is able to provide the service function, but not know the location of the ancillary CE at which the service function is hosted. Indeed, it may be that the service function is hosted at a number of ancillary CEs (ACE2, ACE3, and ACE4 in the figure): the customer may or know the identities of the ancillary CEs, but be unwilling or unable to choose one; or the customer may not know about the ancillary CEs. In this case, the IETF Network Slice Service request contains two P2P
connectivity constructs (CE1-ServiceFunction and ServiceFunction to CE3 represented as xxx in the figure). It is left as a choice for the network operator which ancillary CE to use and how to realise the connectivity constructs.

A.3. Hub and Spoke

Hub and spoke is a popular way to realise any-to-any connectivity in support of multiple P2P traffic flows (where the hub performs routing), or of P2MP flows (where the hub is responsible for replication). In many case, it is the network operator’s choice whether to use hub and spoke to realise a mesh of P2P connectivity constructs or P2MP connectivity constructs: this is entirely their business as the customer is not aware of how the connectivity constructs are supported within the network.

However, it may be the case that the customer wants to control the behavior and location of the hub. In this case, the hub appears as an ancillary CE as shown in Figure 7.

For the P2P mesh case, the customer does not specify a mesh of P2P connectivity constructs (such as CE1-CE2, CE1-CE3, CE2-CE3 and the equivalent reverse direction connectivity), but connects each CE to the hub with P2P connectivity constructs (as CE1-Hub, CE2-Hub, CE3-Hub and the equivalent reverse direction connectivity). This scales better in terms of provisioning compared to a full mesh, but does require that the hub is capable of routing traffic between connectivity constructs.

For the P2MP case, does nor specify a single P2MP connectivity construct (in this case, CE3-{CE1+CE2}), but requests three P2P connectivity constructs (as CE3-Hub, Hub-CE1, and Hub-CE2). It is the hub’s responsibility to replicate the traffic from CE3 and send it to both CE1 and CE2.
A.4. Layer 3 VPN

Layer 3 VPNs are a common service offered by network operators to their customers. They may be modelled as an any-to-any service, but are often realised as a mesh of P2P connections, or if multicast is supported, they may be realised as a mesh of P2MP connections.

Figure 8 shows an IETF Network Slice Service with a single A2A connectivity construct between the SDPs CE1, CE2, CE3, and CE4. It is a free choice how the network operator realises this service. They may use a full mesh of P2P connections, a hub and spoke configuration, or some combination of these approaches.

---

Figure 8: Example L3VPN Service

Authors’ Addresses

Adrian Farrel (editor)
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk

John Drake (editor)
Juniper Networks
United States of America
Email: jdrake@juniper.net
Reza Rokui
Ciena
Email: rrokui@ciena.com

Shunsuke Homma
NTT
Japan
Email: shunsuke.homma.ietf@gmail.com

Kiran Makhijani
Futurewei
United States of America
Email: kiranm@futurewei.com

Luis M. Contreras
Telefonica
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

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

Network slicing can be used to meet the connectivity and performance requirement of different services or customers in a shared network. An IETF network slice may be used for 5G or other network scenarios. In the context of 5G, the 5G end-to-end network slices consist of three major types of network technology domains: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). The transport network slice can be realized as IETF network slices. In the transport network, the IETF network slice may span multiple network administrative domains.

In order to facilitate the mapping between network slices in different network technology domains and administrative domains, it is beneficial to carry the identifiers related to the 5G end-to-end network slice, the multi-domain IETF network slice together with the intra-domain network slice related identifier in the data packet.

This document describes the framework of end-to-end IETF network slicing, and introduces the identifiers related to 5G end-to-end network slice and the multi-domain IETF network slice. These identifiers can be carried in the data packet. The roles of the different identifiers in packet forwarding is also described. The network slice identifiers may be instantiated with different data planes.

Requirements Language

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

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

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

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

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

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

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

Table of Contents

1. Introduction ................................. 3
2. Framework .................................... 4
3. Requirements on E2E IETF Network Slicing ........ 5
   3.1. Data Plane ................................ 6
4. IANA Considerations ........................... 6
5. Security Considerations ..................... 6
6. Acknowledgements ............................ 6
7. References ................................. 7
   7.1. Normative References ..................... 7
   7.2. Informative References ................... 7
Authors’ Addresses ........................... 7
1. Introduction

[I-D.ietf-teas-ietf-network-slices] defines the terminologies and the characteristics of IETF network slices. It also discusses the general framework, the components and interfaces for requesting and operating IETF network slices. A Network Resource Partition (NRP) is a collection of network resources in the underlay network that are available to carry traffic and meet the SLOs and SLEs.

[I-D.ietf-teas-enhanced-vpn] describes the framework and the candidate component technologies for providing enhanced VPN (VPN+) services based on existing VPN and Traffic Engineering (TE) technologies with enhanced characteristics that specific services require above traditional VPNs. It also introduces the concept of Virtual Transport Network (VTN), which is a virtual underlay network consisting of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the underlay, so as to provide the network characteristics required by the customers. Enhanced VPN (VPN+) and VTN can be used for the realization of IETF network slices. In the context of IETF network slicing, NRP can be seen as an instantiation of VTN.

[I-D.dong-teas-nrp-scalability] describes the scalability considerations in the control plane and data plane of NRP, and proposed several suggestions to improve the scalability. In the control plane, it proposes the approach of decoupling the topology and resource attributes of NRP, so that multiple NRPs may share the same topology attributes and the result of topology based path computation. In the data plane, it proposes to carry a global NRP-ID of a network domain in the data packet to determine the set of resources reserved for the corresponding NRP.

An IETF network slice may span multiple network administrative domains. Further in the context of 5G, there are end-to-end network slices which consists of three major types of network technology domains: Radio Access Network (RAN), Transport Network (TN) and Core Network (CN). In order to facilitate the mapping between network slices in different network technology domains and administrative domains, it may be beneficial to carry the identifiers related to the 5G end-to-end network slice, the identifiers of the multi-domain IETF network slices together with the intra-domain network slices related identifiers in the data packet.

This document describes the typical scenarios of end-to-end network slicing, and the framework of concatenating network slices in different network technology domains and administrative domains.
Multiple network slice related identifiers are defined for network slices with different network scopes. These network slice related identifiers can be instantiated using different data planes, such as IPv6 and MPLS.

2. Framework

```
        /----
        |      |
        |  RAN  |
        |      |
        \
      /----
      |      |
      |     / |
      |   /   |
      |  /    |
      \----/ \
     /      \       |
    /       /       |   |
   /       //       |   |
  /       //       |   |
 /       //       |   | Core
\----/        \
    \
    \
    \----/  
```

5G Network Slice

```
IETF Network Slice (VPN+)

Global NRP (VTN)

Local NRP-1  Local NRP-2  Local NRP-3
```

Figure 1. 5G end-to-end network slicing scenario

One typical scenario of 5G end-to-end network slicing is shown in figure 1. The 5G end-to-end network slice is identified by the S-NSSAI (Single Network Slice Selection Assistance Information). In the transport network, the 5G network slice is mapped to an IETF network slice. In a multi-domain transport network, an IETF network slice can be realized with a multi-domain VPN+ service. In the underlay network, the multi-domain VPN+ service can be supported by a multi-domain VTN, which is the concatenation of multiple intra-domain NRPs in different domains. In each domain, a domain-significant NRP-ID can be carried in the packet to identify the set of network resource reserved for the NRP in the corresponding domain. Note this is similar to the Option C mode of inter-domain VPN service [RFC4364]. Using Option A or Option B mode of inter-domain VPN for 5G end-to-end network slicing is also possible, which is out of the scope of the current version of this document.
In order to concatenate multiple domain-wide NRPs into a multi-domain NRP, the global NRP-ID can be carried in the packet, which is used by the domain border nodes to map to the local NRP-IDs in each domain. And in order to facilitate the network slice mapping between RAN, Core network and transport network, the S-NSSAI may be carried in the packet sent to the transport network, which can be used by the transport network to map the 5G end-to-end network slice to the corresponding IETF network slice.

According to the above end-to-end network slicing scenario, there can be three network slice related identifiers in the data packet:

* Domain NRP-ID: This is the NRP-ID as defined in [I-D.dong-teas-nrp-scalability]. It is used by the network nodes in a network domain to determine the set of local network resources reserved for an NRP. It SHOULD be processed by each hop along the path in the domain.

* End-to-end NRP-ID: This is the identifier which uniquely identifies a multi-domain NRP. In each network domain, the domain border nodes map the global NRP-ID to the domain NRP-ID for packet forwarding.

* 5G end-to-end network slice ID (S-NSSAI): This is the identifier of the 5G end-to-end network slice. When required, it may be used by the network nodes to provide traffic monitoring at the end-to-end network slice granularity.

For the above network slice identifiers, the domain NRP-ID is mandatory, the global NRP-ID and the 5G S-NSSAI are optional. The existence of the Global NRP-ID depends on whether the NRP spans multiple network domains in the transport network, and how the domain NRP-IDs are managed. In some network scenarios, different network domains can have consistent NRP ID allocation, then the domain NRP-ID can has the same value as a global NRP-ID. The existence of the 5G S-NSSAI depends on whether an IETF network slice is used as part of the 5G end-to-end network slice.

3. Requirements on E2E IETF Network Slicing

This section lists the requirements on E2E IETF network slicing.
3.1. Data Plane

To facilitate the mapping between 5G end-to-end network slice and IETF network slice, and the mapping between multi-domain IETF network slice and the intra-domain IETF network slice, different network slice related identifiers, including the S-NSSAI, the Global NRP-ID, domain NRP-ID need to be carried in the data packet.

In a multi-domain IETF network slice, the domain border nodes should support to map the Global NRP-ID to the domain NRP-ID of the local domain. In a 5G end-to-end network slicing scenario, the edge nodes of IETF network slice should support to map the S-NSSAI to the global NRP-ID and the domain NRP-ID. When the correlation between S-NSSAI and the NRP-ID needs to be maintained, the edge nodes of IETF network slices should be able to derive the S-NSSAI from the data packet received from RAN and CN, and encapsulate both the S-NSSAI and the NRP-ID into an outer packet header when traversing the transport network domains.

3.2. Management Plane/Control Plane

For multi-domain IETF network slice, a centralized IETF network slice controller is responsible for the allocation of the Global NRP-ID and the domain NRP-IDs, and the provisioning of the mapping relationship between the Global NRP-ID and the domain NRP-IDs to the border nodes in different network domains.

For 5G end-to-end network slice, when S-NSSAI is used for the mapping from RAN or CN network slices to IETF network slices, the IETF network slice controller is responsible for the provisioning of the mapping relationship between S-NSSAI and the Global and local NRP-IDs at the edge nodes of IETF network slices.

4. IANA Considerations

This document makes no request of IANA.

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

5. Security Considerations

TBD

6. Acknowledgements

TBD
7. References

7.1. Normative References

[I-D.ietf-teas-enhanced-vpn]

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


7.2. Informative References

[I-D.dong-teas-nrp-scalability]


Authors’ Addresses

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

Email: lizhenbin@huawei.com

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing
100095
China
Email: jie.dong@huawei.com

Ran Pang
China Unicom
Email: pangran@chinaunicom.cn

Yongqing Zhu
China Telecom
Email: zhuyq8@chinatelecom.cn
IETF Network Slice YANG Data Model
draft-liu-teas-transport-network-slice-yang-05

Abstract

This document describes a YANG data model for managing and controlling IETF network slices, defined in [I-D.ietf-teas-ietf-network-slices].

Status of This Memo

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

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

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

This Internet-Draft will expire on September 7, 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .............................................. 3
   1.1. Terminology .................................. 3
   1.2. Tree Diagrams .................................. 4
2. Modeling Considerations .................................. 4
   2.1. Relationships to Related Topology Models ........... 4
   2.2. Network Slice with TE ............................ 5
   2.3. ACTN for Network Slicing .......................... 6
3. Model Applicability ...................................... 6
   3.1. Network Slicing by Virtualization ................. 7
   3.2. Network Slicing by TE Overlay ..................... 9
4. Model Communication Types ................................ 11
   4.1. P2P ............................................ 11
   4.2. P2MP ......................................... 12
   4.3. MP2MP ......................................... 13
   4.4. A2A ........................................... 15
5. Model Tree Structure ..................................... 16
   5.1. Module ietf-network-slice ......................... 16
   5.2. Module ietf-network-slice-connectivity ............ 17
6. YANG Modules ............................................ 17
   6.1. Module ietf-network-slice ......................... 17
   6.2. Module ietf-network-slice-connectivity ............ 23
7. IANA Considerations ...................................... 26
8. Security Considerations .................................. 27
9. Acknowledgements ....................................... 28
10. References ........................................... 29
    10.1. Normative References ........................... 29
    10.2. Informative References .......................... 30
Appendix A. Data Tree for the Example in Section 3.1. .... 32
   A.1. Native Topology .................................. 32
   A.2. Network Slice Blue ............................... 36
Authors’ Addresses ......................................... 42

1. Introduction

This document defines a YANG [RFC7950] data model for representing, managing, and controlling IETF network slices, defined in [I-D.ietf-teas-ietf-network-slices]

The defined data model is an interface between customers and providers for configurations and state retrievals, so as to support network slicing as a service. Through this model, a customer can learn the slicing capabilities and the available resources of the provider. A customer can request or negotiate with a network slicing provider to create an instance. The customer can incrementally update its requirements on individual topology elements in the slice instance, and retrieve the operational states of these elements. With the help of other mechanisms and data models defined in IETF, the telemetry information can be published to the customer.

As described in Section 3 of [I-D.contreras-teas-slice-controller-models], the data model defined in this document complements the data model defined in [I-D.ietf-teas-ietf-network-slice-nbi-yang]. In addition to the provider's view, the data model defined in this document models the Type 2 service defined in [RFC8453].

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

1.1. Terminology

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

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

- augment
- data model
- data node
1.2. Tree Diagrams

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

2. Modeling Considerations

An IETF network slice is modeled as network topology defined in [RFC8345], with augmentations. A new network type "network-slice" is defined in this document. When a network topology data instance contains the network-slice network type, it represents an instance of an IETF network slice.

2.1. Relationships to Related Topology Models

There are several related YANG data models that have been defined in IETF. Some of these are:

Network Topology Model:
   Defined in [RFC8345].

OTN Topology Model:
   Defined in [I-D.ietf-ccamp-otn-topo-yang].

L2 Topology Model:
   Defined in [I-D.ietf-i2rs-yang-l2-network-topology].

L3 Topology Model:
   Defined in [RFC8346].

TE Topology Model:
   Defined in [RFC8795].

Figure 1 shows the relationships among these models. The box of dotted lines denotes the model defined in this document.
2.2. Network Slice with TE

In many situations, an IETF network slice needs to have TE (Traffic Engineering) capabilities to achieve certain network characteristics. The TE Topology Model defined in [RFC8795] can be used to make an IETF network slice TE capable. To achieve this, an IETF network slice instance will be configured to have both "network-slice" and "te-topology" network types, taking advantage of the multiple inheritance capability featured by the network topology model [RFC8345]. The following diagram shows their relations.
This method can be applied to other types of network topology models too. For example, when a network topology instance is configured to have the types of "network-slice" defined in this document, "te-topology" defined in [RFC8795], and "l3-unicast-topology" defined in [RFC8346], this network topology instance becomes an IETF network slice instance that can perform layer 3 traffic engineering.

2.3. ACTN for Network Slicing

Since ACTN topology data models are based on the network topology model defined in [RFC8345], the augmentations defined in this document are effective augmentations to the ACTN topology data models, resulting in making the ACTN framework [RFC8453] and data models [I-D.ietf-teas-actn-yang] capable of slicing networks with the required network characteristics.

3. Model Applicability

There are many technologies to achieve network slicing. The data model defined in this document can be applied to a wide range of cases. This section describes how this data model is applied to a few cases.
3.1. Network Slicing by Virtualization

In the case shown in Figure 3, node virtualization is used to separate and allocate resources in physical devices. Two virtual routers VR1 and VR2 are created over physical router R1. Each of the virtual routers takes a portion of the resources such as ports and memory in the physical router. Depending on the requirements and the implementations, they may share certain resources such as processors, ASICs, and switch fabric.

As an example, Appendix A. shows the JSON encoded data instances of the native topology and the customized topology for Network Slice Blue.
Figure 3: Network Slicing by Virtualization
3.2. Network Slicing by TE Overlay

Figure 4 shows a case where TE (Traffic Engineering) overlay is applied to achieve logically separated customer IETF network slices. In the underlay TE capable network, TE tunnels are established to support the TE links in the overlay network. These links and tunnels maintain the characteristics required by the customers. The provider selects the proper logical nodes and links in the overlay network, assigns them to specific IETF network slices, and uses the data model defined in this document to send the results to the customers.
Customers

---

Provider

Customized Topology
Provider Network with TE Isolation

Network Slice Blue: R1, R2, R3

---

Network Slice Red: R1, R2, R4

Overlay

Underlay

Native Topology
Provider Network with TE Tunnels

Figure 4: Network Slicing by TE Overlay
4. Model Communication Types

Section 3.2 of [I-D.ietf-teas-ietf-network-slices] describes various communication types that an IETF network slice may serve, including P2P, P2MP, MP2P, MP2MP, and A2A. The data models specified in [RFC8345] and [RFC8795] support only P2P and A2A. In this document, the YANG module ietf-network-slice-connectivity is defined to extend the capabilities to cover P2MP, MP2P, and MP2MP.

The YANG module ietf-network-slice-connectivity is defined in Section 6.2 of this document, with its structure shown in Section 5.2 of this document. This YANG module introduces two modeling constructs in each connectivity construct (that is called connectivity matrix entries in [RFC8795]):

Replication Group:
A replication group contains a list of connectivity constructs (that are called connectivity matrix entries in RFC 8795). When traffic is sent to one entry in this replication group, the traffic is replicated to all other entries in the same replication group.

Receiver Constraint Group:
A receiver constraint group contains a list of connectivity constructs (that are called connectivity matrix entries in RFC 8795). When traffic is sent to one or more entries in this receiver constraint group, the constraints specified in this receiver constraint group are applied to the receiver-side termination points referenced by all entries in this receiver constraint group.

The following sections describe some data examples:

4.1. P2P
NSE3 <-> NSC7 : Bidirectional P2P connectivity
NSE4  -> NSE8 : Unidirectional P2P connectivity

```
{
  "connectivity-matrices": {
    "connectivity-matrix": [
      "id": 1,
      "from": {
        "tp-ref": "NSE3"
      },
      "to": {
        "tp-ref": "NSE7"
      }
    ],
    "connectivity-matrix": [
      "id": 2,
      "from": {
        "tp-ref": "NSE7"
      },
      "to": {
        "tp-ref": "NSE3"
      }
    ],
    "connectivity-matrix": [
      "id": 3,
      "from": {
        "tp-ref": "NSE4"
      },
      "to": {
        "tp-ref": "NSE8"
      }
    ]
  }
}
```

4.2. P2MP
4.3. MP2MP

{NSE14, NSE15} -> {NSE16, NSE17}

{ "connectivity-matrices": { "connectivity-matrix": [ "id": 1, "from": { "tp-ref": "NSE14" }, "to": { "tp-ref": "NSE16" } ], "connectivity-matrix": [ "id": 2, "from": { "tp-ref": "NSE14" }, "to": { "tp-ref": "NSE15" } ] } }
    "tp-ref": "NSE14"
},
    "to": {
        "tp-ref": "NSE17"
    }
],
"connectivity-matrix": [
    "id": 3,
    "from": {
        "tp-ref": "NSE15"
    },
    "to": {
        "tp-ref": "NSE16"
    }
],
"connectivity-matrix": [
    "id": 4,
    "from": {
        "tp-ref": "NSE15"
    },
    "to": {
        "tp-ref": "NSE17"
    }
],
"replication-group": [
    "id": 1,
    "entry": [1, 2]
],
"replication-group": [
    "id": 2,
    "entry": [3, 4]
],
"receiver-constraint-group": [
    "id": 1,
    "entry": [1, 3]
],
"receiver-constraint-group": [
    "id": 2,
    "entry": [2, 4]
]"}
4.4. A2A

\{NSE1, NSE2, NSE6\} -> \{NSE1, NSE2, NSE6\}

{
    "connectivity-matrices": {
        "connectivity-matrix": [
            "id": 1,
            "from": {
                "tp-ref": "NSE1"
            },
            "to": {
                "tp-ref": "NSE2"
            }
        ],
        "connectivity-matrix": [
            "id": 2,
            "from": {
                "tp-ref": "NSE1"
            },
            "to": {
                "tp-ref": "NSE6"
            }
        ],
        "connectivity-matrix": [
            "id": 3,
            "from": {
                "tp-ref": "NSE2"
            },
            "to": {
                "tp-ref": "NSE1"
            }
        ],
        "connectivity-matrix": [
            "id": 4,
            "from": {
                "tp-ref": "NSE2"
            },
            "to": {
                "tp-ref": "NSE6"
            }
        ],
        "connectivity-matrix": [
            "id": 5,
            "from": {
                "tp-ref": "NSE6"
            }
        ]
    }
}
"to": {
  "tp-ref": "NSE1"
}
],
"connectivity-matrix": [
  "id": 6,
  "from": {
    "tp-ref": "NSE6"
  },
  "to": {
    "tp-ref": "NSE2"
  }
]
}
}

5. Model Tree Structure

5.1. Module ietf-network-slice

TODO - Complete IETF network slice attributes that are technology-agnostic and common to all use cases.

module: ietf-network-slice
  augment /nw:networks/nw:network/nw:network-types:
  +--rw network-slice!
  augment /nw:networks/nw:network:
  +--rw network-slice
      +--rw optimization-criterion? identityref
      +--rw delay-tolerance? boolean
      +--rw periodicity* uint64
      +--rw isolation-level? identityref
  augment /nw:networks/nw:network/nw:node:
  +--rw network-slice
      +--rw isolation-level? identityref
      +--rw compute-node-id? string
      +--rw storage-id? string
  augment /nw:networks/nw:network/nt:link:
  +--rw network-slice
      +--rw delay-tolerance? boolean
      +--rw periodicity* uint64
      +--rw isolation-level? identityref
5.2. Module ietf-network-slice-connectivity

module: ietf-network-slice-connectivity
  augment /nw:networks/nw:network/nw:node/tet:te
  /tet:te-node-attributes/tet:connectivity-matrices
  /tet:connectivity-matrix:
    +++rw replication-group* [id]
      +++rw id       uint32
      +++rw entry*   -> ../../../tet:id
    +++rw receiver-constraint-group* [id]
      +++rw id       uint32
      +++rw entry*   -> ../../../tet:id
      +++rw te-bandwidth
      +++rw (technology)?
      +++:(generic)
      +++rw generic?  te-bandwidth
  augment /nw:networks/nw:network/nw:node/tet:te
  /tet:information-source-entry/tet:connectivity-matrices
  /tet:connectivity-matrix:
    +++ro replication-group* [id]
      +++ro id       uint32
      +++ro entry*   -> ../../../tet:id
    +++ro receiver-constraint-group* [id]
      +++ro id       uint32
      +++ro entry*   -> ../../../tet:id
      +++ro te-bandwidth
      +++ro (technology)?
      +++:(generic)
      +++ro generic?  te-bandwidth

6. YANG Modules

6.1. Module ietf-network-slice

This module references [RFC8345], [RFC8776], and [GSMA-NS-Template]
import ietf-network-topology {
    prefix "nt";
    reference "RFC 8345: A YANG Data Model for Network Topologies";
}
import ietf-te-types {
    prefix "te-types";
    reference  
        "RFC 8776: Traffic Engineering Common YANG Types";
}

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

contact
    "WG Web:  <http://tools.ietf.org/wg/teas/>
    WG List:  <mailto:teas@ietf.org>
    Editor:  Xufeng Liu
              <mailto:xufeng.liu.ietf@gmail.com>
    Editor:  Jeff Tantsura
              <mailto:jefftant.ietf@gmail.com>
    Editor:  Igor Bryskin
              <mailto:i_bryskin@yahoo.com>
    Editor:  Luis Miguel Contreras Murillo
              <mailto:luismiguel.contrerasmurillo@telefonica.com>
    Editor:  Qin Wu
              <mailto:bill.wu@huawei.com>
    Editor:  Sergio Belotti
              <mailto:sergio.belotti@nokia.com>
    Editor:  Reza Rokui
              <mailto:reza.rokui@nokia.com>
";

description
    "YANG data model for representing and managing network
     slices.

Copyright (c) 2020 IETF Trust and the persons identified as
authors of the code. All rights reserved."
Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info).

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

revision 2020-11-01 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/* Identities */

identity isolation-level {
  description "Base identity for the isolation-level.";
  reference
    "GSMA-NS-Template: Generic Network Slice Template, Version 3.0.";
}

identity no-isolation {
  base isolation-level;
  description
    "Network slices are not separated.";
}

identity physical-isolation {
  base isolation-level;
  description
    "Network slices are physically separated (e.g. different rack, different hardware, different location, etc.).";
}

identity logical-isolation {
  base isolation-level;
  description
    "Network slices are logically separated.";
}

identity process-isolation {
  base physical-isolation;
  description
    "Process and threads isolation.";
}

identity physical-memory-isolation {
base physical-isolation;
description
  "Process and threads isolation.";
}
identity physical-network-isolation {
  base physical-isolation;
description
  "Process and threads isolation.";
}
identity virtual-resource-isolation {
  base logical-isolation;
description
  "A network slice has access to specific range of resources
  that do not overlap with other network slices
  (e.g. VM isolation).";
}
identity network-functions-isolation {
  base logical-isolation;
description
  "NF (Network Function) is dedicated to the network slice, but
  virtual resources are shared.";
}
identity service-isolation {
  base logical-isolation;
description
  "NSC data are isolated from other NSCs, but virtual
  resources and NFs are shared.";
}

/*
 * Groupings
 */
grouping network-slice-topology-attributes {
  description "Network Slice topology scope attributes.";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf optimization-criterion {
      type identityref {
        base te-types:objective-function-type;
      }
      description
        "Optimization criterion applied to this topology.";
    }
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
leaf-list periodicity {
  type uint64;
  units seconds;
  description "A list of periodicities supported by the network slice."
  reference "GSMA-NS-Template: Generic Network Slice Template, Version 3.0.";
}
leaf isolation-level {
  type identityref {
    base isolation-level;
  }
  description "A network slice instance may be fully or partly, logically
  and/or physically, isolated from another network slice instance. This attribute describes different types of
  isolation:";
}
} // network-slice
} // network-slice-topology-attributes

grouping network-slice-node-attributes {
  description "Network Slice node scope attributes.";
  container network-slice {
    description "Containing Network Slice attributes.";
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description "A network slice instance may be fully or partly, logically
      and/or physically, isolated from another network slice instance. This attribute describes different types of
      isolation:";
    }
    leaf compute-node-id {
      type string;
      description "Reference to a compute node instance specified in
      a data model specifying the computing resources.";
    }
  }
} // network-slice-node-attributes
leaf storage-id {
  type string;
  description
    "Reference to a storage instance specified in
    a data model specifying the storage resources.";
}
} // network-slice
} // network-slice-node-attributes

grouping network-slice-link-attributes {
  description "Network Slice link scope attributes";
  container network-slice {
    description
      "Containing Network Slice attributes.";
    leaf delay-tolerance {
      type boolean;
      description
        "'true' if is not too critical how long it takes to deliver
        the amount of data.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf-list periodicity {
      type uint64;
      units seconds;
      description
        "A list of periodicities supported by the network slice.";
      reference
        "GSMA-NS-Template: Generic Network Slice Template,
        Version 3.0.";
    }
    leaf isolation-level {
      type identityref {
        base isolation-level;
      }
      description
        "A network slice instance may be fully or partly, logically
        and/or physically, isolated from another network slice
        instance. This attribute describes different types of
        isolation:";
    }
  } // network-slice
} // network-slice
} // network-slice-link-attributes

/*
 * Data nodes
 */
augment "/nw:networks/nw:network/nw:network-types" {
  description "Defines the Network Slice topology type.";
  container network-slice {
    presence "Indicates Network Slice topology";
    description "Its presence identifies the Network Slice type.";
  }
}

augment "/nw:networks/nw:network" {
  when "nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment topology configuration and state.";
  uses network-slice-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
  when ".../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment node configuration and state.";
  uses network-slice-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when ".../nw:network-types/ns:network-slice" {
    description "Augment only for Network Slice topology.";
  }
  description "Augment link configuration and state.";
  uses network-slice-link-attributes;
}

6.2. Module ietf-network-slice-connectivity

This module references [RFC8345], [RFC8776], and [RFC8795]
+ "ietf-network-slice-connectivity";
prefix "ns-con-types";

import ietf-network {
  prefix "nw";
  reference "RFC 8345: A YANG Data Model for Network Topologies";
}
import ietf-te-topology {
  prefix "tet";
  reference "RFC 8795: YANG Data Model for Traffic Engineering (TE) Topologies";
}
import ietf-te-types {
  prefix "te-types";
  reference "RFC 8776: Traffic Engineering Common YANG Types";
}

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

contact "WG Web: <http://tools.ietf.org/wg/teas/>
WG List: <mailto:teas@ietf.org>
Editor:  Xufeng Liu
<mailto:xufeng.liu.ietf@gmail.com>
Editor:  Luis Miguel Contreras Murillo
<mailto:luismiguel.contrerasmurillo@telefonica.com>
Editor:  Sergio Belotti
<mailto:sergio.belotti@nokia.com>
";

description "YANG augmentations to support various connectivity types for IETF network slices.

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

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

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

revision 2022-03-04 {
  description "Initial revision";
  reference
    "RFC XXXX: YANG Data Model for Network Slices";
}

/**
 * Groupings
 */

grouping network-slice-connectivity-types {
  description "Network Slice topology scope attributes.";
  list replication-group {
    key "id";
    description "A list of replication groups. Each replication group
        contains a list of connectivity constructs
        (that are called connectivity matrix entries in RFC 8795).
        When traffic is sent to one entry in this replication group,
        the traffic is replicated to all other entries in the same
        replication group.";
    leaf id {
      type uint32;
      description "Identifies the replication group.";
    }
    leaf-list entry {
      type leafref {
        path "../../tet:id";
      }
      description "References a connectivity matrix entry that belongs to
          this replication group.";
    }
  }
  list receiver-constraint-group {
    key "id";
    description "A list of receiver constraint groups. Each receiver
        constraint group contains a list of connectivity constructs
        (that are called connectivity matrix entries in RFC 8795).
        When traffic is sent to one or more entries in this
        receiver constraint group, the constraints specified in this

receiver constraint group are applied to the receiver-side termination points referenced by all entries in this receiver constraint group.

leaf id {
  type uint32;
  description
    "Identifies the receiver constraint group.";
}
leaf-list entry {
  type leafref {
    path "../../tet:id";
  }
  description
    "References a connectivity matrix entry that belongs to this receiver constraint group.";
}
uses te-types:te-bandwidth;
}

/*
 * Data nodes
 */
+ "tet:te-node-attributes/tet:connectivity-matrices/
+ "tet:connectivity-matrix" {
  description "Augment node configuration and state."
  uses network-slice-connectivity-types;
}
+ "tet:information-source-entry/tet:connectivity-matrices/
+ "tet:connectivity-matrix" {
  description "Augment node configuration and state."
  uses network-slice-connectivity-types;
}
}<CODE ENDS>

7. IANA Considerations

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

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:
This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

- **name:** ietf-l3-te-topology
- **namespace:** urn:ietf:params:xml:ns:yang:ietf-network-slice
- **prefix:** ns
- **reference:** RFC XXXX

8. Security Considerations

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

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

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

  This subtree specifies the network slice type. Modifying the configurations can make network slice type invalid and cause interruption to IETF network slices.

- `/nw:networks/nw:network/ns:network-slice`  
  This subtree specifies the topology-wide configurations. Modifying the configurations here can cause traffic...
characteristics changed in this IETF network slice and related networks.

This subtree specifies the configurations of the nodes in an IETF network slice. Modifying the configurations in this subtree can change the traffic characteristics on this node and the related networks.

This subtree specifies the configurations of the links in an IETF network slice. Modifying the configurations in this subtree can change the traffic characteristics on this link and the related networks.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

Unauthorized access to this subtree can disclose the network slice type.

/nw:networks/nw:network/ns:network-slice
Unauthorized access to this subtree can disclose the topology-wide states.

Unauthorized access to this subtree can disclose the operational state information of the nodes in an IETF network slice.

Unauthorized access to this subtree can disclose the operational state information of the links in an IETF network slice.

9. Acknowledgements

The TEAS Network Slicing Design Team (NSDT) members included Aijun Wang, Dong Jie, Eric Gray, Jari Arkko, Jeff Tantsura, John E Drake, Luis M. Contreras, Raksh Ghand, Ran Chen, Reza Rokui, Ricard Vilalta, Ron Bonica, Sergio Belotti, Tomonobu Niwa, Xuesong Geng, and Xufeng Liu.
10. References

10.1. Normative References


10.2. Informative References


Appendix A. Data Tree for the Example in Section 3.1.

A.1. Native Topology

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network" for the native topology depicted in Figure 3.

```json
{
    "ietf-network:networks": {
        "network": [
            {
                "network-id":"example-native-topology",
                "network-types": {
                },
                "node": [
                    {
                        "node-id":"R1",
                        "ietf-network-topology:termination-point": [
                            {
                                "tp-id":"1-0-1"
                            },
                            {
                                "tp-id":"1-0-2"
                            },
                            {
                                "tp-id":"1-2-1"
                            },
                            {
                                "tp-id":"1-2-2"
                            }
                        ]
                    },
                    {
                        "node-id":"R2",
                        "ietf-network-topology:termination-point": [
                            {
                                "tp-id":"2-1-1"
                            },
                            {
                                "tp-id":"2-1-2"
                            },
                            {
                                "tp-id":"2-3-1"
                            },
                            {
                                "tp-id":"2-4-1"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}
```
"node-id":"R3",
"ietf-network-topology:termination-point": [
  {
    "tp-id":"3-0-1"
  },
  {
    "tp-id":"3-2-1"
  }
],
"node-id":"R4",
"ietf-network-topology:termination-point": [
  {
    "tp-id":"4-0-1"
  },
  {
    "tp-id":"4-2-1"
  }
],
"ietf-network-topology:link": [
  {
    "link-id":"R1,1-0-1,,",
    "source": {
      "source-node":"R1",
      "source-tp":"1-0-1"
    }
  },
  {
    "link-id":",,R1,1-0-1",
    "destination": {
      "dest-node":"R1",
      "dest-tp":"1-0-1"
    }
  },
  {
    "link-id":"R1,1-0-2,,",
    "source": {
      "source-node":"R1",
      "source-tp":"1-0-2"
    }
  }
]
"link-id": ",,R1,1-0-2",
"destination": {
  "dest-node": "R1",
  "dest-tp": "1-0-2"
}
}

{"link-id": "R1,1-2-1,R2,2-1-1",
"source": {
  "source-node": "R1",
  "source-tp": "1-2-1"
}
,"destination": {
  "dest-node": "R2",
  "dest-tp": "2-1-1"
}}

{"link-id": "R2,2-1-1,R1,1-2-1",
"source": {
  "source-node": "R2",
  "source-tp": "2-1-1"
}
,"destination": {
  "dest-node": "R1",
  "dest-tp": "1-2-1"
}}

{"link-id": "R1,1-2-2,R2,2-1-2",
"source": {
  "source-node": "R1",
  "source-tp": "1-2-2"
}
,"destination": {
  "dest-node": "R2",
  "dest-tp": "2-1-2"
}}

{"link-id": "R2,2-1-2,R1,1-2-2",
"source": {
  "source-node": "R2",
  "source-tp": "2-1-2"
}
,"destination": {
  "dest-node": "R1",
  "dest-tp": "1-2-2"}
"link-id":"R2,2-3-1,R3,3-2-1",
"source": {
  "source-node":"R2",
  "source-tp":"2-3-1"
},
"destination": {
  "dest-node":"R3",
  "dest-tp":"3-2-1"
}
},

"link-id":"R3,3-2-1,R2,2-3-1",
"source": {
  "source-node":"R3",
  "source-tp":"3-2-1"
},
"destination": {
  "dest-node":"R2",
  "dest-tp":"2-3-1"
}
},

"link-id":"R2,2-4-1,R4,4-2-1",
"source": {
  "source-node":"R2",
  "source-tp":"2-4-1"
},
"destination": {
  "dest-node":"R4",
  "dest-tp":"4-2-1"
}
},

"link-id":"R4,4-2-1,R2,2-4-1",
"source": {
  "source-node":"R4",
  "source-tp":"4-2-1"
},
"destination": {
  "dest-node":"R2",
  "dest-tp":"2-4-1"
}
},

"link-id":"R3,3-0-1,,",

A.2. Network Slice Blue

This section contains an example of an instance data tree in the JSON encoding [RFC7951]. The example instantiates "ietf-network-slice" for the topology customized for Network Slice Blue depicted in Figure 3.

{  "ietf-network:networks": {  "network": [  {  "network-id":"example-customized-blue-topology",  "network-types": {  "ietf-network-slice:network-slice": {  


"supporting-network": [
  {
    "network-ref":"example-native-topology"
  }
],
"node": [
  {
    "node-id":"VR1",
    "supporting-node": [
      {
        "network-ref":"example-native-topology",
        "node-ref":"R1"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
      "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id":"1-0-1"
      },
      {
        "tp-id":"1-3-1"
      }
    ]
  },
  {
    "node-id":"VR3",
    "supporting-node": [
      {
        "network-ref":"example-native-topology",
        "node-ref":"R2"
      }
    ],
    "ietf-network-slice:network-slice": {
      "isolation-level":
      "ietf-network-slice:physical-memory-isolation"
    },
    "ietf-network-topology:termination-point": [
      {
        "tp-id":"3-1-1"
      },
      {
        "tp-id":"3-5-1"
      }
    ]
  }


"supporting-link": [
    {
        "network-ref":"example-native-topology",
        "link-ref":"R1,1-0-1"
    }
],
"ietf-network-slice:network-slice": {
    "isolation-level":
    "ietf-network-slice:physical-network-isolation"
}
},
{
    "link-id":"VR1,1-3-1,VR3,3-1-1",
    "source": {
        "source-node":"VR1",
        "source-tp":"1-3-1"
    },
    "destination": {
        "dest-node":"VR3",
        "dest-tp":"3-1-1"
    },
    "supporting-link": [
        {
            "network-ref":"example-native-topology",
            "link-ref":"R1,1-2-1,R2,2-1-1"
        }
    ],
    "ietf-network-slice:network-slice": {
        "isolation-level":
        "ietf-network-slice:physical-network-isolation"
    }
}
},
{
    "link-id":"VR3,3-1-1,VR1,1-3-1",
    "source": {
        "source-node":"VR3",
        "source-tp":"3-1-1"
    },
    "destination": {
        "dest-node":"R1",
        "dest-tp":"1-3-1"
    },
    "supporting-link": [
        {
            "network-ref":"example-native-topology",
            "link-ref":"R2,2-1-1,R1,1-2-1"
        }
    ]
}
"ietf-network-slice:network-slice": {
  "isolation-level":
  "ietf-network-slice:physical-network-isolation"
},

"link-id":"VR3,3-5-1,VR5,5-3-1",
"source": {
  "source-node":"VR3",
  "source-tp":"3-5-1"
},
"destination": {
  "dest-node":"VR5",
  "dest-tp":"5-3-1"
},
"supporting-link": [
  {
    "network-ref":"example-native-topology",
    "link-ref":"R2,2-3-1,R3,3-2-1"
  }
],
"ietf-network-slice:network-slice": {
  "isolation-level":
  "ietf-network-slice:physical-network-isolation"
},

"link-id":"VR5,5-3-1,VR3,3-5-1",
"source": {
  "source-node":"VR5",
  "source-tp":"5-3-1"
},
"destination": {
  "dest-node":"VR3",
  "dest-tp":"3-5-1"
},
"supporting-link": [
  {
    "network-ref":"example-native-topology",
    "link-ref":"R3,3-2-1,R2,2-3-1"
  }
],
"ietf-network-slice:network-slice": {
  "isolation-level":
  "ietf-network-slice:physical-network-isolation"
},

}
"link-id":"VR5,5-0-1,",
"source": {
    "source-node":"VR5",
    "source-tp":"5-0-1"
},
"supporting-link": [
    {
        "network-ref":"example-native-topology",
        "link-ref":"R3,3-0-1,"
    }
],
"ietf-network-slice:network-slice": {
    "isolation-level":
    "ietf-network-slice:physical-network-isolation"
},
"link-id":",,VR5,5-0-1",
"destination": {
    "dest-node":"VR5",
    "dest-tp":"5-0-1"
},
"supporting-link": [
    {
        "network-ref":"example-native-topology",
        "link-ref":",,R3,3-0-1"
    }
],
"ietf-network-slice:network-slice": {
    "isolation-level":
    "ietf-network-slice:physical-network-isolation"
},
"ietf-network-slice:network-slice": {
    "optimization-criterion":
    "ietf-te-types:of-minimize-cost-path",
    "isolation-level":
    "ietf-network-slice:physical-isolation"
}
Authors’ Addresses

Xufeng Liu
IBM Corporation
EMail: xufeng.liu.ietf@gmail.com

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

Igor Bryskin
Individual
EMail: i_bryskin@yahoo.com

Luis Miguel Contreras Murillo
Telefonica
EMail: luismiguel.contrerasmurillo@telefonica.com

Qin Wu
Huawei
EMail: bill.wu@huawei.com

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

Reza Rokui
Ciena
Canada
EMail: rrokuiciena.com
IETF Network Slice Deployment Status and Considerations
draft-ma-teas-ietf-network-slice-deployment-00

Abstract

Network Slicing is considered as an important approach to provide different services and customers with the required network connectivity, network resources and performance characteristics over a shared network. Operators have started the deployment of network slices in their networks for different purposes. This document introduces several deployment cases of IETF network slices in operator networks. Some considerations collected from these IETF network slice deployments are also provided.

Requirements Language

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

Status of This Memo

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

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

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

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

Network Slicing is considered as an important mechanism to provide different services and customers with the required network connectivity, resources and performance characteristics over a shared network. [I-D.ietf-teas-ietf-network-slices] describes network slicing in the context of networks built from IETF technologies, and discusses the general framework of IETF network slices. [I-D.ietf-teas-enhanced-vpn] describes the framework and candidate
component technologies for providing enhanced VPN services, by utilizing an approach that is based on existing VPN and Traffic Engineering (TE) technologies and adds characteristics that specific services or customers require above traditional overlay VPNs. VPN+ is delivered using a VPN overlay and an underlying Virtual Transport Network (VTN) which has a set of dedicated or shared resources and is associated with a customized logical network topology in the underlay network. A centralized network controller can be used for the creation and operation of the VTNs, and the mapping of the enhanced VPN services to the appropriate VTNs. The enhanced VPN (VPN+) mechanism can be used for the realization of IETF network slices.

Although the concept of network slicing is firstly introduced for the 5G, the use cases of IETF network slices are not limited to 5G. Operators have started the deployment of IETF network slices based on VPN+ in their networks for different service scenarios. This document introduces several deployment cases of IETF network slices in operator networks. Some considerations about the IETF network slice deployments are also collected.

2. IETF Network Slice Deployment Status

2.1. China Telecom Ningxia

   Service scenario: Multiple industrial services

   Resource partitioning: Virtual sub-interface with dedicated bandwidth

   Data Plane: SRv6

   Control plane: SR Policy with link affinity

2.2. China Mobile Hong Kong

   Service scenario: Fixed-Mobile convergence services

   Resource partitioning: Flexible Ethernet interface and virtual sub-interface with dedicated bandwidth

   Data plane: SR-MPLS

   Control Plane: SR Policy with link affinity

3. IETF Network Slice Deployment Cases
3.1. Network Slicing for Multi-Industrial Network

China Telecom NingXia has deployed a dedicated SRv6 based network to carry multiple industrial services. The three major types of service in the network are: Healthcare service, Education service and Broadband services, and the operator plans to migrate a set of industrial and governmental services from dedicated private networks or Multi-Service Transport Platform (MSTP) networks to this IP based multi-industrial network. With the help of network slicing, services of different industries can be isolated from each other, so that the performance of each service can be guaranteed, and the cost of maintaining and expanding the dedicated private networks for each industry can be reduced.

In order to provide the required resource and security isolation between the health care, education and broadband services, three virtual transport networks (VTNs) are created in the network. All the VTNs share the same IGP instance, while each VTN is defined with a logical topology using different link administrative groups (i.e. color), and is allocated with a set of dedicated bandwidth resources on each involved physical link using the virtual sub-interface mechanism. In a VTN, each link is assigned with a SRv6 End.X SID to identify the sub-interface used for packet forwarding. With more industrial and governmental customers migrate to this network, more VTNs with dedicated network resources will be created.

Multiple L3VPNs belonging to the same industry are provisioned in the corresponding VTN. For example, the VTN created for the health care services is used to support the VPNs for the connection between hospitals belonging to the medical consortium, and the VPNs for connecting the hospitals and the insurance systems in the healthcare cloud. The VPN traffic mapped to a VTN is steered into the set of virtual sub-interfaces of the VTN based on the corresponding SRv6 End.X SIDs.

A centralized network controller is responsible for the management of the VTN and the VPNs. This includes the topology and resource planning of VTN, the VTN creation, the mapping of VPN services to the VTN, and the computation of SRv6 TE paths based on the service constraints and the topology and resource attributes of the VTN. The controller also collects the traffic statistics and performance information of the VTNs and the VPN services to enable the network slice services visualization and ensure the service SLAs are always met.
3.2. Network Slicing for Fixed-Mobile Convergence

China Mobile Hong Kong (CMHK) has deployed network slices in their SR-MPLS based Fixed-Mobile Convergence (FMC) network, which is used to carry the mobile services, the enterprise private line services and the residential broadband services together. Each type of service has different traffic characteristics and performance requirements, thus independent network planning and operation for each service type is required.

Currently three VTNs are created for mobile service, enterprise service and the residential service respectively. Depends on the new service requirement of 5G, More VTNs may be created for 5G critical services in the future. According to the operator’s network planning, each VTN is allocated with a set of dedicated bandwidth resources using either virtual sub-interface or Flexible Ethernet (FlexE) interface mechanism. All the VTNs share the same IGP instance, while the links belonging to different VTNs are assigned with different link administrative groups (i.e. color). In a VTN,
each link is assigned with an SR-MPLS Adj-SID to identify the sub-interface or FlexE interface used for packet forwarding.

Multiple VPNs (EVPN, L3VPN and L2VPN) belonging to the one of the three major service types are mapped to the corresponding VTN. For example, the VTN created for the enterprise private line services is used to support the VPNs of a group of enterprise customers. The VPN traffic mapped to a VTN is steered into the set of virtual sub-interfaces or FlexE interfaces allocated to the VTN based on the corresponding SR-MPLS Adj-SIDs.

A centralized network controller is responsible for the management of the VTN and the VPNs. This includes the topology and resource planning of VTN, the VTN creation, the mapping of VPN services to the VTN, and the computation of SRv6 TE paths based on the service constraints together with the topology and resource attributes of the VTN. The controller also collects the traffic statistics and performance information of the VTNs and the VPN services to enable the network slice services visualization and ensure the service SLAs are always met.

![Diagram of IETF network slice deployment in CMHK](image-url)

Figure 2. IETF network slice deployment in CMHK
4. Network Slice Deployment Considerations

Based on the network slice deployment cases collected in section 2, this section describes some of the operators’ considerations about network slice deployment.

4.1. Isolation

Network slicing is introduced to operators’ network to meet the connectivity and performance requirements of different services or customers. Since many services or customers are migrated from their own dedicated networks to network slices, it is expected that services or customers carried by a network slice will not be affected by any other traffic in the network, thus the resource, policy and security isolation from other services becomes a typical requirement.

Operators have considered the usage of several forwarding plane mechanisms, such as FlexE interface or virtual sub-interfaces to allocate different set of network resources for the VTNs used for different services or customers. The services or customers which do not have specific requirement on resource or security isolation may be provisioned as separated VPNs, while these VPNs can be aggregated and mapped to a shared VTN with a set of aggregated network resources.

4.2. Topology and Connection Types

According to the deployment scenarios of network slices, there can be different requirements on the topology and connection type of the network slices. When a network slice is provided for a particular service type or for a particular industry, the network slice usually covers a network scope similar to the scope of the physical network, and there are usually a large number of end points attached to the network slice, which requires meshed multipoint-to-multipoint connectivity between them. When a network slice is provided for a specific private line service customer, the network slice could have a customized topology covering a portion of the physical network, and usually has a small number of end points attached, in this case the network slice may be expressed as a set of point-to-point connections.

The suitable mechanisms to define the topology of the VTN and build the connectivity needed by network slice service streams. For example, the administrative groups (i.e. color) can be used by a centralized controller to specify the topology of a VTN and compute the constraint paths for network slice services in the VTN. The Distributed control plane based mechanism for topology definition and
the constraint path computation may be used for network slices which require meshed connectivity between a large number of end points.

4.3. Scalability

As shown in several IETF network slice deployments, the number of VTNs at the initial stage can be small (e.g. less than 10). While there are also cases in which hundreds of network slices are needed for industrial and premium private line customers. It is expected that the number of VTNs required in the future could be at the hundreds or even thousands level. Thus the scalability considerations and optimization mechanisms as described in [I-D.dong-teas-enhanced-vpn-vtn-scalability] need to be considered to allow the deployment of a larger number of network slices in the network in future.

4.3.1. Data Plane Scalability

The current deployment of network slices are mainly based on SR-MPLS or SRv6 data plane, with which each VTN is allocated with a separate group of SR SIDs, and the SIDs are associated with a group of dedicated network resources [I-D.ietf-spring-resource-aware-segments]. This provides a practical approach to deliver IETF network slices to meet the requirements in the early stage. While with the number of the required VTNs increases, the increasing amount of SR SIDs will bring challenges both to the forwarding tables and to the network management and operation. It is expected that the mechanisms with dedicated VTN-ID encapsulation as defined in [I-D.dong-6man-enhanced-vpn-vtn-id] could help to reduce the number of SR SIDs needed, and simplify the large scale network slice provisioning and management.

4.4. Automation

The centralized network controller plays an important role in the life cycle management of network slices. With the number of network slices increases, it is necessary that the planning, creation, monitoring and the optimization of IETF network slices can be automated to reduce the burden in the network slice management and operation.

For example, in a network where multiple IETF network slices are deployed, when the bandwidth utilization of one VTN reaches a specific threshold, there are two possible approaches for the VTN capacity expansion. The first approach is to expand the capacity of the physical network, which usually can take a long time. The second approach is to adjust the resource allocation of different VTNs based on the utilization ratio. The network controller can provide the
monitoring and visualization of the resource utilization of the VTNs and VPNs, and gives recommendations about the optimal resource adjustment strategy to the network operator.

5. IANA Considerations

This document makes no request of IANA.

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

6. Security Considerations

TBD

7. Contributors

Terence Ho
Email: terenceho@hk.chinamobile.com

Jimmy Tu
Email: jimmytu@hk.chinamobile.com

Jonathan Chung
Email: jonathanchung@hk.chinamobile.com

Kristy Li
Email: kristyli@hk.chinamobile.com

Tommy Zou:
Email: tommyzou@hk.chinamobile.com

Zhenbin Li
Email: lizhenbin@huawei.com

Zhibo Hu
Email: huobin@huawei.com

8. Acknowledgements

The authors would like to thank XXX for his valuable comments.

9. References
9.1. Normative References

[I-D.ietf-teas-enhanced-vpn]


9.2. Informative References

[I-D.dong-6man-enhanced-vpn-vtn-id]

[I-D.dong-teas-enhanced-vpn-vtn-scalability]

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

[I-D.ietf-spring-sr-for-enhanced-vpn]

[I-D.ietf-teas-ietf-network-slices]
Authors’ Addresses

Yusong Ma
China Telecom Ningxia
Email: mayusong.nx@chinatelecom.cn

Rui Luo
China Telecom Ningxia
Email: luorui.nx@chinatelecom.cn

Alex Chan
China Mobile Hong Kong
Email: alexckchan@hk.chinamobile.com

Ben Suen
China Mobile Hong Kong
Email: bensuen@hk.chinamobile.com

Jie Dong
Huawei Technologies
Email: jie.dong@huawei.com
IETF Network Slice Deployment Status and Considerations
draft-ma-teas-ietf-network-slice-deployment-01

Abstract

Network Slicing is considered as an important approach to provide different services and customers with the required network connectivity, network resources and performance characteristics over a shared network. Operators have started the deployment of network slices in their networks for different purposes. This document introduces several deployment cases of IETF network slices in operator networks. Some considerations collected from these IETF network slice deployments are also provided.

Requirements Language

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

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

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

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

Table of Contents

1. Introduction .................................................. 3
2. IETF Network Slice Deployment Status ........................ 3
   2.1. China Telecom Ningxia .................................... 3
   2.2. China Mobile Hong Kong ................................. 4
   2.3. China Unicom Hebei ..................................... 4
   2.4. Algeria Telecom ........................................ 4
3. IETF Network Slice Deployment Cases .......................... 4
   3.1. Network Slicing for Multi-Industrial Network .......... 4
   3.2. Network Slicing for Fixed-Mobile Convergence ........... 6
   3.3. Network Slicing for Government Affairs Separation ..... 8
   3.4. Network Slicing for Live Video Service ................ 9
4. Network Slice Deployment Considerations ..................... 11
   4.1. Isolation ............................................... 11
   4.2. Topology and Connection Types .......................... 12
   4.3. Scalability ............................................. 12
   4.3.1. Data Plane Scalability ................................ 12
   4.4. Automation ............................................... 13
   4.5. Hierarchical Network Slicing ............................ 13
5. IANA Considerations ............................................ 13
6. Security Considerations ....................................... 13
7. Contributors .................................................. 13
8. Acknowledgements .............................................. 14
9. References .................................................... 14
   9.1. Normative References .................................... 14
   9.2. Informative References ................................... 15
1. Introduction

Network Slicing is considered as an important mechanism to provide different services and customers with the required network connectivity, resources and performance characteristics over a shared network. [I-D.ietf-teas-ietf-network-slices] describes network slicing in the context of networks built from IETF technologies, and discusses the general framework of IETF network slices. [I-D.ietf-teas-ietf-network-slices] also introduces the concept Network Resource Partition (NRP) as a set of network resources that are available to carry traffic and meet the SLOs and SLEs.

[I-D.ietf-teas-enhanced-vpn] describes the framework and candidate component technologies for providing enhanced VPN services, by utilizing an approach that is based on existing VPN and Traffic Engineering (TE) technologies and adds characteristics that specific services or customers require above traditional overlay VPNs. VPN+ is delivered using a VPN overlay and an underlying Virtual Transport Network (VTN) which has a set of dedicated or shared resources and is associated with a customized logical network topology in the underlay network. A centralized network controller can be used for the creation and operation of the VTNs, and the mapping of the enhanced VPN services to the appropriate VTNs. The enhanced VPN (VPN+) mechanism can be used for the realization of IETF network slices. VTN and NRP are considered as similar concepts, and NRP can be seen as an instantiation of VTN in the context of network slicing.

Although the concept of network slicing is firstly introduced for the 5G, the use cases of IETF network slices are not limited to 5G. Operators have started the deployment of IETF network slices based on VPN+ in their networks for different service scenarios. This document introduces several deployment cases of IETF network slices in operator networks. Some considerations about the IETF network slice deployments are also collected.

2. IETF Network Slice Deployment Status

2.1. China Telecom Ningxia

Service scenario: Multiple industrial services

Resource partitioning: Virtual sub-interface with dedicated bandwidth

Data Plane: SRv6
Control plane: SR Policy with link affinity

2.2. China Mobile Hong Kong

Service scenario: Fixed-Mobile convergence services

Resource partitioning: Flexible Ethernet interface and virtual sub-interface with dedicated bandwidth

Data plane: SR-MPLS

Control Plane: SR Policy with link affinity

2.3. China Unicom Hebei

Service scenario: Multiple type of services

Resource partitioning: Flexible Ethernet interface

Data Plane: SRv6

Control Plane: SR Policy with link affinity

2.4. Algeria Telecom

Service scenario: Live Video and other services

Data Plane: SRv6 with NRP-ID

Control plane: SR Policy with NRP-ID

3. IETF Network Slice Deployment Cases

3.1. Network Slicing for Multi-Industrial Network

China Telecom has deployed a dedicated SRv6 based network in Ningxia to carry multiple industrial services. The three major types of service in the network are: Healthcare service, Education service and Broadband services, and the operator plans to migrate a set of industrial and governmental services from dedicated private networks or Multi-Service Transport Platform (MSTP) networks to this IP based multi-industrial network. With the help of network slicing, services of different industries can be isolated from each other, so that the performance of each service can be guaranteed, and the cost of maintaining and expanding the dedicated private networks for each industry can be reduced.
In order to provide the required resource and security isolation between the health care, education and broadband services, three NRPs are created in the network. All the NRPs share the same IGP instance, while each NRP is defined with a logical topology using different link administrative groups (i.e. color), and is allocated with a set of dedicated bandwidth resources on each involved physical link using the virtual sub-interface mechanism. In an NRP, each link is assigned with a SRv6 End.X SID to identify the sub-interface used for packet forwarding. With more industrial and governmental customers migrate to this network, more NRPs with dedicated network resources will be created.

Multiple L3VPNs belonging to the same industry are provisioned in the corresponding NRP. For example, the NRP created for the health care services is used to support the VPNs for the connection between hospitals belonging to the medical consortium, and the VPNs for connecting the hospitals and the insurance systems in the healthcare cloud. The VPN traffic mapped to a NRP is steered into the set of virtual sub-interfaces of the NRP based on the corresponding SRv6 End.X SIDs.

A centralized network controller is responsible for the management of the NRP and the VPNs. This includes the topology and resource planning of NRP, the NRP creation, the mapping of VPN services to the NRP, and the computation of SRv6 TE paths based on the service constraints and the topology and resource attributes of the NRP. The controller also collects the traffic statistics and performance information of the NRPs and the VPN services to enable the network slice services visualization and ensure the service SLAs are always met.
3.2. Network Slicing for Fixed-Mobile Convergence

China Mobile Hong Kong (CMHK) has deployed network slices in their SR-MPLS based Fixed-Mobile Convergence (FMC) network, which is used to carry the mobile services, the enterprise private line services and the residential broadband services together. Each type of service has different traffic characteristics and performance requirements, thus independent network planning and operation for each service type is required.
Currently three NRPs are created for mobile service, enterprise service and the residential service respectively. Depends on the new service requirement of 5G, More NRPs may be created for 5G critical services in the future. According to the operator’s network planning, each NRP is allocated with a set of dedicated bandwidth resources using either virtual sub-interface or Flexible Ethernet (FlexE) interface mechanism. All the NRPs share the same IGP instance, while the links belonging to different NRPs are assigned with different link administrative groups (i.e. color). In a NRP, each link is assigned with an SR-MPLS Adj-SID to identify the sub-interface or FlexE interface used for packet forwarding.

Multiple VPNs (EVPN, L3VPN and L2VPN) belonging to the one of the three major service types are mapped to the corresponding NRP. For example, the NRP created for the enterprise private line services is used to support the VPNs of a group of enterprise customers. The VPN traffic mapped to a NRP is steered into the set of virtual sub-interfaces or FlexE interfaces allocated to the NRP based on the corresponding SR-MPLS Adj-SIDs.

A centralized network controller is responsible for the management of the NRP and the VPNs. This includes the topology and resource planning of NRP, the NRP creation, the mapping of VPN services to the NRP, and the computation of SRv6 TE paths based on the service constraints together with the topology and resource attributes of the NRP. The controller also collects the traffic statistics and performance information of the NRPs and the VPN services to enable the network slice services visualization and ensure the service SLAs are always met.
3.3. Network Slicing for Government Affairs Separation

China Unicom has deployed an SRv6 based cloud network in Hebei for the transport of multiple types of services, including 5G mobile services, government affair service, business private line services and broadband service.

In order to meet the performance and isolation requirement of different type of services, four NRPs are provisioned in the network. All the NRPs share the same IGP instance, while each NRP is defined with a logical topology using different link administrative groups (i.e. color), and is allocated with a set of dedicated bandwidth resources on each involved physical link using FlexE. In an NRP, each FlexE client interface is assigned with a SRv6 End.X SID to steer packets to the set of link resources allocated to the NRP.
According to the service requirement, one or multiple EVPN instances are provisioned for each type of service, and are mapped to the corresponding NRP. For example, an NRP created for the government affair service is used to support the VPNs for the connection between government institution in different cities and towns, and the VPNs for connecting the government institution with the government affair cloud. Based on SRv6 Policy, VPN traffic is steered into a SRv6 TE path which comprises of the FlexE client interfaces of the NRP according to the corresponding SRv6 SID list.

```
+-------------------+       Centralized
 | Network Controller|   Control & Management
+-------------------+
      \|/
       \|

VPN-1 o----/ o----o------o----------o----o NRP-1
VPN-2 o----/ / / / NRP-2 Government affairs
VPN-3 o----/ o----o------o----------o----o 5G mobile

VPN-4 o----/ o----o------o----------o----o NRP-3 Business
VPN-5 o----/ o----o------o----------o----o private lines

VPN-6 o----/ o----o------o----------o----o NRP-0 Broadband

VPN-7 o----/ / / / NRP-1
VPN-8 o----/ o----o------o----------o----o

....

VPN-m o----/ ... /----o NRP-0
                /--------------------/
                Broadband
```

Figure 3. IETF network slice deployment in China Unicom Hebei

3.4. Network Slicing for Live Video Service

Algeria Telecom has deployed an SRv6 based metro network. The recent requirement is to deliver live video broadcast service for sports games, and the related intranet services and internet services together happening on the same sites, the SLA requirement of each type of service is different. There are also existing services which need to coexist with these three types of services.
In order to meet the performance and isolation requirement of these type of services, four NRPs are provisioned in the network:

* NRP-1 for live video services
* NRP-2 for intranet services
* NRP-3 for internet services
* NRP-0 for other services

Figure 4. IETF network slice deployment in Algeria Telecom

All these NRPs share the same IGP instance, while each NRP is allocated with a subset of dedicated network resources. On each physical link which participates in an NRP, a set of link bandwidth is allocated using FlexE, and the FlexE client interface is associated with the NRP-ID.
According to the service requirement, one or multiple L2 or L3 EVPN instances are provisioned for each type of service, and are mapped to the corresponding NRP. For example, an NRP created for the live video broadcast service is used to support the EVPN for the connection between the stadiums and the video broadcasting center.

A network controller performs the path computation using the topology and resources of the NRP as constraints, and SRv6 Policy is used to provision the SRv6 TE paths associated with each NRP to the ingress nodes, using the mechanism defined in [I-D.dong-idr-sr-policy-nrp]. SRv6 Policy is also used to steer the VPN service traffic to SRv6 paths which could meet the service requirement. For VPN traffic which is steered into an SRv6 Policy in an NRP, in addition to encapsulating the SRv6 SID list, the packet is also encapsulated with the global unique NRP-ID in the IPv6 HBH extension header based on the mechanism defined in [I-D.ietf-6man-enhanced-vpn-vtn-id], and the NRP-ID is used to determine the FlexE client interfaces which are used to forward the traffic mapped to the NRP.

4. Network Slice Deployment Considerations

Based on the network slice deployment cases collected in section 2, this section describes some of the operators’ considerations about network slice deployment.

4.1. Isolation

Network slicing is introduced to operators’ network to meet the connectivity and performance requirements of different services or customers. Since many services or customers are migrated from their own dedicated networks to network slices, it is expected that services or customers carried by a network slice will not be affected by any other traffic in the network, thus the resource, policy and security isolation from other services becomes a typical requirement.

Operators have considered the usage of several forwarding plane mechanisms, such as FlexE interface or virtual sub-interfaces to allocate different set of network resources for the NRPs used for different services or customers. The services or customers which do not have specific requirement on resource or security isolation may be provisioned as separated VPNs, while these VPNs can be aggregated and mapped to a shared NRP with a set of aggregated network resources.
4.2. Topology and Connection Types

According to the deployment scenarios of network slices, there can be different requirements on the topology and connection type of the network slices. When a network slice is provided for a particular service type or for a particular industry, the network slice usually covers a network scope similar to the scope of the physical network, and there are usually a large number of end points attached to the network slice, which requires meshed multipoint-to-multipoint connectivity between them. When a network slice is provided for a specific private line service customer, the network slice could have a customized topology covering a portion of the physical network, and usually has a small number of end points attached, in this case the network slice may be expressed as a set of point-to-point connections.

The suitable mechanisms to define the topology of the NRP and build the connectivity needed by network slice service streams. For example, the administrative groups (i.e. color) can be used by a centralized controller to specify the topology of a NRP and compute the constraint paths for network slice services in the NRP. The Distributed control plane based mechanism for topology definition and the constraint path computation may be used for network slices which require meshed connectivity between a large number of end points.

4.3. Scalability

As shown in several IETF network slice deployments, the number of NRPs at the initial stage can be small (e.g. less than 10). While there are also cases in which hundreds of network slices are needed for industrial and premium private line customers. It is expected that the number of NRPs required in the future could be at the hundreds or even thousands level. Thus the scalability considerations and optimization mechanisms as described in [I-D.dong-teas-nrp-scalability] need to be considered to allow the deployment of a larger number of network slices in the network in future.

4.3.1. Data Plane Scalability

The current deployment of network slices are mainly based on SR-MPLS or SRv6 data plane, with which each NRP is allocated with a separate group of SR SIDs, and the SIDs are associated with a group of dedicated network resources [I-D.ietf-spring-resource-aware-segments]. This provides a practical approach to deliver IETF network slices to meet the requirements in the early stage. While with the number of the required NRPs increases, the increasing amount of SR SIDs will bring challenges
both to the forwarding tables and to the network management and operation. It is expected that the mechanisms with dedicated NRP-ID encapsulation as defined in [I-D.ietf-6man-enhanced-vpn-vtn-id] could help to reduce the number of SR SIDs needed, and simplify the large scale network slice provisioning and management.

4.4. Automation

The centralized network controller plays an important role in the life cycle management of network slices. With the number of network slices increases, it is necessary that the planning, creation, monitoring and the optimization of IETF network slices can be automated to reduce the burden in the network slice management and operation.

For example, in a network where multiple IETF network slices are deployed, when the bandwidth utilization of one NRP reaches a specific threshold, there are two possible approaches for the NRP capacity expansion. The first approach is to expand the capacity of the physical network, which usually can take a long time. The second approach is to adjust the resource allocation of different NRPs based on the utilization ratio. The network controller can provide the monitoring and visualization of the resource utilization of the NRPs and VPNs, and gives recommendations about the optimal resource adjustment strategy to the network operation.

4.5. Hierarchical Network Slicing

In the beginning of the network slice deployment, a group of network slice services are provisioned in the same NRP for a particular industry or service type, such as an NRP for all the business private line services. While some of customers within an industry or service type may require to have a set of dedicated network resources allocated within the industry or service type based NRP. This brings the requirement of hierarchical network slicing to the operators. Thus it is expected that the deployed network slices can evolve to support hierarchical network slices according to the service demand.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

TBD

7. Contributors

8. Acknowledgements

The authors would like to thank XXX for his valuable comments.

9. References

9.1. Normative References

[I-D.ietf-teas-enhanced-vpn]

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

9.2. Informative References

[I-D.dong-idr-sr-policy-nrp]

[I-D.dong-teas-nrp-scalability]

[I-D.ietf-6man-enhanced-vpn-vtn-id]

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

[I-D.ietf-spring-sr-for-enhanced-vpn]
Authors’ Addresses

Yusong Ma
China Telecom
Email: mayusong.nx@chinatelecom.cn

Rui Luo
China Telecom
Email: luorui.nx@chinatelecom.cn

Alex Chan
China Mobile Hong Kong
Email: alexckchan@hk.chinamobile.com

Ben Suen
China Mobile Hong Kong
Email: bensuen@hk.chinamobile.com

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

Yang Liu
China Unicom
Email: liuyang118@chinaunicom.cn

Houcine Allahoum
Algeria Telecom
Email: allahoum@algerietelecom.dz
A YANG Data Model for Network Resource Partition (NRP)
draft-wd-teas-nrp-yang-00

Abstract

This document defines a YANG data model for managing Network Resource Partition (NRP) topologies and associated resource allocation. The model can be used for the realization of IETF network slice services.

Status of This Memo

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

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

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

This Internet-Draft will expire on 3 August 2022.

Copyright Notice

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

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

[I-D.ietf-teas-ietf-network-slices] defines IETF network slice services that provide connectivity coupled with network resources commitment between a number of endpoints over a shared network infrastructure and, for scalability concerns, defines network resource partition (NRP) to host one or a group of network slice services according to characteristics including SLOs and SLEs.

[I-D.dong-teas-nrp-scalability] analyzes the scalability issues of network slice services in detail and suggests candidate technologies of control and forwarding planes of the NRP.

This document defines a YANG model of NRP that the IETF NSC (Network Slice controller) can use to manage NRP instances to realize the network slicing services. According to the YANG model classification of [RFC8309], the NRP model is a network configuration model.

2. Terminology

The following terms are defined in [RFC6241] and are used in this specification:

* configuration data

* state data

The following terms are defined in [RFC7950] and are used in this specification:
* augment
* data model
* data node

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

2.1. Tree Diagrams

The tree diagram used in this document follows the notation defined in [RFC8340].

3. NRP Modelling Consideration

As specified in [I-D.ietf-teas-ietf-network-slices], an NRP is a subset of dedicated or shared nodes and links in a network, and includes associated control plane and forwarding plane technologies so that the traffic received from NRP edge nodes that is characterized to match the NRP traffic classification rule is constrained to the NRP exclusive topology and resource allocation. The NRP allows network operators to manage the resources of IETF network slices which are used to provide network slice service traffic with specific SLOs and SLEs.

An NRP is a subset of resources allocated from a physical network or logical network. Depending on the SLO and SLE requirements of the slicing service and also the available resources of the operator’s network, there are several options of creating an NRP. One option is that each physical link is allocated to only one specific NRP, and different NRPs do not share any physical link. One more typical option is that multiple NRPs share the same physical links, and each NRP is built with virtual links with a certain subset of the bandwidth available on the physical links to provide network resource isolation.

To constrain the traffic that matches NRP traffic classification to be forwarded based on the NRP topology and resources, an NRP also includes the control and forwarding plane functions. As defined in [I-D.dong-teas-nrp-scalability], the draft discusses NRP control plane and data plane requirements in different provisioning scenarios, and describes that the NRP control plane is used to exchange network resource attributes and associated logical topology information between nodes of the NRP so that NRP-specific routing and forwarding tables could be generated. For the NRP control plane, distributed control plane mechanism, such as Multi-topology, Flex-Algo or centralized SDN or hybrid combination could be defined. To
help with forwarding entries, several data-plane encapsulation options are also discussed to carry NRP information in the NRP traffic packets. The example NRP data plane identifier could be the IPv6 addresses or the MPLS forwarding labels or dedicated NRP data-plane identifiers.

An example of NRP instances and a physical network is illustrated in Figure 1. In the example, each NRP instance has a customized network topology comprised of a set of links and nodes in the physical network. In control plane, each NRP could be associated with a multi-topology or a Flex-Algo. And it also has its own forwarding plane resources and identifiers which provide NRP-specific packet forwarding.

```
++++   ++++   ++++
+-+=-----+++++++--
+-+-==----+++++++-
++++   +++\ +++++
||     || \  ||
++++   ++++   ++++  \+++   ++++
+---+-++++++-+++++--+
+---+-++++++-+++++--+
++++   ++++   ++++   ++++   ++++
PE1                         PE2

|   |
/   |

o----o-----o
/          /
NRP-1
o-----o-----o----o----o
o----o
/   /\
NRP-2
o-----o----o---o------o
...

o----o
/  /
NPR-n
o-----o----o----o-----o

o  is a virtual node
--- is a virtual link
```

Figure 1: An NRP Example
[I-D.ietf-teas-ietf-network-slices] also describes the management of the NRP. After an NRP created, the NRP may need to be refined and modified as the network status and slice services change, and could be extended if necessary to meet the customers’ demands. In addition to configuration management, the NRP should also provide detailed monitoring information about underlying resources to further provide monitoring for the hosted slice services.

3.1. NRP Model Usage example

One major application of network slices is 5G services. Figure 2 shows the use of the NRP model to realize the IETF Network Slice for the 5G use case, based on the reference framework defined in [I-D.ietf-teas-ietf-network-slices]. The figure shows that the NSC uses the L3VPN network model [I-D.ietf-opsawg-l3sm-l3nm] to map to an IETF Network Slice service and uses the NRP model to map VPN traffic to underlying network resources, so that the SLO and SLE required by the IETF network slice service are ensured when the VPN service traverses the underlying network.

```
+------------------------------------------+
|                 Customer                  |
+------------------------------------------+
| A  Network slice service interface       |
| V                                          |
+------------------------------------------+
| IETF Network Slice Controller (NSC)      |
+------------------------------------------+
| A   L3NM model   | NSC SBI   NRP model |
| V   NRP as VPN underlay                    |
+------------------------------------------+
| Network Controller(s)                     |
+------------------------------------------+
| A  Device model                           |
| V                                          |
+------------------------------------------+

Figure 2: Reference Module Use Case

In the process of realizing an IETF network slice service, the NSC can use a static NRP instance or dynamically create one as one or a group of VPNs underlay construct. Compared with existing VPN underlying built with full mesh tunneling mechanisms, the NRP could
provide resource isolation, topology constraints, and simplified configuration. Additionally, specific service flows of a VPN can be further optimized using SR policies defined in [I-D.dong-idr-sr-policy-vtn].

3.2. NRP Modeling Design

An NRP is modeled as network topology defined in [RFC8345] with augmentations. A new network type "nrp" is defined. A network topology data instance containing the nrp network type, indicates an NRP instance.

As discussed in [I-D.dong-teas-nrp-scalability], an NRP could have multiple control plane implementation options. For a better network scalability, an NRP does not require an independent Layer 3 topology, that is, multiple NRPs can share a same Layer 3 topology or TE topology. Thus, an NRP can use a predefined basic TE topology by referring to the TE network instance or a predefined basic Layer3 TE topology by referring to the network instance with both TE and Layer3 type enabled or other topology combination. The Figure 3 shows the example references between this module and other YANG modules.
But in some situations, an NRP may need its own Layer 3 topology or Traffic Engineering (TE) topology to support route forwarding or TE forwarding capability. Inheriting the extensibility from [RFC8345], an NRP can have several types of networks simultaneously. The Layer 3 Topologies model defined in [RFC8346] can be used to enable an NRP unicast capable. And the TE Topology model defined in [RFC8795] can be used to make an NRP TE capable. The Figure 4 shows the relationship between this module and other YANG modules.

Figure 3: Topology References
The container "nrp" under 'network' of [RFC8345] defines global parameters for an NRP, which defines the specific control plane and data plane mechanisms of an NRP. And also, the traffic steering policy of the NRP may include a dynamic color based policies or an ACL-based static ones.

Each NRP instance consists of a set of nodes and a set of links. Each node and link have different attributes that represent the allocated resources or the operational status of the NRP. An NRP could support several resource partition methods, which are defined by ‘link-partition-type’ under an NRP link, which can further be supported by FlexE or independent queue techniques.

There are multiple modes of NRP operations to be supported as follows:

- **NRP instantiation:** Depending on the slice services types and also network status, there can be two types of approaches. One method is to create an NRP instance before the network controller processes the IETF network slice service request. Another one is that the network controller may start creating an NRP instance while configuring the IETF network slice service request.
* NRP modification: When the capacity of an existing NRP link is close to capacity, the bandwidth of the link could be increased. And when the NRP link or node resources are insufficient, new NRP links and nodes could be added.

* NRP Deletion: If the NSC determines that no slice service is using an NRP, the NSC can delete the NRP instance.

* NRP Monitoring: The NSC can use the NRP model to track and monitor NRP resource status and usage.

4. Description of NRP YANG Module

The description of the NRP data nodes are as follows:

* "nrp-id": Is an identifier that is used to uniquely identify an NRP instance within the network scope.

* NRP resources reservation: The nodes and links represent the network resource allocated for an NRP instance. 'bandwidth-reservation' specifies the bandwidth allocated to an NRP instance, or is overridden by the configuration of the NRP link. 'link-partition-type' specifies the resource partition types of the physical interfaces associated with an NRP link.

* NRP control plane: When an NRP shares an IGP topology or TE topology with other NRPs, "network-ref" or "te-topology-identifier" is used to refer to the existing IGP network instance or TE topology instance. And an NRP can further use Multi-Topology Routing (MTR) or Flex-algo to refer to the IGP instance to generate its own NRP-specific forwarding tables. Multi-Topology Routing (MTR) is defined in [RFC4915], [RFC5120], and [I-D.ietf-lsr-isis-sr-vtn-mt] or Flex-algo is defined in [I-D.ietf-lsr-flex-algo].

* NRP data plane: Defines the data plane mechanism and the NRP identifier of the network domain managed by the network controller. The data plane mechanism could be based on MPLS or IPv6 forwarding. The container "data plane" is used to specify the NRP data plane encapsulation types and values that are used to identify NRP-specific network resources. The NRP data plane identifier is defined in [I-D.ietf-spring-sr-for-enhanced-vpn] and[I-D.dong-6man-enhanced-vpn-vtn-id].

* NRP steering policy: The leaf-list "color-id" is used for dynamic traffic steering based on SR policy of an NRP and The leaf-list "acl-ref" is used for common traffic steering.
5. NRP Yang Module Tree

module: ietf-nrp
   augment /nw:networks/nw:network/nw:network-types:
      ++-rw nrp!
   augment /nw:networks/nw:network:
      ++-rw nrp
      ++-rw nrp-id?         uint32
      ++-rw nrp-name?       string
      ++-rw bandwidth-reservation
         ++-rw (bandwidth-type)?
            ++-(bandwidth-value)
            |  ++-rw bandwidth-value?       uint64
            ++-(bandwidth-percentage)
            |  ++-rw bandwidth-percent?    rt-types:percentage
      ++-rw control-plane
         ++-rw topology-ref
            ++-rw igp-topology-ref
            ++-rw multi-topology-id?    uint32
            ++-rw flex-algo-id?         uint32
            ++-rw te-topology-identifier
               ++-rw provider-id?    te-global-id
               ++-rw client-id?      te-global-id
               ++-rw topology-id?    te-topology-id
      ++-rw data-plane
         ++-rw global-resource-identifier
            ++-rw nrp-dataplane-ipv6-type
               |  ++-rw nrp-dp-value? inet:ipv6-address
            ++-rw nrp-dataplane-mpls-type
               |  ++-rw nrp-dp-value? uint32
            ++-rw nrp-aware-dp
            ++-rw nrp-aware-srv6-type!
            ++-rw nrp-aware-sr-mpls-type!
      ++-rw steering-policy
         ++-rw color-id*          uint32
         ++-rw acl-ref*           -> /acl:acls/acl/name
   augment /nw:networks/nw:network/nw:node:
      ++-rw nrp
      |  ++-rw nrp-aware-srv6
         |  ++-rw nrp-dp-value?    srv6-types:srv6-sid
         ++-rw nrp-aware-sr-mpls
         |  ++-rw nrp-dp-value?    rt-types:mpls-label
   augment /nw:networks/nw:network/nt:link:
      ++-rw nrp
      |  ++-rw link-partition-type? identityref
      |  ++-rw bandwidth-reservation
6. NRP Yang Module

<CODE BEGINS> file "ietf-nrp@2022-01-29.yang"

module ietf-nrp {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-nrp";
  prefix nrp;

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-te-types {
    prefix te-types;
    reference
  }

"RFC 8776: Traffic Engineering Common YANG Types";
}
import ietf-te-packet-types {
  prefix te-packet-types;
  reference
    "RFC 8776: Traffic Engineering Common YANG Types";
}
import ietf-srv6-types {
  prefix srv6-types;
}
import ietf-inet-types {
  prefix inet;
  reference
    "RFC 6991: Common YANG Data Types";
}
import ietf-access-control-list {
  prefix acl;
  reference
    "RFC 8519: YANG Data Model for Network Access Control Lists (ACLs)";
}

organization
  "IETF TEAS Working Group";

contact
  "
  WG Web: <http://tools.ietf.org/wg/teas/>
  WG List:<mailto:teas@ietf.org>
  Editor: Bo Wu <lana.wubo@huawei.com>
    : Dhruv Dhody <dhruv.ietf@gmail.com>
  
  "This YANG module defines a network data module for NRP (Network Resource Partition).

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

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

  This version of this YANG module is part of RFC XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.";
revision 2022-01-29 {
  description
    "This is the initial version of NRP YANG model.";
  reference
    "RFC XXX: A YANG Data Model for Network Resource Partition"
}

identity link-partition-type {
  description
    "Base identity for interface partition type."
}

identity virtual-sub-interface-partition {
  base link-partition-type;
  description
    "Identity for virtual interface or sub-interface, e.g. FlexE."
}

identity queue-partition {
  base link-partition-type;
  description
    "Identity for queue partition type."
}

identity nrp-dataplane-type {
  description
    "Base identity for NRP data plane type."
}

identity nrp-dataplane-ipv6 {
  base nrp-dataplane-type;
  description
    "Identity for NRP specific packet forwarding of IPv6."
}

identity nrp-dataplane-mpls {
  base nrp-dataplane-type;
  description
    "Identity for NRP specific packet forwarding of MPLS."
}

identity nrp-dataplane-sr-mpls {
  base nrp-dataplane-type;
  description
    "Identity for NRP specific packet forwarding of SR MPLS."
}

identity nrp-dataplane-srv6 {
base nrp-dataplane-type;
description
   "Identity for NRP specific packet forwarding of SRv6.";
}

/*
 * Groupings
 */

grouping nrp-bandwidth-reservation {
   description
      "Grouping for NRP bandwidth reservation.";
   container bandwidth-reservation {
      description
         "Container for NRP bandwidth reservation.";
      choice bandwidth-type {
         description
            "Choice of bandwidth reservation type.";
         case bandwidth-value {
            leaf bandwidth-value {
               type uint64;
               units "bps";
               description
                  "Bandwidth allocation for the NRP as absolute value.";
            }
         } case bandwidth-percentage {
            leaf bandwidth-percent {
               type rt-types:percentage;
               description
                  "Bandwidth allocation for the NRP as a percentage of a link.";
            }
         }
      }
   }
}

grouping nrp-control-plane-attributes {
   description
      "Grouping for NRP control plane attributes.";
   container control-plane {
      description
         "The container of NRP control plane mechanisms.";
      container topology-ref {
         description
            "Container for topology reference.";
      }
      container igp-topology-ref {
         description
            "Container for IGP topology reference.";
      }
   }
}
description
 "Container for IGP topology reference.";
 uses nw:network-ref;
 leaf multi-topology-id {
   type uint32;
   description
   "The MT-id of an NRP.";
 }
 leaf flex-algo-id {
   type uint32;
   description
   "The flex-algo-id of an NRP.";
 }
 uses te-types:te-topology-identifier;
}

grouping nrp-data-plane-attributes {
 description
 "Grouping for NRP data plane attributes.";
 container data-plane {
 description
 "The data plane mechanisms of an NRP. The forwarding plane
 could be MPLS, IPv6, SRv6, or SR-MPLS.";
 container global-resource-identifier {
 description
 "The container of global NRP data-plane ID.";
 container nrp-dataplane-ipv6-type {
 description
 "The container of IPv6 based NRP data-plane identifier.";
 leaf nrp-dp-value {
   type inet:ipv6-address;
   description
   "Indicates the IPv6 NRP data-plane identifier.";
 }
 }
 container nrp-dataplane-mpls-type {
 description
 "The container of MPLS based NRP data-plane identifier.";
 leaf nrp-dp-value {
   type uint32;
   description
   "Indicates MPLS metadata values to identify MPLS NRP
   data plane identifier, e.g. Ancillary data.";
 }
 }
container nrp-aware-dp {
  description
    "The container of SR based NRP data-plane identifier.";
  container nrp-aware-srv6-type {
    presence "Enables SRv6 data plane type.";
    description
      "The container of SRv6 based NRP data-plane identifier.";
  }
  container nrp-aware-sr-mpls-type {
    presence "Enables SR MPLS data plane type.";
    description
      "The container of SR MPLS based NRP data-plane identifier.";
  }
}

grouping nrp-traffic-steering-policy {
  description
    "The grouping of the NRP traffic steering policy.";
  container steering-policy {
    description
      "The container of a policy set matching an NRP traffic classifier.";
    leaf-list color-id {
      type uint32;
      description
        "A list of color ID for NRP traffic steering based on SR policy.";
    }
    leaf-list acl-ref {
      type leafref {
        path "/acl:acls/acl:acl/acl:name";
      }
      description
        "A list of ACL for NRP traffic classification.";
    }
  }
}

grouping nrp-aware-id {
  description
    "The grouping of NRP aware SR ID.";
  container nrp-aware-srv6 {
    description
      "The container of SRv6 based NRP data plane identifier.";
    leaf nrp-dp-value {
      
"The container of SRv6 based NRP data-plane identifier.";
  }
  container nrp-aware-sr-mpls-type {
    presence "Enables SR MPLS data plane type.";
    description
      "The container of SR MPLS based NRP data-plane identifier.";
  }
}

}
type srv6-types:srv6-sid;
description
  "Indicates the SRv6 SID value as the NRP data plane identifier.";
}
}
container nrp-aware-sr-mpls {
description
  "The container of SR MPLS based NRP data plane identifier.";
leaf nrp-dp-value {
type rt-types:mpls-label;
description
  "Indicates the SR MPLS ID value as the NRP data plane identifier.";
}
}
}
}

// nrp
}

// nrp-node-attributes

grouping nrp-node-attributes {
description
  "NRP node scope attributes.";
container nrp {

grouping nrp-link-attributes {
    description "NRP link scope attributes.";
    container nrp {
        description "Containing NRP attributes.";
        leaf link-partition-type {
            type identityref {
                base link-partition-type;
            }
            description "Indicates the resource partition type of a link.";
        }
        uses nrp-bandwidth-reservation;
        uses nrp-aware-id;
    }
}

// nrp-statistics

grouping statistics-per-nrp {
    description "Statistics attributes per NRP.";
}

// nrp-node-statistics

grouping statistics-per-node {
    description "Statistics attributes per NRP node.";
}

// one-way-performance-metrics

grouping one-way-performance-bandwidth {
    description "Grouping for one-way performance bandwidth.";
    leaf one-way-available-bandwidth {
        type rt-types:bandwidth-ieee-float32;
        units "bytes per second";
    }
}

default "0x0p0";

description
"Available bandwidth that is defined to be NRP link bandwidth minus bandwidth utilization. For a bundled link, available bandwidth is defined to be the sum of the component link available bandwidths."
}

leaf one-way-utilized-bandwidth {
  type rt-types:bandwidth-ieee-float32;
  units "bytes per second";
  default "0x0p0";
  description
  "Bandwidth utilization that represents the actual utilization of the link (i.e. as measured in the router). For a bundled link, bandwidth utilization is defined to be the sum of the component link bandwidth utilizations.";
}

// nrp-link-statistics

grouping nrp-statistics-per-link {
  description
  "Statistics attributes per NRP link.";
  container statistics {
    config false;
    description
    "Statistics for NRP link.";
    leaf admin-status {
      type te-types:te-admin-status;
      description
      "The administrative state of the link.";
    }
    leaf oper-status {
      type te-types:te-oper-status;
      description
      "The current operational state of the link.";
    }
    uses one-way-performance-bandwidth;
    uses te-packet-types:one-way-performance-metrics-packet;
  }
}

augment "/nw:networks/nw:network/nw:network-types" {
  description
  "Defines the NRP topology type.";
  container nrp {

presence "Indicates NRP topology";
    description
      "The presence identifies the NRP type.";

}  

augment "/nw:networks/nw:network" {
  when 'nw:network-types/nrp:nrp' {
    description
      "Augment only for NRP topology.";
  }
  description
    "Augment NRP configuration and state.";
  uses nrp-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
  when '../nw:network-types/nrp:nrp' {
    description
      "Augment only for NRP topology.";
  }
  description
    "Augment node configuration and state.";
  uses nrp-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
  when '../nw:network-types/nrp:nrp' {
    description
      "Augment only for NRP topology.";
  }
  description
    "Augment link configuration and state.";
  uses nrp-link-attributes;
  uses nrp-statistics-per-link;
}

7. 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.

nrp-link: A malicious client could attempt to remove a link from a topology, add a new link. In each case, the structure of the topology would be sabotaged, and this scenario could, for example, result in an NRP topology that is less than optimal.

The entries in the nodes above include the whole network configurations corresponding with the NRP, 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.

8. 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:

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-nrp
Prefix: nrp
Reference: RFC XXXX

9. Contributor
10. References

10.1. Normative References

[I-D.dong-6man-enhanced-vpn-vtn-id]

[I-D.dong-idr-sr-policy-vtn]

[I-D.ietf-lsr-flex-algo]

[I-D.ietf-lsr-isis-sr-vtn-mt]

[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
[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,

Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF",
RFC 4915, DOI 10.17487/RFC4915, June 2007,

Topology (MT) Routing in Intermediate System to
Intermediate Systems (IS-ISs)", RFC 5120,
DOI 10.17487/RFC5120, February 2008,

and A. Bierman, Ed., "Network Configuration Protocol
(NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure
Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011,

RFC 7950, DOI 10.17487/RFC7950, August 2016,

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG",
RFC 7951, DOI 10.17487/RFC7951, August 2016,

Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017,

Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018,

BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,


10.2. Informative References


Appendix A.  An Example

This section contains an example of an instance data tree in JSON encoding [RFC7951]. The example instantiates ietf-nrp for the topology that is depicted in the following diagram. There are three nodes, D1, D2, and D3. D1 has three termination points, 1-0-1, 1-2-1, and 1-3-1. D2 has three termination points as well, 2-1-1, 2-0-1, and 2-3-1. D3 has two termination points, 3-1-1 and 3-2-1. In addition there are six links, two between each pair of nodes with one going in each direction.

Figure 5: An NRP Instance Example

The corresponding NRP instance data tree is depicted below:
{  
  "ietf-network:networks":{
    "network":{
      "network-types":{
        "ietf-nrp:nrp":{
          
        },
      },
      "network-id":"nrp-example",
      "ietf-nrp:nrp":{
        "nrp-id":"NRP1",
        "bandwidth-reservation":{
          "bandwidth-value":10000
        },
        "control-plane":{
          "topology-ref":{
            "igp-topology-ref":{
              "network-ref":"L3-topology",
              "flex-algo-id":129
            }
          }
        },
        "data-plane":{
          "global-resource-identifier":{
            "nrp-dataplane-ipv6-type":{
              "nrp-dp-value":100
            }
          }
        },
        "steering-policy":{
          "color-id":100
        }
      }
    },
    "node":{
      "node-id":"D1",
      "termination-point":{
        "tp-id":"1-0-1"
      },
      "tp-id":"1-2-1"
    },
    "tp-id":"1-3-1"
  }
}
{
    "node-id":"D2",
    "termination-point":[
        {
            "tp-id":"2-0-1"
        },
        {
            "tp-id":"2-1-1"
        },
        {
            "tp-id":"2-3-1"
        }
    ],
    "node-id":"D3",
    "termination-point":[
        {
            "tp-id":"3-2-1"
        }
    ]
},
"ietf-network-topology:link":{
    "link-id":"D1-1-2-1,D2-2-1-1",
    "source":{
        "source-node":"D1",
        "source-tp":"1-2-1"
    },
    "destination":{
        "dest-node":"D2",
        "dest-tp":"2-1-1"
    },
    "ietf-nrp:nrp":{
        "link-partition-type": "virtual-sub-interface-partition",
        "bandwidth-reservation":{
            "bandwidth-value":10000
        },
        "nrp-aware-srv6":{
            "nrp-dp-value":101
        }
    }
}
{
    "link-id":"D2,2-1-1,D1,1-2-1",
    "source":{
        "source-node":"D2",
        "source-tp":"2-1-1"
    },
    "destination":{
        "dest-node":"D1",
        "dest-tp":"1-2-1"
    },
    "ietf-nrp:nrp":{
        "link-partition-type": "virtual-sub-interface-partition",
        "bandwidth-reservation":{
            "bandwidth-value":"10000"
        },
        "nrp-aware-srv6":{
            "nrp-dp-value":101
        }
    }
},
{
    "link-id":"D1,1-3-1,D3,3-1-1",
    "source":{
        "source-node":"D1",
        "source-tp":"1-3-1"
    },
    "destination":{
        "dest-node":"D3",
        "dest-tp":"3-1-1"
    },
    "ietf-nrp:nrp":{
        "link-partition-type": "virtual-sub-interface-partition",
        "bandwidth-reservation":{
            "bandwidth-value":"10000"
        },
        "nrp-aware-srv6":{
            "nrp-dp-value":101
        }
    }
},
{
    "link-id":"D3,3-1-1,D1,1-3-1",
    "source":{
        "source-node":"D3",
        "source-tp":"3-1-1"
    },
    "destination":{
        "dest-node":"D1",
        "dest-tp":"1-3-1"
    },
    "ietf-nrp:nrp":{
        "link-partition-type": "virtual-sub-interface-partition",
        "bandwidth-reservation":{
            "bandwidth-value":"10000"
        },
        "nrp-aware-srv6":{
            "nrp-dp-value":101
        }
    }
}
"destination":{
   "dest-node":"D1",
   "dest-tp":"1-3-1"
},
"ietf-nrp:nrp":{
   "link-partition-type":
   "virtual-sub-interface-partition",
   "bandwidth-reservation":{
      "bandwidth-value":"10000"
   },
   "nrp-aware-srv6":{
      "nrp-dp-value":101
   }
},
"link-id":"D2,2-3-1,D3,3-2-1",
"source":{
   "source-node":"D2",
   "source-tp":"2-3-1"
},
"destination":{
   "dest-node":"D3",
   "dest-tp":"3-2-1"
},
"ietf-nrp:nrp":{
   "link-partition-type":
   "virtual-sub-interface-partition",
   "bandwidth-reservation":{
      "bandwidth-value":"10000"
   },
   "nrp-aware-srv6":{
      "nrp-dp-value":101
   }
},
"link-id":"D3,3-2-1,D2,2-3-1",
"source":{
   "source-node":"D3",
   "source-tp":"3-2-1"
},
"destination":{
   "dest-node":"D2",
   "dest-tp":"2-3-1"
},
"ietf-nrp:nrp":{
   "link-partition-type":
   "virtual-sub-interface-partition",
   "bandwidth-reservation":{
      "bandwidth-value":"10000"
   },
   "nrp-aware-srv6":{
      "nrp-dp-value":101
   }
}
"virtual-sub-interface-partition",
"bandwidth-reservation":{
  "bandwidth-value":"10000"
},
"nrp-aware-srv6":{
  "nrp-dp-value":101
}
}
}
}
}

Figure 6: Instance data tree

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 560066
Karnataka
India

Email: dhruv.ietf@gmail.com

Ying Cheng
China Unicom
Beijing
China

Email: chengying10@chinaunicom.cn
A YANG Data Model for Network Resource Partition (NRP)
draft-wd-teas-nrp-yang-01

Abstract

This document defines a YANG data model for managing Network Resource Partition (NRP) topologies and associated resource allocation. The model can be used for the realization of IETF network slice services.

Status of This Memo

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

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

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

This Internet-Draft will expire on 12 January 2023.

Copyright Notice

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

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

[I-D.ietf-teas-ietf-network-slices] defines IETF network slice services that provide connectivity coupled with network resources commitment between a number of endpoints over a shared network infrastructure and, for scalability concerns, defines network resource partition (NRP) to host one or a group of network slice services according to characteristics including SLOs and SLEs. [I-D.dong-teas-nrp-scalability] analyzes the scalability issues of network slice services in detail and suggests candidate technologies of control and forwarding planes of the NRP.

This document defines a YANG model of NRP that the IETF NSC (Network Slice controller) can use to manage NRP instances to realize the network slicing services. According to the YANG model classification of [RFC8309], the NRP model is a network configuration model.

2. Terminology

The following terms are defined in [RFC6241] and are used in this specification:

* configuration data

* state data
The following terms are defined in [RFC7950] and are used in this specification:

- augment
- data model
- data node

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

2.1. Tree Diagrams

The tree diagram used in this document follows the notation defined in [RFC8340].

3. NRP modelling requirements

[I-D.ietf-teas-ietf-network-slices] section 6.1 introduces the concept of NRP, which is a collection of resources (bufferage, queuing, scheduling, etc.) in the underlay network to provide specific SLOs and SLEs for connectivity constructs of IETF Network Slice services. [I-D.ietf-teas-ns-ip-mpls] provides some solutions to realize network slicing in IP/MPLS networks. [I-D.dong-teas-nrp-scalability] provides analysis and possible optimizations of the control plane and data plane of NRP in IP/MPLS networks for better scalability. The following are some common NRP attributes for NRP management identified based on the analysis:

- NRP instantiation
  - NRP partition type: Refers to various NRP resource partition methods, such as control plane partition, data plane partition, no partition, etc.
  - NRP topology generation method: Topologies can be created using multiple methods. For example, NRP links can be all links in the native topology, or explicitly selected links from native topology or implicitly selected from the various existing topologies.
  - NRP resource reservation: Reserves link resources for the NRP.
  - NRP control plane: Mechanisms that provide routing and forwarding to one or a group of network slice traffic to ensure the corresponding SLO and SLE through NRP link resources.
- NRP data plane: Dataplane identifier carried in a data packet, which is used to mark the link resources and behaviors allocated to the NRP.

- NRP steering policy: Policies for steering slice traffic to the NRP.

* NRP modification or updates: Modifications or additions to existing NRP-allocated resources, e.g. some congested links need to be expanded.

* NRP monitoring: NRP-allocated resources, including NRP-specific link or node SID, link bandwidth usage, link delay, and packet loss status, etc.

4. NRP Modelling Consideration

An NRP is a subset, or all, of resources allocated from a physical network or logical network. Depending on the SLO and SLE requirements of the slicing service and also the available resources of the operator’s network, there are several options of creating an NRP. One option is that each physical link is allocated to only one specific NRP, and different NRPs do not share any physical link. One more typical option is that multiple NRPs share the same physical links, and each NRP is built with virtual links with a certain subset of the bandwidth available on the physical links to provide network resource isolation.

In addition to specifying resource allocation from the underlay network, An NRP also needs to have associated control plane and forwarding plane technologies, which can provide specific routing and forwarding so that the traffic received from NRP edge nodes that is characterized to match the NRP traffic classification rule is constrained to the NRP exclusive topology and resource allocation. The NRP allows network operators to manage the resources of IETF network slices which are used to provide network slice service traffic with specific SLOs and SLEs.

As defined in [I-D.dong-teas-nrp-scalability], the draft discusses NRP control plane and data plane requirements in different provisioning scenarios, and describes that the NRP control plane is used to exchange network resource attributes and associated logical topology information between nodes of the NRP so that NRP-specific routing and forwarding tables could be generated. For the NRP control plane, distributed control plane mechanism, such as Multi-topology, Flex-Algo or centralized SDN or hybrid combination could be defined. To help with forwarding entries, several data-plane encapsulation options are also discussed to carry NRP information in
the NRP traffic packets. The example NRP data plane identifier could be the IPv6 addresses or the MPLS forwarding labels or dedicated NRP data-plane identifiers.

An example of NRP instances and a physical network is illustrated in Figure 1. In the example, each NRP instance has a customized network topology comprised of a set of links and nodes in the physical network. In control plane, each NRP could be associated with a multi-topology or a Flex-Algo. And it also has its own forwarding plane resources and identifiers which provide NRP-specific packet forwarding.

```
+----+----+----+----+----+
|    |    |    |    |    |
+----+----+----+----+----+

+----+----+----+----+----+
|    |    |    |    |    |
+----+----+----+----+----+

+----+----+----+----+----+
+----+----+----+----+----+
```

**Physical Network**

```
++++    ++++    ++++    ++++    ++++
```

PE1      PE2

```
 PE1     PE2
 o----o-----o----o----o----o----o----o
      /              /              /              /              /
```

```
NRP-1:
 o-----o------o------o
      /              /
```

```
NRP-2:
 o-----o
      /              /
```

```
NRP-n:
 o-----o
      /              /
```

--- is a virtual link

Figure 1: An NRP Example
4.1. NRP Model Usage example

One major application of network slices is 5G services. Figure 2 shows the use of the NRP model to realize the IETF Network Slice for the 5G use case, based on the reference framework defined in [I-D.ietf-teas-ietf-network-slices]. The figure shows that the NSC uses the L3VPN network model [I-D.ietf-opsawg-l3sm-l3nm] to map to an IETF Network Slice service and uses the NRP model to map VPN traffic to underlying network resources, so that the SLO and SLE required by the IETF network slice service are ensured when the VPN service traverses the underlying network.

```
+------------------------------------------+
|                 Customer                 |
|                                          |
+------------------------------------------+

+------------------------------------------+
|  IETF Network slice service interface    |
|                                          |
+------------------------------------------+

+------------------------------------------+
|    IETF Network Slice Controller (NSC)   |
|                                          |
+------------------------------------------+

+------------------------------------------+
|    L3NM | Network Configuration Interface         |
|        | NRP Model                               |
|        +------------------------------------+
|        | Network Controller(s)                   |
|        +------------------------------------+
|        | Device model                            |
|        +------------------------------------+
|                                          |
+------------------------------------------+

Figure 2: Reference Module Use Case

In the process of realizing an IETF network slice service, the NSC can use a static NRP instance or dynamically create one as one or a group of VPNs underlay construct. Compared with existing VPN underlying built with full mesh tunneling mechanisms, the NRP could
provide resource isolation, topology constraints, and simplified configuration. Additionally, specific service flows of a VPN can be further optimized using SR policies defined in [I-D.dong-idr-sr-policy-vtn].

4.2. NRP Modeling Design

As defined in [I-D.ietf-teas-ietf-network-slices], a network resource partition (NRP) is a collection of resources in the underlay network. An NRP can have a dedicated topology or can use a shared topology with other NRPs.

Therefore, an NRP is modeled as network topology defined in [RFC8345] with augmentations. A new network type "nrp" is defined. A network topology data instance containing the nrp network type, indicates an NRP instance. The Figure 3 shows the relationship between this module and other topology modules.

```
+-----------------------+
| Network Topology Model |
| RFC8345               |
+-----------------------+

| V | V | V | V |
```

Figure 3: NRP Model Relationship

The container "nrp" under 'network' of [RFC8345] defines global parameters for an NRP, which defines NRP partition type, NRP topology generation method, and the specific control plane and data plane mechanisms of an NRP. And also, the traffic steering policy of the NRP may include a dynamic color based policies or an ACL-based static ones.

The NRP partition type is used to describe multiple NRP resource partition methods, for example, no partition, control plane resource partition, data plane resource partition, or a combination of two types.
As an NRP may consist of the entire or a subset of links in the underlay network, there are various methods to generate NRP topology, which include:

The NRP with a subset of links in the underlay network, which has the same topology as the pre-built L3 topology, MT topology, flexalgo, or TE topology, and also has the same resource reservation requirements. The topology definition may come directly from the topology defined by "control plane".

For other NRPs that require a dedicated topology, "nrp-topology-group" is used to configure the selected links from the base topology. Generally, the base topology refers to the underlay network topology. An NRP can be configured with one or more "nrp-topology-group" to create topology resources required by the NRP. For example, if an NRP needs to reserve the same bandwidth for a group of links, the same "group-id" can be assigned to the links and "bandwidth-reservation" is specified, such as access network link group, aggregation network link group, etc. If some inter-domain links, have multiple bandwidth reservation requirements, they can also be classified into a group. Then, each link can override the bandwidth reservation of the group bandwidth reservation.

As discussed in [I-D.dong-teas-nrp-scalability], an NRP could have multiple control plane implementation options. For a better network scalability, an NRP does not require an independent distributed control protocol instance or a independent centralized control plane instance, that is, multiple NRPs can share a same control plane instance. Thus, an NRP can use a predefined native or abstract TE topology by referring to a TE network instance or a predefined control protocol instance by referring to Layer3 network instance.

In addition to global NRP parameters, each NRP instance also consists of a set of nodes and a set of links, which have different attributes that represent the allocated resources or the operational status of the NRP. An NRP could support several data plane resource partition methods, which are defined by 'link-partition-type’ under an NRP link, which can further be supported by FlexE or independent queue techniques.

There are multiple modes of NRP operations to be supported as follows:
* NRP instantiation: Depending on the slice services types and also network status, there can be two types of approaches. One method is to create an NRP instance before the network controller processes the IETF network slice service request. Another one is that the network controller may start creating an NRP instance while configuring the IETF network slice service request.

* NRP modification: When the capacity of an existing NPR link is close to capacity, the bandwidth of the link could be increased. And when the NRP link or node resources are insufficient, new NRP links and nodes could be added.

* NRP Deletion: If the NSC determines that no slice service is using an NRP, the NSC can delete the NRP instance.

* NRP Monitoring: The NSC can use the NRP model to track and monitor NRP resource status and usage.

5. Description of NRP YANG Module

The description of the NRP data nodes are as follows:

* "nrp-id": Is an identifier that is used to uniquely identify an NRP instance within the network scope.

* NRP partition type: Refers to control plane resource partition, data plane resource partition, or a combination of two types.

* NRP resources reservation: The nodes and links represent the network resource allocated for an NRP instance. 'bandwidth-reservation' specifies the bandwidth allocated to an NRP instance, or is overridden by the configuration of the NRP link. 'link-partition-type' specifies the resource partition types of the physical interfaces associated with an NRP link.

* NRP control plane: When an NRP shares an IGP instance or TE instance with other NRPs, "igp-topology-ref" or "te-topology-identifier" is used to refer to the existing IGP network instance or TE instance. And an NRP can further use Multi-Topology Routing (MTR) or Flex-algo to refer to the IGP instance to generate its own NRP-specific forwarding tables. Multi-Topology Routing (MTR) is defined in [RFC4915], [RFC5120], and [I-D.ietf-lsr-isis-sr-vtn-mt] or Flex-algo is defined in [I-D.ietf-lsr-flex-algo].

* NRP data plane: Defines the data plane mechanism and the NRP identifier of the network domain managed by the network controller. The data plane mechanism could be based on MPLS or
IPv6 forwarding. The container "data plane" is used to specify the NRP data plane encapsulation types and values that are used to identify NRP-specific network resources. The NRP data plane identifier is defined in [I-D.ietf-spring-sr-for-enhanced-vpn] and [I-D.dong-6man-enhanced-vpn-vtn-id].

* NRP steering policy: The leaf-list "color-id" is used for dynamic traffic steering based on SR policy of an NRP and The leaf-list "acl-ref" is used for common traffic steering.

* NRP topology group: The list "nrp-topology-group" is used to explicitly select subset of links of a underlay topology.

6. NRP Yang Module Tree

module: ietf-nrp
  augment /nw:networks/nw:network/nw:network-types:
    +--rw nrp!
  augment /nw:networks/nw:network:
    +--rw nrp
      +--rw nrp-id?                   uint32
      +--rw nrp-name?                 string
      +--rw partition-type?           identityref
       
      +--rw bandwidth-reservation
       +--rw (bandwidth-type)?
       |    +--:(bandwidth-value)
       |       +--rw bandwidth-value?    uint64
       |    +--:(bandwidth-percentage)
       |       +--rw bandwidth-percent?  rt-types:percentage
       
    +--rw control-plane
      +--rw topology-ref
       +--rw igp-topology-ref
       |  +--rw network-ref?
       |  |    +-- /nw:networks/network/network-id
       |  +--rw multi-topology-id?      uint32
       |  +--rw flex-algo-id?           uint32
       +--rw te-topology-identifier
       |  +--rw provider-id?            te-global-id
       |  +--rw client-id?              te-global-id
       +--rw topology-id?             te-topology-id

    +--rw data-plane
      +--rw global-resource-identifier
      |  +--rw nrp-dataplane-ipv6-type
      |  |  +--rw nrp-dp-value?         inet:ipv6-address
      |  +--rw nrp-dataplane-mpls-type
      |     +--rw nrp-dp-value?        uint32
      +--rw nrp-aware-dp
       +--rw nrp-aware-srv6-type!
| +--rw nrp-aware-sr-mpls-type!  
|  +--rw steering-policy  
|  |  +--rw color-id*   uint32  
|  |  +--rw acl-ref*   -> /acl:acls/acl/name  
| +--rw nrp-topology-group* [group-id]  
|  +--rw group-id                 string  
|  +--rw base-topology-ref  
| +--rw links* [link-ref]  
|  +--rw link-ref                   leafref  
|  +--rw link-attributes-override  
|  |  +--rw bandwidth-reservation  
|  |  |  +--rw (bandwidth-type)?  
|  |  |  |  +--:(bandwidth-value)  
|  |  |  |  |  +--rw bandwidth-value?     uint64  
|  |  |  |  +--:(bandwidth-percentage)  
|  |  |  |  +--rw bandwidth-percent?   rt-types:percentage  
|  |  +--rw bandwidth-reservation  
|  |  |  +--rw (bandwidth-type)?  
|  |  |  |  +--:(bandwidth-value)  
|  |  |  |  |  +--rw bandwidth-value?     uint64  
|  |  |  |  +--:(bandwidth-percentage)  
|  |  |  |  +--rw bandwidth-percent?   rt-types:percentage  
|  augment /nw:networks/nw:network/nw:node:  
| +--ro nrp  
|  +--ro nrp-aware-dp-id  
|  |  +--ro nrp-dp-srv6?      srv6-types:srv6-sid  
|  |  +--ro nrp-dp-sr-mpls?   rt-types:mpls-label  
| augment /nw:networks/nw:network/nt:link:  
| +--rw nrp  
|  +--rw bandwidth-reservation  
|  |  +--rw (bandwidth-type)?  
|  |  |  +--:(bandwidth-value)  
|  |  |  |  +--rw bandwidth-value?     uint64  
|  |  |  +--:(bandwidth-percentage)  
|  |  |  +--rw bandwidth-percent?   rt-types:percentage  
|  +--rw partition-type?          identityref  
|  +--ro nrp-aware-dp-id  
|  |  +--ro nrp-dp-srv6?      srv6-types:srv6-sid  
|  |  +--ro nrp-dp-sr-mpls?   rt-types:mpls-label  
| +--ro statistics  
|  +--ro admin-status?  
|  |  |  te-types:te-admin-status  
|  +--ro oper-status?  
|  |  |  te-types:te-oper-status  
|  +--ro one-way-available-bandwidth?  
|  |  |  rt-types:bandwidth-ieee-float32
---ro one-way-utilized-bandwidth?
    | rt-types:bandwidth-ieee-float32
---ro one-way-min-delay?           uint32
---ro one-way-max-delay?           uint32
---ro one-way-delay-variation?     uint32
---ro one-way-packet-loss?         decimal64

7. NRP Yang Module

<CODE BEGINS> file "ietf-nrp@2022-07-11.yang"

module ietf-nrp {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-nrp";
    prefix nrp;

    import ietf-network {
        prefix nw;
        reference
        "RFC 8345: A YANG Data Model for Network Topologies";
    }
    import ietf-network-topology {
        prefix nt;
        reference
        "RFC 8345: A YANG Data Model for Network Topologies";
    }
    import ietf-routing-types {
        prefix rt-types;
        reference
        "RFC 8294: Common YANG Data Types for the Routing Area";
    }
    import ietf-te-types {
        prefix te-types;
        reference
        "RFC 8776: Traffic Engineering Common YANG Types";
    }
    import ietf-te-packet-types {
        prefix te-packet-types;
        reference
        "RFC 8776: Traffic Engineering Common YANG Types";
    }
    import ietf-srv6-types {
        prefix srv6-types;
    }
    import ietf-inet-types {
        prefix inet;
        reference
        "RFC 6991: Common YANG Data Types";
    }
import ietf-access-control-list {
    prefix acl;
    reference
        "RFC 8519: YANG Data Model for Network Access Control Lists (ACLs)";
}

organization
    "IETF TEAS Working Group";
contact
    "WG Web: <http://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 YANG module defines a network data module for
    NRP (Network Resource Partition).

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

    Redistribution and use in source and binary forms, with or
    without modification, is permitted pursuant to, and subject
    to the license terms contained in, the Revised BSD License
    set forth in Section 4.c of the IETF Trust’s Legal Provisions
    Relating to IETF Documents

    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
    for full legal notices."

revision 2022-07-11 {
    description
        "This is the initial version of NRP YANG model.";
    reference
        "RFC XXX: A YANG Data Model for Network Resource Partition";
}

identity nrp-partition-type {
    description
        "Base identity for NRP partition type.";
}

identity nrp-control-plane-partition {

base nrp-partition-type;
description
   "Identity for control plane partition."
}

identity nrp-data-plane-partition {
    base nrp-partition-type;
description
   "Identity for data plane partition."
}

identity nrp-hybrid-plane-partition {
    base nrp-partition-type;
description
   "Identity for both planes partition."
}

identity nrp-no-partition {
    base nrp-partition-type;
description
   "Identity for no partition."
}

identity nrp-link-partition-type {
    description
   "Base identity for interface partition type."
}

identity virtual-sub-interface-partition {
    base nrp-link-partition-type;
description
   "Identity for virtual interface or sub-interface, e.g. FlexE."
}

identity queue-partition {
    base nrp-link-partition-type;
description
   "Identity for queue partition type."
}

identity nrp-dataplane-type {
    description
   "Base identity for NRP data plane type."
}

identity nrp-dataplane-ipv6 {
    base nrp-dataplane-type;
description
   "Base identity for NRP data plane type."
}
"Identity for NRP specific packet forwarding of IPv6."
}

identity nrp-dataplane-mpls {
    base nrp-dataplane-type;
    description
        "Identity for NRP specific packet forwarding of MPLS."
}

identity nrp-dataplane-sr-mpls {
    base nrp-dataplane-type;
    description
        "Identity for NRP specific packet forwarding of SR MPLS."
}

identity nrp-dataplane-srv6 {
    base nrp-dataplane-type;
    description
        "Identity for NRP specific packet forwarding of SRv6."
}

/*
 * Groupings
 */

grouping nrp-bandwidth-reservation {
    description
        "Grouping for NRP bandwidth reservation."
    container bandwidth-reservation {
        description
            "Container for NRP bandwidth reservation."
        choice bandwidth-type {
            description
                "Choice of bandwidth reservation type."
            case bandwidth-value {
                leaf bandwidth-value {
                    type uint64;
                    units "bps";
                    description
                        "Bandwidth allocation for the NRP as absolute value."
                }
            }
            case bandwidth-percentage {
                leaf bandwidth-percent {
                    type rt-types:percentage;
                    description
                        "Bandwidth allocation for the NRP as a percentage of a link."
                }
            }
        }
    }
}
grouping nrp-control-plane-attributes {
  description "Grouping for NRP control plane attributes.";
  container control-plane {
    description "The container of NRP control plane mechanisms.";
    container topology-ref {
      description "Container for topology reference.";
      container igp-topology-ref {
        description "Container for IGP topology reference.";
        uses nw:network-ref;
        leaf multi-topology-id {
          type uint32;
          description "The MT-id of an NRP.";
        }
        leaf flex-algo-id {
          type uint32;
          description "The flex-algo-id of an NRP.";
        }
      }
      uses te-types:te-topology-identifier;
    }
  }
}

grouping nrp-data-plane-attributes {
  description "Grouping for NRP data plane attributes.";
  container data-plane {
    description "The data plane mechanisms of an NRP. The forwarding plane could be MPLS, IPv6, SRv6, or SR-MPLS.";
    container global-resource-identifier {
      description "The container of global NRP data-plane ID.";
      container nrp-dataplane-ipv6-type {
        description "The container of IPv6 based NRP data-plane identifier.";
      }
    }
  }
}
leaf nrp-dp-value {
    type inet:ipv6-address;
    description
        "Indicates the IPv6 NRP data-plane identifier.";
}

container nrp-dataplane-mpls-type {
    description
        "The container of MPLS based NRP data-plane identifier.";
    leaf nrp-dp-value {
        type uint32;
        description
            "Indicates MPLS metadata values to identify MPLS NRP
data plane identifier, e.g. Ancillary data.";
    }
}

container nrp-aware-dp {
    description
        "The container of SR based NRP data-plane identifier.";
    container nrp-aware-srv6-type {
        presence "Enables SRv6 data plane type.";
        description
            "The container of SRv6 based NRP data-plane identifier.";
    }
    container nrp-aware-sr-mpls-type {
        presence "Enables SR MPLS data plane type.";
        description
            "The container of SR MPLS based NRP data-plane identifier.";
    }
}

grouping nrp-traffic-steering-policy {
    description
        "The grouping of the NRP traffic steering policy.";
    container steering-policy {
        description
            "The container of a policy set
matching an NRP traffic classifier.";
    leaf-list color-id {
        type uint32;
        description
            "A list of color ID for NRP traffic steering based on
SR policy.";
    }
    leaf-list acl-ref {

type leafref {
    path "/acl:acls/acl:acl/acl:name";
} 

description
    "A list of ACL for NRP traffic classification."

}

}


grouping nrp-aware-id {
    description
        "The grouping of NRP aware dataplane ID.";
    container nrp-aware-dp-id {
        config false;
        description
            "The container of NRP data plane identifier.";
        leaf nrp-dp-srv6 {
            type srv6-types:srv6-sid;
            description
                "Indicates the SRv6 SID value as the NRP data plane identifier.";
        }
        leaf nrp-dp-sr-mpls {
            type rt-types:mpls-label;
            description
                "Indicates the SR MPLS ID value as the NRP data plane identifier.";
        }
    }
}

}


grouping nrp-topology-attributes {
    description
        "NRP global attributes.";
    container nrp {
        description
            "Containing NRP topology attributes.";
        leaf nrp-id {
            type uint32;
            description
                "NRP identifier.";
        }
        leaf nrp-name {
            type string;
            description
                "NRP Name.";
        }
        leaf partition-type {

type identityref {
    base nrp-partition-type;
}
default "nrp-no-partition";
description
    "Indicates the resource partition type of the NRP, such as
    control plane partition, data plane partition,
    or no partition.";

uses nrp-bandwidth-reservation;
uses nrp-control-plane-attributes;
uses nrp-data-plane-attributes;
uses nrp-traffic-steering-policy;
list nrp-topology-group {
    key "group-id";
description
    "List of groups for NRP topology elements (node or links)
    that share common attributes.";
leaf group-id {
    type string;
description
    "The NRP topology group identifier.";
}
container base-topology-ref {
    description
    "Container for the base topology reference.";
    uses nw:network-ref;
}
list links {
    key "link-ref";
description
    "A list of links with common attributes";
leaf link-ref {
    type leafref {
        path
    }
    description
    "A reference to a link in the base topology.";
}
container link-attributes-override {
    description
    "Container for overriding link attributes,
    e.g. resource reservation.";
    uses nrp-bandwidth-reservation;
}
uses nrp-bandwidth-reservation;
}
}

// nrp
}

// nrp-node-attributes

grouping nrp-node-attributes {
  description
  "NRP node scope attributes.";
  container nrp {
    config false;
    description
    "Containing NRP attributes.";
    uses nrp-aware-id;
  }
}

// nrp-node-attributes

grouping nrp-link-states {
  description
  "NRP link scope states.";
  container nrp {
    description
    "Containing NRP attributes.";
    uses nrp-bandwidth-reservation;
    leaf partition-type {
      type identityref {
        base nrp-partition-type;
      }
      description
      "Indicates the resource partition type of a link.";
    }
    uses nrp-aware-id;
    uses nrp-statistics-per-link;
  }
}

// one-way-performance-metrics

grouping one-way-performance-bandwidth {
  description
  "Grouping for one-way performance bandwidth.";
  leaf one-way-available-bandwidth {
    type rt-types:bandwidth-ieee-float32;
units "bytes per second";
default "0x0p0";
description
"Available bandwidth that is defined to be NRP link bandwidth minus bandwidth utilization. For a bundled link, available bandwidth is defined to be the sum of the component link available bandwidths.";

leaf one-way-utilized-bandwidth {
type rt-types:bandwidth-ieee-float32;
units "bytes per second";
default "0x0p0";
description
"Bandwidth utilization that represents the actual utilization of the link (i.e. as measured in the router). For a bundled link, bandwidth utilization is defined to be the sum of the component link bandwidth utilizations.";
}

// nrp-link-statistics

grouping nrp-statistics-per-link {
description
"Statistics attributes per NRP link.";
container statistics {
config false;
description
"Statistics for NRP link.";
leaf admin-status {
type te-types:te-admin-status;
description
"The administrative state of the link.";
}
leaf oper-status {
type te-types:te-oper-status;
description
"The current operational state of the link.";
}
uses one-way-performance-bandwidth;
uses te-packet-types:one-way-performance-metrics-packet;
}


grouping nrp-augment {
description
"Augmentation for NRPs.";

container nrp {
    presence "NRP support";
    description
        "Indicates NRP support.";
}
// nrp

// nrp-augment

augment "/nw:networks/nw:network/nw:network-types" {
    description
        "Defines the NRP topology type.";
    container nrp {
        presence "Indicates NRP topology";
        description
            "The presence identifies the NRP type.";
    }
}

augment "/nw:networks/nw:network" {
    when 'nw:network-types/nrp:nrp' {
        description
            "Augment only for NRP topology.";
    }
    description
        "Augment NRP configuration and state.";
    uses nrp-topology-attributes;
}

augment "/nw:networks/nw:network/nw:node" {
    when '../nw:network-types/nrp:nrp' {
        description
            "Augment only for NRP topology.";
    }
    description
        "Augment node configuration and state.";
    uses nrp-node-attributes;
}

augment "/nw:networks/nw:network/nt:link" {
    when '../nw:network-types/nrp:nrp' {
        description
            "Augment only for NRP topology.";
    }
    description
        "Augment link configuration and state.";
    uses nrp-link-states;
}
8. 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.

nrp-link: A malicious client could attempt to remove a link from a topology, add a new link. In each case, the structure of the topology would be sabotaged, and this scenario could, for example, result in an NRP topology that is less than optimal.

The entries in the nodes above include the whole network configurations corresponding with the NRP, 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.

9. 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:

- 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-nrp
Prefix: nrp
Reference: RFC XXXX

10. Contributor

Zhenbin Li
Huawei
Email: lizhenbin@huawei.com

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

11. References

11.1. Normative References

[I-D.dong-6man-enhanced-vpn-vtn-id]

[I-D.dong-idr-sr-policy-vtn]

[I-D.ietf-lsr-flex-algo]
[I-D.ietf-lsr-isis-sr-vtn-mt]
Xie, C., Ma, C., Dong, J., and Z. Li, "Using IS-IS Multi-Topology (MT) for Segment Routing based Virtual Transport Network", Work in Progress, Internet-Draft, draft-ietf-lsr-isis-sr-vtn-mt-03, 10 July 2022,

[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", Work in Progress, Internet-Draft, draft-ietf-spring-sr-for-enhanced-vpn-02, 5 March 2022,

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
DOI 10.17487/RFC3688, January 2004,

RFC 4915, DOI 10.17487/RFC4915, June 2007,

RFC 5120, DOI 10.17487/RFC5120, February 2008,

RFC 6241, DOI 10.17487/RFC6241, June 2011,

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)",
RFC 6242, DOI 10.17487/RFC6242, June 2011,

RFC 7950, DOI 10.17487/RFC7950, August 2016,

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG",
RFC 7951, DOI 10.17487/RFC7951, August 2016,
11.2. Informative References

[I-D.dong-teas-nrp-scalability]

Appendix A. An Example

This section contains an example of an instance data tree in JSON encoding [RFC7951]. The example instantiates ietf-nrp for the topology that is depicted in the following diagram. There are three nodes, D1, D2, and D3. D1 has three termination points, 1-0-1, 1-2-1, and 1-3-1. D2 has three termination points as well, 2-1-1, 2-0-1, and 2-3-1. D3 has two termination points, 3-1-1 and 3-2-1. In addition there are six links, two between each pair of nodes with one going in each direction.
The corresponding NRP instance data tree is depicted below:

```json
{
  "ietf-network:networks": {
    "network": [
      {
        "network-types": {
          "ietf-nrp:nrp": {}
        },
        "network-id": "nrp-example",
        "ietf-nrp:nrp": {
          "nrp-id": "1",
          "nrp-name": "NRP1",
          "partition-type": "nrp-data-plane-partition",
          "bandwidth-reservation": {
            "bandwidth-value": "10000"
          },
          "control-plane": {
            "topology-ref": {
              "igp-topology-ref": {
                "network-ref": "L3-topology-1",
                "flex-algo-id": "129"
              }
            }
          }
        }
      }
    ]
  }
}
```

Figure 4: An NRP Instance Example
"data-plane": {
  "global-resource-identifier": {
    "nrp-dataplane-ipv6-type": {
      "nrp-dp-value": "100"
    }
  }
},
"steering-policy": {
  "color-id": "100"
},
"nrp-topology-group": [
  {
    "group-id": "group1",
    "base-topology-ref": {
      "network-ref": "native-topology"
    }
  }
],
"node": [
  {
    "node-id": "D1",
    "termination-point": [
      {
        "tp-id": "1-0-1"
      },
      {
        "tp-id": "1-2-1"
      },
      {
        "tp-id": "1-3-1"
      }
    ]
  },
  {
    "node-id": "D2",
    "termination-point": [
      {
        "tp-id": "2-0-1"
      },
      {
        "tp-id": "2-1-1"
      },
      {
        "tp-id": "2-3-1"
      }
    ]
  }
]
"node-id": "D3",
"termination-point": [
{
},
{
"tp-id": "3-2-1"
}
]
],
"ietf-network-topology:link": [
{
"link-id": "D1,1-2-1,D2,2-1-1",
"source": {
"source-node": "D1",
"source-tp": "1-2-1"
},
"destination": {
"dest-node": "D2",
"dest-tp": "2-1-1"
},
"ietf-nrp:nrp": {
"partition-type": "virtual-sub-interface-partition",
"bandwidth-reservation": {
"bandwidth-value": "10000"
}
}
},
{
"link-id": "D2,2-1-1,D1,1-2-1",
"source": {
"source-node": "D2",
"source-tp": "2-1-1"
},
"destination": {
"dest-node": "D1",
"dest-tp": "1-2-1"
},
"ietf-nrp:nrp": {
"partition-type": "virtual-sub-interface-partition",
"bandwidth-reservation": {
"bandwidth-value": "10000"
}
}
},
{
"link-id": "D1,1-3-1,D3,3-1-1",
"source": {
"source-node": "D1",
"source-tp": "1-3-1"
},
"destination": {
"dest-node": "D3",
"dest-tp": "3-1-1"
},
"ietf-nrp:nrp": {
"partition-type": "virtual-sub-interface-partition",
"bandwidth-reservation": {
"bandwidth-value": "10000"
}
}
}]}
"source-node": "D1",
"source-tp": "1-3-1"
},
"destination": {
  "dest-node": "D3",
  "dest-tp": "3-1-1"
},
"ietf-nrp:nrp": {
  "partition-type": "virtual-sub-interface-partition",
  "bandwidth-reservation": {
    "bandwidth-value": "10000"
  }
}
},
{
  "link-id": "D3,3-1-1,D1,1-3-1",
  "source": {
    "source-node": "D3",
    "source-tp": "3-1-1"
  },
  "destination": {
    "dest-node": "D1",
    "dest-tp": "1-3-1"
  },
  "ietf-nrp:nrp": {
    "partition-type": "virtual-sub-interface-partition",
    "bandwidth-reservation": {
      "bandwidth-value": "10000"
    }
  }
},
{
  "link-id": "D2,2-3-1,D3,3-2-1",
  "source": {
    "source-node": "D2",
    "source-tp": "2-3-1"
  },
  "destination": {
    "dest-node": "D3",
    "dest-tp": "3-2-1"
  },
  "ietf-nrp:nrp": {
    "partition-type": "virtual-sub-interface-partition",
    "bandwidth-reservation": {
      "bandwidth-value": "10000"
    }
  }
},
}
Figure 5: Instance data tree

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 560066
Karnataka
India
Email: dhruv.ietf@gmail.com