TEAS
Internet-Draft
Intended status: Standards Track
Expires: 25 April 2024

B. Wu
D. Dhody
Huawei Technologies
R. Rokui
Ciena
T. Saad
J. Mullooly
Cisco Systems, Inc

23 October 2023

A YANG Data Model for the IETF Network Slice Service
draft-ietf-teas-ietf-network-slice-nbi-yang-08

Abstract

This document defines a YANG data model for the IETF Network Slice Service. The model can be used in the IETF Network Slice Service interface between a customer and a provider that offers 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 25 April 2024.

Table of Contents

1. Introduction

This document defines a YANG [RFC7950] data model for the IETF Network Slice Service as defined in [I-D.ietf-teas-ietf-network-slices].i
M

This YANG module can be used in the IETF Network Slice Service Interface exposed by a provider to its customers (including of provider's internal use) in order to manage (e.g., subscribe, delete, or change) IETF Network Slice Services. The agreed service will then trigger the appropriate IETF Network Slice operation, such as instantiating, modifying, or deleting an IETF Network Slice.

As discussed in [I-D.ietf-teas-ietf-network-slices], the mapping between an IETF Network Slice Service and its realization is implementation and deployment specific.

The IETF Network Slice Service Model (NSSM) focuses on the requirements of an IETF Network Slice Service from the point of view

Commenté [BMI1]: Might update the terminology to align with the outcome of the IESG review of the framework.

of the customer, not how it is implemented within a provider network. The module is classified as customer service model (Section 2 of [RFC8309]).

The IETF Network Slice Service YANG model conforms to the Network Management Datastore Architecture (NMDA) [RFC8342].

Editorial Note: (To be removed by RFC Editor)

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

This document also makes use of the terms defined in [I-D.ietf-teas-ietf-network-slices]:

* Attachment Circuit (AC): See Section 3.2 of [I-D.ietf-teas-ietf-network-slices].

* Connectivity Construct: See Sections 3.2 and 4.2.1 of [I-D.ietf-teas-ietf-network-slices].

* Customer: See Section 3.2 of [I-D.ietf-teas-ietf-network-slices].

* Customer Higher-level Operation System: See Section 6.3.1 of [I-D.ietf-teas-ietf-network-slices].

* Service Demarcation Point (SDP): See Sections 3.2 and 5.2 [I-D.ietf-teas-ietf-network-slices].

In addition, this document defines the following term:

* Connection Group: Refers to one or more connectivity constructs that are grouped for administrative purposes, such as the following:

  Combine multiple connectivity constructs to support a set of well-known connectivity service types, such as bidirectional unicast service, multipoint-to-point (MP2P) service, or hub-and-spoke service.

  Assign the same SLO/SLE policies to multiple connectivity constructs unless SLO/SLE policy is explicitly overridden at the individual connectivity construct level.

  Share specific SLO limits within multiple connectivity constructs.

2.1. Acronyms

The following acronyms are used in the document:

A2A	Any-to-any
AC	Attachment Circuit
CE	Customer Edge
NSC	Network Slice Controller
NSSM	Network Slice Service Model
MTU	Maximum Transmission Unit
PE	Provider Edge
P2P	Point-to-point
P2MP	Point-to-multipoint
QoS	Quality of Service
SDP	Service Demarcation Point
SLE	Service Level Expectation
SLO	Service Level Objective

3. IETF Network Slice Service Overview

As defined in Section 3.2 of [I-D.ietf-teas-ietf-network-slices], an IETF Network Slice Service is specified in terms of a set of Service Demarcation Points (SDPs), a set of one or more connectivity constructs between subsets of these SDPs, and a set of Service Level Objectives (SLOs) and Service Level Expectations (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.

The SDPs serve as the IETF Network Slice Service ingress/egress points. An SDP is identified by a unique identifier in the context of an IETF Network Slice Service.

Examples of IETF Network Slice Services that contain only one connectivity construct are shown in Figure 1.

Commenté [BMI2]: Expand at first use.

Commenté [BMI3]: Section 5.2.2 says the following:

"be point-to-point (P2P), point-to-multipoint (P2MP), any-to-any (A2A) or a combination of these types."

Please align both statements

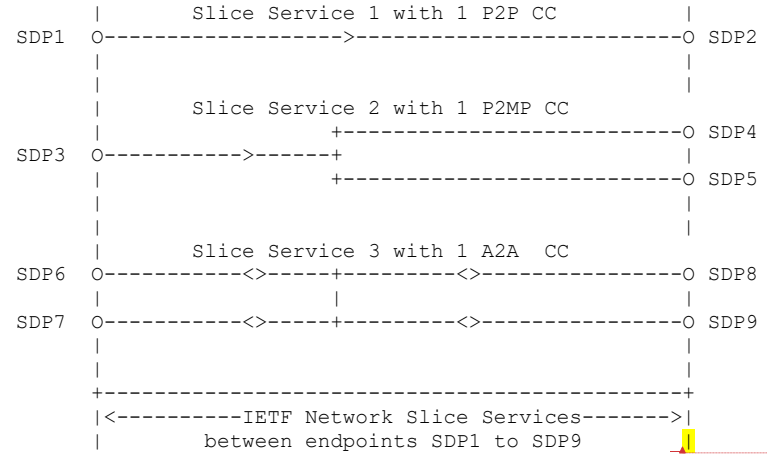


Figure 1: Examples of IETF Network Slice Services

An example of IETF Network Slice Services that contain multiple connectivity constructs is shown in Figure 2.

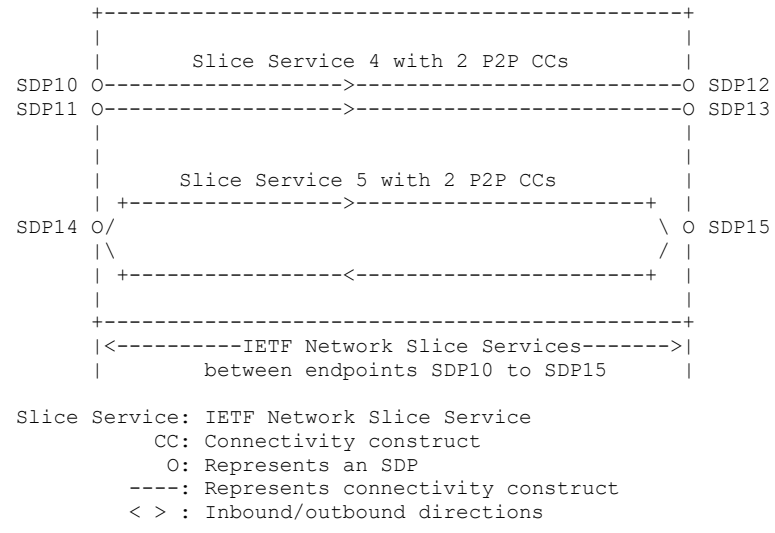


Figure 2: Examples of IETF Network Slice Services

As shown in Figure 2, the IETF Network Slice Service 4 contains two P2P connectivity constructs between the set of SDPs. The IETF Network Slice Service 5 is a bidirectional unicast service between

a mis en forme : Surlignage

SDP14 and SDP15 that consists of two unidirectional P2P connectivity constructs.

4. IETF Network Slice Service Model (NSSM) Usage

The NSSM can be used by a provider to expose its IETF Network Slice Services~~Service~~, and by a customer to manage its IETF Network Slices Services (e.g., request, delete, or modify). The details about how service requests are handled by the provider (specifically, a controller), including which network operations are triggered, are internal to the provider. The details of the IETF Network Slices realization are hidden from customers.

~~The IETF Network Slices are applicable to use cases, such as (but not limited to) network wholesale services, network infrastructure sharing among operators, Network Function Virtualization (NFV) connectivity, Data Center interconnect, and 5G.~~[I-D.ietf-teas-ietf-network-slice-use-cases] provides some sample use cases for Network Slices.

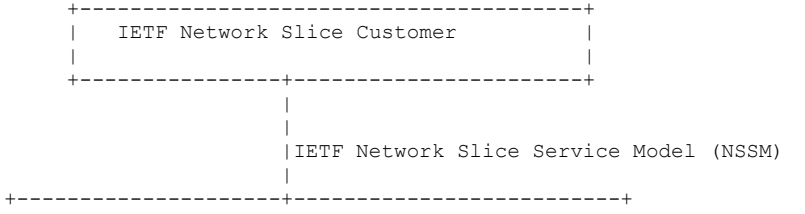
Commenté [BMI4]: Instead ..point to the use case I-D.

An IETF Network Slice Controller (NSC) is an entity that exposes the IETF Network Slice Service Interface to customers to manage IETF Network Slice Services. Typically, an NSC receives requests from its customer-facing interface (e.g., from a management system). During service creation, this interface can convey data objects that 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. Depending of whether the requirements and authorization checks are met, ~~These~~ these service 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.

As shown in Figure 3, ~~in all the use cases,~~ the NSSM is used by the customer's higher level operation system to communicate with ~~the~~ an NSC for life cycle management of IETF Network Slice Services including both enablement and monitoring. For example, in the 5G ~~E2E (~~End-to-end~~)~~ network slicing use-case the ~~E2E~~ 5G network slice orchestrator acts as the higher layer system to ~~request~~ manage the IETF Network Slice Services. The interface is used to support dynamic IETF Network Slice ~~creation and its lifecycle~~ management to facilitate end-to-end network slice services.

Commenté [BMI5]: This is about the service

Commenté [BMI6]: As this is not only restricted to creation



```
|      IETF Network Slice Controller (NSC)       |
+------------------------------------------------+
```

Figure 3: IETF Network Slice Service Reference Architecture

Note: The NSSM can be used recursively (hierarchical mode), i.e., an NSSM can map to a child NSSes. As described in Section A.5 of [I-D.ietf-teas-ietf-network-slices], the IETF Network Slice Service can support a recursive composite architecture that allows one layer of IETF Network Slice Services to be used by other layers.

5. IETF Network Slice Service Model (NSSM) Description

The NSSM, "ietf-network-slice-service", includes two main data nodes: "slice-service" and "slo-sle-templates" and "slice-service" (see Figure 4).

Commenté [BMI7]: To preserve the order in the tree structure.

The "slo-sle-templates" container is used by an NSC to maintain a set of common network slice SLO and SLE templates that apply to one or several IETF Network Slice Services. Refer to Section 5.1.

The "slice-service" list includes the set of IETF Network Slice Services that are maintained by a provider. "slice-service" is the data structure that abstracts the IETF Network Slice Service. Under the "slice-service", the "sdp" list is used to abstract the SDPs. The "connection-group" is used to abstract connectivity constructs between SDPs. Refer to Section 5.2.

~~The "slo-sle-templates" container is used by an 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~~Figure 4 describes the overall tree structure of the NSSM.~~:~~

```
module: ietf-network-slice-service
  +--rw network-slice-services
     +--rw slo-sle-templates
     |  +--rw slo-sle-template* [id]
     |        ...
     +--rw slice-service* [id]
        +--rw id                              string
        +--rw description?                    string
        +--rw service-tags
        |     ...
        +--rw (slo-sle-policy)?
        |     ...
        +--rw compute-only?                   empty
        +--rw status
        |     ...
        +--rw sdps
        |     ...
        +--rw connection-groups
        |     ...
        +--rw custom-topology
              ...
```

Figure 4: The NSSM Tree Structure

5.1. IETF Network Slice Service SLO and SLE Templates

The "slo-sle-templates" container (Figure 45) is used by an IETF Network Slice Service provider to define and maintain a set of common IETF Network Slice Service templates that apply to one or several IETF Network Slice Services. The exact definition of the templates is deployment specific to each ~~network~~ provider.

```
+--rw slo-sle-templates
   +--rw slo-sle-template* [id]
      +--rw id                string
      +--rw description?      string
      +--rw template-ref?     leafref
      +--rw slo-policy
      |  +--rw metric-bound* [metric-type]
      |  |  +--rw metric-type          identityref
      |  |  +--rw metric-unit          string
      |  |  +--rw value-description?   string
      |  |  +--rw percentile-value?    percentile
      |  |  +--rw bound?               uint64
      |  +--rw availability?   decimal64
      |  +--rw mtu?            uint16
      +--rw sle-policy
         +--rw security*               identityref
         +--rw isolation*              identityref
         +--rw max-occupancy-level?    uint8
         +--rw steering-constraints
            +--rw path-constraints
            +--rw service-function
```

Figure 5: "slo-sle-templates" Subtree Structure

The NSSM includes the identifiers of SLO and SLE templates and the common attributes defined in Section 5.1 of [I-D.ietf-teas-ietf-network-slices]. Considering that there are many attributes defined and some attributes could vary with service requirements, e.g., bandwidth, or latency, multiple standard templates as well as custom "service-slo-sle-policy" are defined:

1: Standard template with no attribute specified: The exact definition of the templates is deployment specific to the provider.

2: Standard template with attributes specified: Provides the customers with the ability to define templates, or reference a predefined template "template-ref" and override specific attributes, and apply them to IETF Network Slice Service configuration.

3: Custom "service-slo-sle-policy": More description is provided in Section 5.2.3.

The following shows an example where two network slice templates can be retrieved by the customers:

Commenté [BMI8]: Indicate whether these are exposed to the customer.

Commenté [BMI9]: Do you mean that when the same attribute is covered in the template and "core" nssm, the template takes precedence?

```
   ========== NOTE: '\' line wrapping per RFC 8792 ==========

   {
     "network-slice-services": {
       "slo-sle-templates": {
         "slo-sle-template": [
           {
             "id": "PLATINUM-template",
             "description": "Two-way bandwidth: 1 Gbps,\
                             95th percentile latency 50ms",
             "slo-policy": {
               "metric-bound": [
                 {
                   "metric-type": "two-way-delay-percentile",
                   "metric-unit": "milliseconds",
                   "percentile-value": "95.000",
                   "bound": "50"
                 }
               ]
             },
             "sle-policy": {
               "isolation": ["service-traffic-isolation"]
             }
           },
           {
             "id": "GOLD-template",
             "description": "Two-way bandwidth: 1 Gbps,\
                             maximum latency 100ms",
             "slo-policy": {
               "metric-bound": [
                 {
                   "metric-type": "two-way-delay-maximum",
                   "metric-unit": "milliseconds",
                   "bound": "100"
                 }
               ]
             },
             "sle-policy": {
               "isolation": ["service-traffic-isolation"]
             }
           }
         ]
       }
     }
   }

   ========== NOTE: '\' line wrapping per RFC 8792 ==========
```

Commenté [BMI10]: As per 8792 requirements

Commenté [BMI11]: Should be prefixed

Commenté [BMI12]: Please note that the description of the leaf uses "ms". Some consistency is needed here.

Commenté [BMI13]: To insist this is "fraction-digits 3"

Commenté [BMI14]: Should be prefixed

Commenté [BMI15]: Should be prefixed

Commenté [BMI16]: Idem as above

Commenté [BMI17]: Idem as above

5.2. IETF Network Slice Services

The "slice-service" is the data structure that abstracts an IETF Network Slice Service. Each "slice-service" ~~, which~~ is uniquely identified by "id" specified in the context of an NSC.

An IETF Network Slice Service has the following main ~~parameters~~data nodes:

* "id": Is ~~an~~ a unique identifier for internal management reference of the IETF Network Slice Service within an NSC.

* "description": Provides a textual description of an IETF Network Slice Service.

* "service-tags": Indicates a management tag (e.g., ~~"~~"customer-name") that is used to correlate the operational information of Customer Higher-level Operation System and IETF Network Slices. It might be used by the IETF Network Slice Service ~~operator~~ provider to provide additional information to ~~the~~ an NSC during the automation of the IETF ~~network~~ Network ~~slices~~Slices. E.g. adding tags with "customer-name" when multiple actual customers use a same ~~network~~ Network ~~slice~~ Slice ~~service~~Service. Another use-case for "service-tag" might be for an operator to provide additional attributes to an NSC which might be used during the realization of IETF Network Slice Services such as type of services (e.g., Layer 2 or Layer 3 ~~service~~technology). These additional attributes can also be used by an NSC for various ~~use-cases~~purposes such as monitoring and assurance of the IETF Network Slice Services where the NSC can ~~notify~~ issue notifications to the customer system ~~by issuing the notifications~~. Note that all these attributes are ~~OPTIONAL~~ optional~~but might be useful for some use-cases~~.

* "slo-sle-policy": Defines SLO and SLE policies for the "slice-service". More details are provided in Section 5.2.3.

* "compute-only": Is used to check the feasibility before instantiating a Network Slice Service. More details are provided in Section 5.2.6.

* "status": Is used to show the ~~operative~~ both operational and administrative status of ~~the~~ an IETF Network Slice Service. It~~, and~~ can be used as indicator to detect ~~network~~ Network ~~slice~~ Slice Service anomalies.

* "sdps": Represents a set of SDPs that are involved in the IETF Network Slice Service with each "sdp" belonging to a single "slice-service". More details 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 Points

An SDP ~~belong~~ belongs 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"s or removing existing "sdp"s.

Commenté [BMI18]: Not sure what is meant here.

Commenté [BMI19]: For consistency with the use of "provider" in the document.

Commenté [BMI20]: I don't parse this.

Commenté [BMI21]: This contradicts this part from the framework:

"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."

Commenté [BMI22]: How the controller manages CCs that involves a deleted SDP?

```
   +--rw sdps
      +--rw sdp* [id]
         +--rw id                          string
         +--rw description?                string
         +--rw location
         |     ...
         +--rw node-id?                    string
         +--rw sdp-ip-address*             inet:ip-address
         +--rw tp-ref?                     leafref
         +--rw service-match-criteria
         |     ...
         +--rw incoming-qos-policy
         |     ...
         +--rw outgoing-qos-policy
         |     ...
         +--rw sdp-peering
         |     ...
         +--rw ac-svc-name*                string
         +--rw attachment-circuits
         |     ...
         +--rw status
         |     ...
         +--ro sdp-monitoring
               ...
```

Figure X: XXXX

Section 5.2 of [I-D.ietf-teas-ietf-network-slices] describes four possible ways in which an SDP may be placed:

* Within the CE

* Provider-facing ports on the CE

* Customer-facing ports on the PE

* Within the PE

Although there are four options, they can be categorized into two: CE-based or PE-based. To simplify the model, ~~the~~ an NSC and the customer's system can agree on the choice of these two types without marking the type on each SDP.

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 AC definition ~~of AC~~ in Section 5.2 of [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, which 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.

To abstract the SDP options and SLOs/SLEs profiles, an SDP has the following characteristics:

Commenté [BMI23]: Any reason why the grouping in https://datatracker.ietf.org/doc/rfc9179/ are not reused here?

Commenté [BMI24]: How?

* "id": Uniquely identifies the SDP within ~~the~~ an ~~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": A reference to the node that hosts the SDP, which helps the NSC to identify an SDP.

* "sdp-ip-address": The SDP IP information, which helps the NSC to identify an SDP.

* "tp-ref": A reference to a Termination Point (TP) in the custom topology defined in Section 5.2.5.

* "incoming-qos-policy" and "outgoing-qos-policy": Sets the incoming and outgoing QoS policies to apply on a given SDP, including QoS policy and specific ingress and egress traffic limits to ensure access security. When applied in the incoming direction, the policy is applicable to the traffic that passes through the AC from the customer network or from another provider's network to the Network Slice. When applied in the outgoing direction, the policy is applied to the traffic from the Network Slice towards the customer network or towards another provider's network. If an SDP has multiple ACs, the "rate-limits" of "attachment-circuit" can be set to an AC specific value, but the rate cannot exceed the "rate-limits" of the SDP. If an SDP only contains a single AC, then the "rate-limits" of "attachment-circuit" is the same with the SDP. The definition of AC refers to Section 5.2 [I-D.ietf-teas-ietf-network-slices].

* "ac-svc-name": Indicates the names of AC services, for association purposes, to refer to the ACs that have been created. When both "ac-svc-name" and the attributes of "attachment-circuits" are defined, the "ac-svc-name" takes precedence.

* "attachment-circuits": Specifies the list of ACs ~~by~~ over which the service traffic is received or be forwarded. This is an optional SDP attribute.
When an SDP has multiple ACs and some AC specific attributes are needed, each "attachment-circuit" can specify attributes, such as interface specific IP addresses, service MTU, etc.

* "sdp-peering": Specifies the peers and peering protocols for an SDP to exchange control ~~-~~ plane information, e.g., Layer 1 signaling
protocol or Layer 3 routing protocols, etc.

  - "peer-sap-id": Indicates the references to the remote endpoints of attachment circuits. This information can be used for correlation purposes, such as identifying Service Attachment Points (SAPs) defined in [RFC9408], which defines a model of an abstract view of the provider network topology that contains the points from which the services can be attached.

  - "protocols": Serves as an augmentation target. Appendix A The

Commenté [BMI25]: You may add some information about how the customer has access to this information, especially for PE option described above.

Commenté [BMI26]: As many classes may present in the same NSS, the model should be updated to allow such option. Please see L2NM, for example.

example protocols of an SDP can be BGP, static routing, etc.

* "status": Enables the control of the ~~operative and~~ administrative status and report the operational status of the SDP. These status values ~~,~~ can be used as indicator to detect SDP anomalies.

* "service-match-criteria": Defines matching policies for the IETF Network Slice Service traffic to apply on a given SDP.

Depending on the requirements of different cases, "service-match-criteria" can be used for the following purposes:

* Specify the AC type: physical or logical connection

* Distinguish the SDP traffic if the SDP is located in the CE or PE

* Distinguish the traffic of different connection groups (CGs) or connectivity constructs (CCs) when multiple CGs/CCs of different SLO/SLE may be set up between the same pair of SDPs, as illustrated in Figure 5. 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.

* Define specific out-of-profile policies: The customer may choose to use an explicit "service-match-criteria" to map any SDP's traffic or a subset of the SDP's traffic to a specific connection group or connectivity construct. If a subset of traffic is matched (e.g., 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~~ are no additional connectivity constructs where the "sdp-id" source is specified, then that traffic will be dropped.

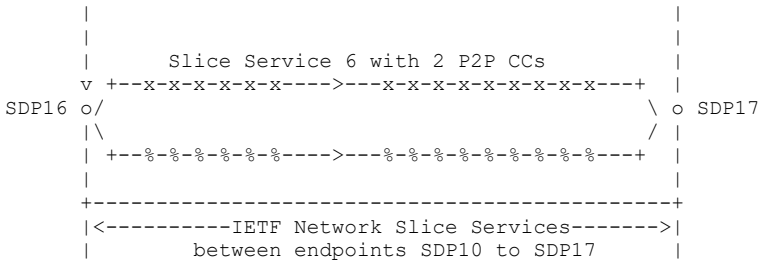


Figure 5: Application of Match Criteria

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

Commenté [BMI27]: Please fix this sentence.

Commenté [BMI28]: These are not shown in the tree diagram

Commenté [BMI29]: Not shown in the tree diagram

a mis en forme : Retrait : Gauche : 1,25 cm

a mis en forme : Surlignage

the port or AC can be used to unambiguously identify the traffic and the SDP is the only source of the connectivity-construct. Appendix B.1 shows an example of both the implicit and explicit approaches.

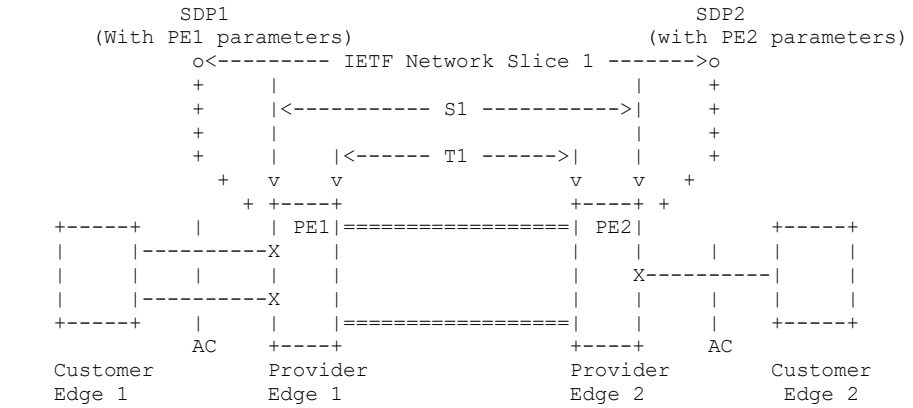
While explicit matching is optional in some use cases, it provides a more clear and readable implementation, but the choice is left to the operator.

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.

Commenté [BMI30]: Please add an explicit Figure X

* SDPs at customer-facing ports on the PEs: As shown in Figure 6—, a customer of the IETF Network Slice Service would like to connect two SDPs to satisfy specific service needs, e.g., network wholesale services. In this case, the IETF Nnetwork ~~slice~~ Slice SDPs are mapped to customer-facing ports of PE nodes. The NSC uses "node-id" (PE device ID), "attachment-circuits" (ACs) to map SDPs to the customer-facing ports on the PEs.

Commenté [BMI31]: What about ac-svc-name? Idem for peer-sap-id?



Legend:
o: Representation of an SDP
+: Mapping of an SDP to customer-facing ports on the PE
X: Physical interfaces used for realization of the 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

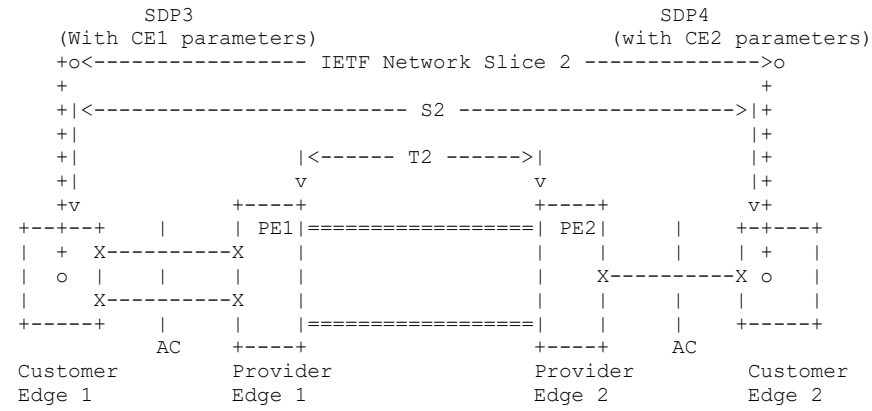
Figure 6: TITLE….

* SDPs within CEs: As shown in Figure 7—, a 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 NSC uses "node-id" (CE device ID), "sdp-ip-address" (IP address of SDP for management), "service-match-criteria" (VLAN tag), "attachment-circuits" (CE ACs) to map SDPs to the CE. The NSC can use these CE parameters (and

Commenté [BMI32]: This id may not be unique in the scope of NSC as this belong to the customer. Please elaborate how unicity is ensured here.

Commenté [BMI33]: What about ac-svc-name?

optionally the "peer-sap-id") to retrieve the corresponding PE device, interface and AC mapping details to complete the end-to-end network slice service provisioning (the implementation details are left to the NSC provider).



Legend:
o: Representation of an SDP
+: Mapping of an SDP to CE
X: Physical interfaces used for realization of the IETF Network Slice Service
S2: L0/L1/L2/L3 services used for realization of the IETF Network Slice Service
T2: Tunnels used for realization of IETF network slice

Figure 7: Title XXXX

5.2.2. IETF Network Slice Service Connectivity Constructs

Based on the customer's service traffic requirements, an IETF Network Slice Service connectivity type ~~could~~ may be ~~point-to-point (P2P),~~ ~~point-to-multipoint (P2MP),~~ ~~any-to-any (A2A),~~ or a combination ~~of these types~~thereof.

```
+--rw connection-groups
  +--rw connection-group* [id]
     +--rw id                                  string
     +--rw connectivity-type?                  identityref
     +--rw (slo-sle-policy)?
     |  +--:(standard)
     |  |  +--rw slo-sle-template?             -> /network-slice-
services/slo-sle-templates/slo-sle-template/id
     |  +--:(custom)
     |     +--rw service-slo-sle-policy
     |           ...
     +--rw service-slo-sle-policy-override?    identityref
     +--rw connectivity-construct* [id]
     |  +--rw id                                    uint32
     |  +--rw (type)?
     |  |  +--:(p2p)
```

Commenté [BMI34]: Please print tree with -69 limit.

```
      |  |  |     ...
      |  |  +--:(p2mp)
      |  |  |     ...
      |  |  +--:(a2a)
      |  |        ...
      |  +--rw (slo-sle-policy)?
      |  |  +--:(standard)
      |  |  |     ...
      |  |  +--:(custom)
      |  |        ...
      |  +--rw service-slo-sle-policy-override?     Identityref
```

Figure X: XXXX

~~[I-D.ietf-teas-ietf-network-slices] defines the basic connectivity construct (CC) for an IETF Network Slice Service, and the~~A 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.

Commenté [BMI35]: Do you mean CC? Or communications?

Commenté [BMI36]: Idem

"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. Additionally, in the case of hub and spoke connectivity, it may be inefficient when there are a large number of ~~SDP~~ SDPs with the multiple CCs. As illustrated in Appendix B.3, "connectivity-type" of "ietf-vpn-common:hub-spoke" and "connection-group-sdp-role" of "ietf-vpn-common:hub-role" or "ietf-vpn-common:spoke-role" can be specified [RFC9181].

5.2.3. IETF Network Slice Service SLO and SLE Policy

As defined in ~~section~~ Section 5 of [I-D.ietf-teas-ietf-network-slices], the SLO and SLE policy of the IETF Network Slice Services define some common attributes.

"slo-sle-policy" is used to represent these 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 override precedence order among them is as follows:

* Connectivity-construct at an 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. Appendix B.5 gives an example of the preceding policy, which shows a slice service having an A2A connectivity as default and several specific SLO connections.

The SLO attributes ~~are as follows, including~~ include performance metric attributes, availability, and MTU.

The list "metric-bound" supports the generic performance metric variations and the combinations and each "metric-bound" could specify a particular "metric-type". "metric-type" is defined with YANG identity and supports the following options:

"one-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two SDPs. ~~And t~~The bandwidth is unidirectional.

"two-way-bandwidth": Indicates the guaranteed minimum bandwidth between any two SDPs. ~~And T~~the bandwidth is bidirectional.

"one-way-delay-maximum": Indicates the maximum one-way latency between two SDPs.

"two-way-delay-maximum": Indicates the maximum round-trip latency between two SDPs.

"one-way-delay-percentile": Indicates the percentile objective of the one-way latency between two SDPs.

"two-way-delay-percentile": Indicates the percentile objective of the round-trip latency between two SDPs.

"one-way-delay-variation-maximum": 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.

"two-way-delay-variation-maximum": 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.

"one-way-delay-variation-percentile": Indicates the percentile objective of the delay variation, and is measured by the difference in the one-way latency between sequential packets in a flow.

"two-way-delay-variation-percentile": Indicates the percentile objective of the delay variation, and is measured by the difference in the two-way latency between sequential packets in a flow.

"one-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two SDPs.

"two-way-packet-loss": Indicates maximum permissible packet loss rate, which is defined by the ratio of packets dropped to packets transmitted between two SDPs.

"availability": Specifies service availability defined as the ratio of uptime to the sum of uptime and downtime, where uptime is the time the IETF Network Slice is available in accordance with the SLOs associated with it.

"mtu": Refers to the service MTU. The ~~service~~ provider MUST support customer traffic using any PDU up to this size.

The following ~~common~~ SLEs are defined:

"security": The security leaf-list defines the list of security Functions (authentication, encryption, etc.) that the customer requests the operator to apply to traffic between the two SDPs~~, including authentication, encryption, etc~~.

"isolation": Specifies the isolation types that a customer expects.

"max-occupancy-level": Specifies the number of flows that the operator admits.

"steering-constraints": Specifies the constraints the customer requests the operator to ~~route~~ steer traffic for the IETF Network Slice Service.

~~The following~~Figure X shows an example ~~where~~ with a network slice policy ~~can be configured:~~.

```
{
  "slice-services": {
    "slice-service": {
      "id": "exp-slice",
      "service-slo-sle-policy": {
        "description": "video-service-policy",
        "slo-policy": {
          "metric-bound": [
            {
              "metric-type": "one-way-bandwidth",
              "metric-unit": "Mbps",
              "bound": "1000"
            },
            {
              "metric-type": "two-way-delay-maximum",
```

Commenté [BMI37]: Add refs to the IPPM RFCs that defines one-way delay, etc.

Commenté [BMI38]: May be worth to indicates which layers are included.

Commenté [BMI39]: Not sure the normative language is justified here.

I would add a mention about the behavior if a packet exceeds this MUT: discard/fragment?

Commenté [BMI40]: Add a pointer to where islation is discussed in the framework

Commenté [BMI41]: How the flow is defined?

Commenté [BMI42]: Should be prefixed

Commenté [BMI43]: Should be prefixed

```
                "metric-unit": "milliseconds",
                "bound": "10"
              }
            ],
            "availability": "ietf-nss:level-4",
            "mtu": "1500"
          }
        }
      }
    }
  }
```

For more complex slicing scenarios, for example a multiple connectivity-construct slice service, an "override" option is provided to completely override all or part of ~~the~~ an "slo-sle" template with new values. For example, if a particular connection-group or a connectivity-construct has a unique bandwidth or latency setting, that are different from those defined in the slice service, a new set of SLOs/SLEs with full or partial override can be applied. In the case of partial override, only the newly specified parameters are replaced from the original template, while maintaining on pre-existing parameters not specified. While a full override removes all pre-existing parameters, and in essence starts a new set of SLOs/SLEs which are specified. The "service-slo-sle-policy-override" is used to specify the requirements.

Commenté [BMI44]: I'm not sure the intended behavior is clear. Also, please make sure that the override behavior defined here is consistent with the one defined for template discussed earlier in the document.

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".~~

Th~~is model also~~e NSSM describes operational and performance status of an IETF Network Slice Service. The statistics are described in the following granularity:

Commenté [BMI45]: I may miss other similar uses in the document. Please make sure your reason about the service.

* 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~~ Mechanisms such as those defined in [RFC8640] and [RFC8641] can be used for either periodic or on-demand subscription.

a mis en forme : Surlignage

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

Commenté [BMI46]: Consider having an example in the appendix

IETF Network Slice Service running status.

5.2.5. IETF Network Slice Service on Custom Topology

The Slice Service customer might ~~ask~~ request for some level of control over the topology or resources constraints. "custom-topology" is defined as an augmentation target that references the context topology. The leaf "network-ref" under this container is used to reference a predefined topology as a customized topology constraint for an Network Slice Service. ~~As per~~ Section 1 ~~in~~ of [RFC8345] defines a general abstract topology concept to accommodate both the provider's resource capability and the customer's preferences. The abstract topology is a topology that contains abstract topological elements (nodes, links, and termination points).

This document defines only the minimum attributes of ~~the~~ a custom topology, which can be extended based on the implementation requirements.

The following nodes are defined for the custom topology~~.~~:

"custom-topology": This container ~~is served~~serves as an augmentation target for the Slice Service topology context, which can be multiple. This node is located directly under the "~~Slice~~slice-~~Services~~service" list.

"network-ref": This leaf is under the container "custom-topology", which is defined to reference a predefined topology as a customized topology constraint for ~~an~~ a Network Slice Service, such as a VN topology to customize the service paths in a ~~network~~ Network ~~slice~~Slice by using type 2 Virtual Network (VN) defined in ~~section~~ Section 2.2 of [I-D.ietf-teas-actn-vn-yang] ~~,or~~, or ~~an~~ a SAP topology to request SDP feasibility checks on a ~~Service Attachment Points (~~SAP~~s)~~ network topology described in Section 3 of [RFC9408].

"tp-ref": A reference to Termination Point (TP) in the custom topology, under the list "sdp", can be used to associate an SDP with a TP of the customized topology. The TPs can be access points of the VN topology or parent termination points of the SAP topology.

5.2.6. IETF Network Slice Service Compute

An IETF Network Slice Service is, by default, provisioned so that it can instantiate ~~instantiated~~ and trigger ~~deliver the~~ service delivery. ~~The~~ An IETF Network Slice Service customer may request to check the feasibility of a request before instantiating a Network Slice Service. In such a case, the IETF Network Slice Service is configured in "compute-only" mode to distinguish it from the default behavior.

A "compute-only" Network Slice Service is configured as usual with the

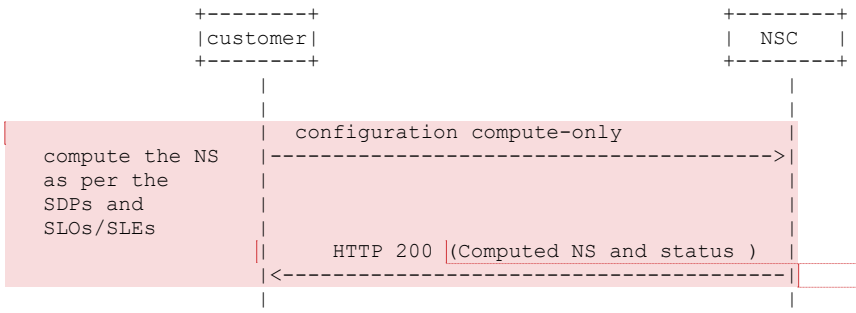
Commenté [BMI47]: exapnd

Commenté [BMI48]: You may explain why a specific node is needed compared to using native NETCONF options such as:

==
test-option: The <test-option> element MAY be specified only if the device advertises the :validate:1.1 capability ([Section 8.6](https://www.rfc-editor.org/rfc/rfc6241#section-8.6)). The <test-option> element has one of the following values: test-then-set: Perform a validation test before attempting to set. If validation errors occur, do not perform the <edit-config> operation. This is the default test-option. set: Perform a set without a validation test first. test-only: Perform only the validation test, without attempting to set.
==

Commenté [BMI49]: May be mention that this is only for creation/modification, not for deletion ?

associated per slice SLOs/SLEs. The NSC computes the feasible connectivity constructs to the configured SLOs/SLEs. This computation does not create the Network Slice or reserve any resources in the provider's network, it simply computes the resulting Network Slice based on the request. The Network Slice "administrative-status" and the connection groups or connectivity construct list are used to convey the result. For example, "admin-pre-deployment" can be used to show the status.



NS: IETF Network Slice

Figure X: XXX

6. IETF Network Slice Service Module

The "ietf-network-slice-service" module uses types defined in [RFC6991], [RFC8345], [RFC9181], [RFC8776], and [RFC7640].

<CODE BEGINS> file "ietf-network-slice-service@2023-10-23.yang"

```
module ietf-network-slice-service {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-network-slice-service";
  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-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
```

Commenté [BMI50]: Please check

Commenté [BMI51]: Why not defining a new type for the intended usage?

Commenté [BMI52]: I would not mention the protocol used in the interface.

Commenté [BMI53]: What if the check is not available immediately?

```
          "RFC 8345: A YANG Data Model for Network
                     Topologies, Section 6.2";
      }
      import ietf-te-packet-types {
        prefix te-packet-types;
        reference
          "RFC 8776: Common YANG Data Types for Traffic Engineering,
                     Section 5";
      }

      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: Bo Wu
              <lana.wubo@huawei.com>
         Editor: Dhruv Dhody
              <dhruv.ietf@gmail.com>
         Editor: Reza Rokui
              <rrokui@ciena.com>
         Editor: Tarek Saad
              <tsaad@cisco.com>
         Editor: John Mullooly
              <jmullool@cisco.com>";
      description
        "This YANG module defines a model for the IETF Network Slice
         Service.

         The model fully conforms to the Network Management
         Datastore Architecture (NMDA).

         Copyright (c) 2023 IETF Trust and the persons identified as
         authors of the code.  All rights reserved.

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

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

      revision 2023-10-23 {
        description
          "Initial revision.";
        reference
          "RFC XXXX: A YANG Data Model for the IETF Network Slice
Service";
      }

      /* Features */
      /* Identities */
```

```
  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, e.g., Layer 2 or
       Layer 3 service.";
  }

  identity service-tag-opaque {
    base service-tag-type;
    description
      "The IETF Network Slice Service opaque tag typeI.";
  }

  identity attachment-circuit-tag-type {
    description
      "Base identity for the attachment circuit tag type.";
  }

  identity vlan-id {
    base attachment-circuit-tag-type;
    description
      "Identity for VLAN ID tag type, e.g., dot1Q or QinQ VLAN IDs.";
  }

  identity ip-mask {
    base attachment-circuit-tag-type;
    description
      "Identity for IP address mask tag type.";
  }

  identity service-isolation-type {
    description
      "Base identity for IETF Network Slice Service isolation type.";
  }

  identity service-traffic-isolation {
    base service-isolation-type;
    description
      "Specify Specifies the requirements for separating the traffic
of the
       customer's IETF Network Slice Service from other services,
       which may be provided by the service provider using VPN
       technologies, such as L3VPN, L2VPN, EVPN, etc.";
  }

  identity service-security-type {
    description
```

Commenté [BMI54]: No need to repeat the "service-tag" already in the base identity.

Commenté [BMI55]: What is a "customer ID tag"?

Commenté [BMI56]: This does not add much compared to the name

Commenté [BMI57]: Consider expanding

Commenté [BMI58]: ip-address-mask?

Commenté [BMI59]: This is simply an identity!

```
    "Base identity for IETF Network Slice Service security type.";
}

identity authentication {
  base service-security-type;
  description
    "Indicates that the Slice Service requires authentication.";
}

identity integrity {
  base service-security-type;
  description
    "Indicates that the Slice Service requires data integrity.";
}

identity encryption {
  base service-security-type;
  description
    "Indicates that 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-multipoint IETF Network Slice
     Service connectivity.";
}

identity multipoint-to-multipoint {
  base vpn-common:vpn-topology;
  description
    "Identity for multipoint-to-multipoint IETF Network Slice
     Service connectivity.";
}

identity multipoint-to-point {
  base vpn-common:vpn-topology;
  description
    "Identity for multipoint-to-point IETF Network Slice
     Service connectivity.";
}

identity sender-role {
  base vpn-common:role;
  description
    "Indicates that an SDP is acting as a sender.";
}

identity receiver-role {
  base vpn-common:role;
  description
    "Indicates that an SDP is acting as a receiver.";
```

```
}

identity service-slo-metric-type {
  description
    "Base identity for IETF Network Slice Service SLO metric type.";
}

identity 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 two-way-bandwidth {
  base service-slo-metric-type;
  description
    "SLO bandwidth metric. Minimum guaranteed bandwidth between
     two SDPs at any time.";
}

identity 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 one-way-delay-maximum {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is the upper bound of network
     delay when transmitting between two SDPs.";
  reference
    "RFC_7679: A One-Way Delay Metric for IP Performance
     _________Metrics (IPPM)";
}

identity one-way-delay-percentile {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is percentile objective of
     network delay when transmitting between two SDPs.
     The metric is defined in RFC7679.";
  reference
    "RFC_7679: A One-Way Delay Metric for IP Performance
     _________Metrics (IPPM)";
}

identity two-way-delay-maximum {
  base service-slo-metric-type;
  description
    "SLO two-way delay is the upper bound of network delay when
     transmitting between two SDPs";
  reference
    "RFC_2681: A Round-trip Delay Metric for IPPM";
}
```

Commenté [BMI60]: Consider adding a ref stmt

Commenté [BMI61]: Consider adding a ref stmt

```
identity two-way-delay-percentile {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is the percentile
     objective of network delay when the traffic transmitting
     between two SDPs.";
  reference
    "RFC_2681: A Round-trip Delay Metric for IPPM";
}

identity one-way-delay-variation-maximum {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is maximum bound of the
     difference in the one-way delay between sequential packets
     between two SDPs.";
  reference
    "RFC_3393: IP Packet Delay Variation Metric for IP Performance
     ___________ Metrics (IPPM)";
}

identity one-way-delay-variation-percentile {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is the percentile objective
     in the one-way delay between sequential packets between two
     SDPs.";
  reference
    "RFC_3393: IP Packet Delay Variation Metric for IP Performance
     __________Metrics (IPPM)";
}

identity two-way-delay-variation-maximum {
  base service-slo-metric-type;
  description
    "SLO two-way delay variation is the difference in the
     round-trip delay between sequential packets between two SDPs.";
  reference
    "RFC_5481: Packet Delay Variation Applicability Statement";
}

identity two-way-delay-variation-percentile {
  base service-slo-metric-type;
  description
    "The SLO objective of this metric is the percentile objective
     in the round-trip delay between sequential packets between
     two SDPs.";
  reference
    "RFC_5481: Packet Delay Variation Applicability Statement";
}

identity one-way-packet-loss {
  base service-slo-metric-type;
  description
    "This metric type refers to the ratio of packets dropped
     to packets transmitted between two SDPs in one-way
     over a period of time.";
  reference
```

Commenté [BMI62]: Does the model allow to control/report the "period of time" for computing the stats?

```
      "RFC7680: A One-Way Loss Metric for IP Performance
       Metrics (IPPM)";
  }

  identity two-way-packet-loss {
    base service-slo-metric-type;
    description
      "This metric type refers to the ratio of packets dropped
       to packets transmitted between two SDPs in two-way
       over a period of time.";
    reference
      "RFC7680: A One-Way Loss Metric for IP Performance
       Metrics (IPPM)";
  }

  /*
   * Identity for availability-type
   */

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

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

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

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

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

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

  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
            "Uses the physical interface as match criteria for
             Slice Service traffic.";
        }

        identity service-vlan-match {
          base service-match-type;
          description
            "Uses the VLAN ID as match criteria for the Slice Service
             traffic.";
        }

        identity service-label-match {
          base service-match-type;
          description
            "Uses the MPLS label as match criteria for the Slice Service
             traffic.";
        }

        identity service-source-ip-prefix-match {
          base service-match-type;
          description
            "Uses source ip IP prefix as match criteria for the Slice
    Service
             traffic. Examples of 'value' of this match type are
             '192.0.2.0/24' and '2001:db8::1/64'.";
        }

        identity service-destination-ip-prefix-match {
          base service-match-type;
          description
            "Uses destination ip IP prefix as match criteria for the Slice
             Service traffic. Examples of 'value' of this match type are
             '203.0.113.1/32', '2001:db8::2/128'.";
        }

        identity service-dscp-match {
          base service-match-type;
          description
            "Uses DSCP field in the IP packet header as match criteria
             for the Slice Service traffic.";
        }

        identity service-acl-match {
          base service-match-type;
          description
            "Uses Access Control List (ACL) as match criteria
             for the Slice Service traffic.";
          reference
            "RFC 8519: YANG Data Model for
             ___________Network Access Control Lists (ACLs)";
        }
```

```
identity service-any-match {
  base service-match-type;
  description
    "Matches any Slice Service traffic.";
}

identity slo-sle-policy-override {
  description
    "Base identity for SLO/SLE policy override options.";
}

identity slo-sle-policy-full-override {
  base slo-sle-policy-override;
  description
    "The policy of SLO/SLE(s) that is are defined at a
     child level override a parent SLO/SLE policy,
     which means that no SLO/SLE(s) are inherited from parent
     if a child SLO/SLE policy exists.";
}
identity slo-sle-policy-partial-override {
  base slo-sle-policy-override;
  description
    "The policy of SLO/SLE(s) that is are defined at a
     child level updates the parent SLO/SLE policy.
     For example, if a specific SLO is defined
     at the child level, that specific SLO overrides the
     one inherited from a parent SLO/SLE policy, while all other
     SLOs in the parent SLO-SLE policy still apply.";
}

/* Typedef */

typedef percentile {
  type decimal64 {
    fraction-digits 3;
    range "0..100";
  }
  description
    "The percentile is a value between 0 and 100
     to 3 decimal places, e.g., 10.000, 99.900 ,99.990, etc.
     For example, for a given one-way delay measurement,
     if the percentile is set to 95.000 and the 95th percentile
     one-way delay is 2 milliseconds, then the 95 percent of
     the sample value is less than or equal to 2 milliseconds.";
}

/* Groupings */

grouping service-slos {
  description
    "Directly measurable objectives of a Slice Service.";
  container slo-policy {
    description
      "Contains the SLO policy.";
    list metric-bound {
      key "metric-type";
      description
        "List of Slice Service metric bounds.";
```

```
          leaf metric-type {
            type identityref {
              base service-slo-metric-type;
            }
            description
              "Identifies an entry in the list of metric type
               bounds for the Slice Service.";
          }
          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 the provided value.";
          }
          leaf percentile-value {
            type percentile;
            description
              "The percentile value of the metric type.";
          }
          leaf bound {
            type uint64;
            default "0";
            description
              "The bound on the Slice Service connection metric.
               When set to zero, this indicates an unbounded
               upper limit for the specific metric-type.";
          }
        }
        leaf availability {
          type identityref {
            base availability-type;
          }
          description
            "Service availability level";
        }
        leaf mtu {
          type uint16;
          units "bytes";
          description
            "The MTU specifies the maximum length of data
             packets of the Slice Service.
             The value needs to be less than or equal to the
             minimum MTU value of all 'attachment-circuits'
             in the SDPs.";
        }
      }
    }

    grouping service-sles {
      description
        "Indirectly measurable objectives of a Slice Service.";
      container sle-policy {
```

Commenté [BMI63]: Why not defining those as identyrefs?

Commenté [BMI64]: What if the unit does not comply with the metric type? Ex. delay + unit=byte or mbps?

Commenté [BMI65]: Do you really need this. Please note that FC8407 says:

"The following guidelines apply to reusable groupings, in order to make them as robust as possible: ...
Do not include a "default" substatement on a leaf or choice unless the value applies on all possible contexts."

Commenté [BMI66]: Please note that RFC9291 uses the following type:

+--rw svc-mtu? uint32

I suggest to align what that type.

```
    description
      "Contains the SLE policy.";
    leaf-list security {
      type identityref {
        base service-security-type;
      }
      description
        "The security functions that the customer requests
         the operator to apply to traffic between the two SDPs.";
    }
    leaf-list isolation {
      type identityref {
        base service-isolation-type;
      }
      description
        "The Slice Service isolation requirement.";
    }
    leaf max-occupancy-level {
      type uint8 {
        range "1..100";
      }
      description
        "The maximal occupancy level specifies the number of flows
         to be admitted.";
    }
    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 sdp-peering {
  description
    "A grouping for the Slice Service SDP peering.";
  container sdp-peering {
    description
      "Describes SDP peering attributes.";
    leaf peer-sap-id {
      type string;
      description
        "Indicates a reference to the remote endpoints of an
         attachment circuit. This information can be used for
         correlation purposes, such as identifying a service
         attachment point (SAP) of a provider equipment when
         requesting a service with CE based SDP attributes.";
      reference
```

Commenté [BMI67]: Where this grouping is reused?

Please check that all groupings are reused.

Commenté [BMI68]: One or multiple peer SAPs can terminates an AC. Why leaf-list is not used here?

```
            "RFC9408: A YANG Network Data Model for
             Service Attachment Points (SAPs)";
        }
        container protocols {
          description
            "Serves as an augmentation target.
             Protocols can be augmented into this container,
             e.g., BGP, or static routing.";
        }
      }
    }

    grouping sdp-attachment-circuits {
      description
        "Grouping for the SDP attachment circuit definition.";
      container attachment-circuits {
        description
          "List of attachment circuits.";
        list attachment-circuit {
          key "id";
          description
            "The IETF Network Slice Service SDP attachment circuit
             related parameters.";
          leaf id {
            type string;
            description
              "Uniquely identifies an attachment circuit with an NSC.";
          }
          leaf description {
            type string;
            description
              "The attachment circuit's description.";
          }
          leaf ac-svc-name {
            type string;
            description
              "Indicates an attachment circuit (AC) service name,
               for association purposes, to refer to an AC that has been
               created before the slice creation.
               This node can override 'ac-svc-name' of the parent SDP.";
          }
          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-ipv4-address {
            type inet:ipv4-address;
            description
              "The IPv4 address of the AC.";
          }
          leaf ac-ipv4-prefix-length {
```

a mis en forme : Surlignage

```
            type uint8;
            description
              "The IPv4 subnet prefix length expressed in bits.";
          }
          leaf ac-ipv6-address {
            type inet:ipv6-address;
            description
              "The IPv6 address of the AC.";
          }
          leaf ac-ipv6-prefix-length {
            type uint8;
            description
              "The IPv6 subnet prefix length expressed in bits.";
          }
          leaf mtu {
            type uint16;
            units "bytes";
            description
              "Maximum size of the Slice Service data packet
               that can traverse an SDP.";
          }
          container ac-tags {
            description
              "Container for the attachment circuit tags.";
            list ac-tags {
              key "tag-type";
              description
                "The attachment circuit tags list.";
              leaf 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 values. For example, the
                   tag may indicate 'c-vlan' and 's-vlan'.";
              }
            }
          }
          uses service-qos;
          uses sdp-peering;
          uses vpn-common:service-status;
        }
      }
    }

    grouping sdp-monitoring-metrics {
      description
        "Grouping for the SDP monitoring metrics.";
      container sdp-monitoring {
        config false;
        description
          "Container for SDP monitoring metrics.";
        leaf incoming-bw-value {
```

Commenté [BMI69]: Hmm. Please ajust the type to be applicable to IPv4

Commenté [BMI70]: Hmm. Please ajust the type to be applicable to IPv6

Commenté [BMI71]: See the comment about the MTU type above

Commenté [BMI72]: Please refer to 8407bis for naming convention guidance.

Commenté [BMI73]: Update the examples accordingly

Commenté [BMI74]: How to know which vlan tag is provided?

Commenté [BMI75]: Please note that 8407 says:

Do not include a "config" substatement on a data node unless the value applies on all possible contexts.

```
      type uint64;
      units "bps";
      description
        "Indicates the absolute value of the incoming bandwidth
         at an SDP from the customer network or
         from another provider's network.";
    }
    leaf incoming-bw-percent {
      type decimal64 {
        fraction-digits 5;
        range "0..100";
      }
      units "percent";
      mandatory true;
      description
        "Indicates a percentage of the incoming bandwidth
         at an SDP from the customer network or
         from another provider's network.";
    }
    leaf outgoing-bw-value {
      type uint64;
      units "bps";
      description
        "Indicates the absolute value of the outgoing bandwidth
         at an SDP towards the customer network or towards
         another provider's network.";
    }
    leaf outgoing-bw-percent {
      type decimal64 {
        fraction-digits 5;
        range "0..100";
      }
      units "percent";
      mandatory true;
      description
        "Indicates a percentage of the outgoing bandwidth
         at an SDP towards the customer network or towards
         another provider's network.";
    }
  }
}

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 {
  description
    "A grouping containing a GPS location.";
  container location {
    description
      "A container containing a GPS location.";
    leaf altitude {
      type int64;
      units "millimeter";
```

Commenté [BMI76]: Refer to the comment about existing location grouping

```
      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.";
    }
  }
}

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-qos {
        description
          "The rate limits grouping.";
        container incoming-qos-policy {
          description
            "The QoS policy imposed on ingress direction of the traffic ,
             from the customer network or from another provider's
network.";
          leaf 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.";
          }
          container rate-limits {
            description
              "Container for the asymmetric traffic control.";
            uses bw-rate-limits;
          }
        }
        container outgoing-qos-policy {
          description
            "The QoS policy imposed on egress direction of the traffic ,
             towards the customer network or towards another
             provider's network.";
          leaf 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.";
          }
          container rate-limits {
            description
              "The rate-limit imposed on outgoing traffic.";
            uses bw-rate-limits;
          }
        }
      }
```

Commenté [BMI77]: This should indexed per type (per-AC, per-class, etc.)

Commenté [BMI78]: Idem as above

```
    grouping sdp {
      description
        "Slice Service SDP related information";
      leaf id {
        type string;
        description
          "Unique identifier for the referred Slice Service SDP.";
      }
      leaf description {
        type string;
        description
          "Provides a description of the SDP.";
      }
      uses geolocation;
      leaf node-id {
        type string;
        description
          "Uniquely identifies an edge node of the SDP.";
      }
      leaf-list sdp-ip-address {
        type inet:ip-address;
        description
          "IPv4 or IPv6 address of the SDP.";
      }
      leaf tp-ref {
        type leafref {
          path
            "/nw:networks/nw:network[nw:network-id =current()/../../"
          + "../custom-topology/network-ref]/"
          + "nw:node/nt:termination-point/nt:tp-id";
        }
        description
          "A reference to Termination Point (TP) in the custom
           topology";
        reference
          "RFC 8345: A YANG Data Model for Network Topologies";
      }
      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 identifier that uniquely identifies a match
criteria.";
          }
          leaf match-type {
            type identityref {
              base service-match-type;
            }
            mandatory true;
            description
              "Indicates the match type of the entry in the list of
```

Commenté [BMI79]: What is the unicity scope?

```
                the Slice Service match criteria.";
            }
            leaf-list value {
              type string;
              description
                "Provides a value for the Slice Service match criteria,
                 e.g., IP prefix and VLAN ID.";
            }
            leaf target-connection-group-id {
              type leafref {
                path "../../../../../ietf-nss:connection-groups"
                   + "/ietf-nss:connection-group"
                   + "/ietf-nss: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
                "Specifies the role of SDP in the connection group
                 When the service connection type is MP2MP,
                 such as hub and spoke service connection type. In
addition,
                 this helps to create connectivity construct automatically
                 , rather than explicitly specifying each one.";
            }
            leaf target-connectivity-construct-id {
              type leafref {
                path "/ietf-nss:network-slice-services"
                   + "/ietf-nss:slice-service"
                   + "/ietf-nss:connection-groups"
                   + "/ietf-nss:connection-group[id"
                   + "=current()/../target-connection-group-id]"
                   + "/ietf-nss:connectivity-construct/ietf-nss:id";
              }
              description
                "Reference to a Network Slice connection construct.";
            }
          }
        }
        uses service-qos;
        container sdp-peering {
          description
            "Describes SDP peering attributes.";
          leaf-list peer-sap-id {
            type string;
            description
              "Indicates the reference to the remote endpoints of the
               attachment circuits. This information can be used for
               correlation purposes, such as identifying service
               attachment points (SAPs) of provider equipmentsequipment
when
               requesting a service with CE based SDP attributes.";
```

Commenté [BMI80]: This should be defined as a typedef to ease referencing

```
      }
      container protocols {
        description
          "Serves as an augmentation target.
           Protocols can be augmented into this container,
           e.g., BGP, static routing.";
      }
    }
    leaf-list ac-svc-name {
      type string;
      description
        "Indicates the attachment circuit service names,
         for association purposes, to refer to ACs that have been
         created before the slice creation.";
    }
    uses sdp-attachment-circuits;
    uses vpn-common:service-status;
    uses sdp-monitoring-metrics;
  }
  grouping connectivity-construct {
    description
      "Grouping for Slice Service connectivity construct.";
    list connectivity-construct {
      key "id";
      description
        "List of connectivity constructs.";
      leaf id {
        type uint32;
        description
          "The connectivity construct identifier.";
      }
      choice type {
        default "p2p";
        description
          "Choice for connectivity construct type.";
        case p2p {
          description
            "P2P connectivity construct.";
          leaf p2p-sender-sdp {
            type leafref {
              path "../../../../sdps/sdp/id";
            }
            description
              "Reference to a sender SDP.";
          }
          leaf p2p-receiver-sdp {
            type leafref {
              path "../../../../sdps/sdp/id";
            }
            description
              "Reference to a receiver SDP.";
          }
        }
        case p2mp {
          description
            "P2MP connectivity construct.";
          leaf p2mp-sender-sdp {
            type leafref {
```

Commenté [BMI81]: You may add a reference stmt to the ACaaS draft.

Commenté [BMI82]: To be defined as typedef to ease referencing

Commenté [BMI83]: To be defined as typedef to ease referencing

```
                path "../../../../sdps/sdp/id";
              }
              description
                "Reference to a sender SDP.";
            }
            leaf-list p2mp-receiver-sdp {
              type leafref {
                path "../../../../sdps/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/id";
                }
                description
                  "Reference to an 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.
         */
        uses service-slo-sle-policy-override;
        uses vpn-common:service-status;
        container connectivity-construct-monitoring {
          config false;
          description
            "SLO status per connectivity construct.";
          uses connectivity-construct-monitoring-metrics;
        }
      }
    }

    grouping connection-group {
      description
        "Grouping for Slice Service connection group.";
      leaf id {
        type string;
        description
          "The connection group identifier.";
      }
      leaf connectivity-type {
        type identityref {
          base vpn-common:vpn-topology;
        }
```

a mis en forme : Surlignage

a mis en forme : Surlignage

a mis en forme : Surlignage

```
    default "vpn-common:any-to-any";
    description
      "Connection group connectivity type.";
  }
  uses service-slo-sle-policy;
  uses service-slo-sle-policy-override;
  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;
  }
}

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";
      description
        "List for SLO and SLE template identifiers.";
      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 description {
        type string;
        description
          "Describes the SLO and SLE policy template.";
      }
      leaf template-ref {
        type leafref {
          path "/ietf-nss:network-slice-services"
             + "/ietf-nss:slo-sle-templates"
             + "/ietf-nss:slo-sle-template"
             + "/ietf-nss:id";
        }
        description
          "The reference to a standard template. When set it
            indicates the base template over which further
            SLO/SLE policy changes are made.";
      }
      uses service-slos;
      uses service-sles;
    }
  }
}
```

Commenté [BMI84]: As typedef.

a mis en forme : Surlignage

a mis en forme : Surlignage

```
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"
             + "/ietf-nss:slo-sle-template"
             + "/ietf-nss:id";
        }
        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 description {
          type string;
          description
            "Describes the SLO and SLE policy.";
        }
        uses service-slos;
        uses service-sles;
      }
    }
  }
}

grouping service-slo-sle-policy-override {
  description
    "Slice Service policy override grouping.";
  leaf service-slo-sle-policy-override {
    type identityref {
      base slo-sle-policy-override;
    }
    default "ietf-nss:slo-sle-policy-full-override";
    description
      "SLO/SLE policy override option.";
  }
}

/* Main IETF Network Slice Services Container */

container network-slice-services {
  description
    "Contains a list of IETF Network Slice Services";
  uses slice-service-template;
```

Commenté [BMI85]: typedef

Commenté [BMI86]: Please refer to 8407 for defaults in groupings.

```
list slice-service {
  key "id";
  description
    "A Slice Service is identified by a service id.";
  leaf id {
    type string;
    description
      "A unique Slice Service identifier.";
  }
  leaf 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
        "The service tag list.";
      leaf tag-type {
        type identityref {
          base service-tag-type;
        }
        description
          "Slice service tag type.";
      }
      leaf-list value {
        type string;
        description
          "The tag values, e.g., customer names when multiple
           customers sharing same Slice Service in 5G scenario.";
      }
    }
  }
  uses service-slo-sle-policy;
  leaf compute-only {
    type empty;
    description
      "When present, the slice is computed. No resources are
       committed or reserved in the network.";
  }
  uses vpn-common:service-status;
  container sdps {
    description
      "Slice Service SDPs.";
    list sdp {
      key "id";
      min-elements 2;
      uses sdp;
      description
        "List of SDPs in this Slice Service.";
    }
  }
  container connection-groups {
    description
      "Contains connection groups.";
```

Commenté [BMI87]: Which scope?

```
          list connection-group {
            key "id";
            description
              "List of connection groups.";
            uses connection-group;
          }
        }
        container custom-topology {
          description
            "Serves as an augmentation target.
             Container for custom topology, which is indicated by the
             referenced topology predefined, e.g., an abstract RFC8345
             topology.";
          uses nw:network-ref;
        }
      }
    }
  }
  <CODE ENDS>
```

7. Security Considerations

Commenté [BMI88]: Please make sure the template in https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-01#section-3.7.1 is used.

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

Commenté [BMI89]: What about the templates?

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 request to register~~s~~ ~~a~~ the following URI in the IETF XML registry [RFC3688]~~.~~

~~Following the format in [RFC3688], the following registration is requested to be made:~~:

```
URI: urn:ietf:params:xml:ns:yang:ietf-network-slice-service
Registrant Contact: The IESG.
```

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

This document requests to register ~~a~~ the following YANG module in the YANG Module
Names registry [RFC7950].

Name: ietf-network-slice-service
Namespace: urn:ietf:params:xml:ns:yang:ietf-network-slice-service
Prefix: ietf-nss
Maintained by IANA: N
Reference: RFC XXXX

9. Acknowledgments

The authors wish to thank Mohamed Boucadair, Kenichi Ogaki, Sergio Belotti, Qin Wu, Yao Zhao, Susan Hares, Eric Grey, Daniele Ceccarelli, Ryan Hoffman, Adrian Farrel, Aihua Guo, Italo Busi, and many others for their helpful comments and suggestions.

Thanks to Ladislav Lhotka for the YANG Doctors review.

10. Contributors

The following authors contributed significantly to this document:

Luis M. Contreras
Telefonica
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

Liuyan Han
China Mobile
Email: hanliuyan@chinamobile.com

11. References

11.1. Normative References

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

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

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

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

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types",

RFC 6991, DOI 10.17487/RFC6991, July 2013,
<https://www.rfc-editor.org/info/rfc6991>.

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

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

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

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

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

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

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

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

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

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

[RFC9181] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., and Q. Wu, "A Common YANG Data Model for Layer 2 and Layer 3 VPNs", RFC 9181, DOI 10.17487/RFC9181, February 2022, <https://www.rfc-editor.org/info/rfc9181>.

11.2. Informative References

[I-D.boro-opsawg-teas-attachment-circuit]
Boucadair, M., Roberts, R., de Dios, O. G., Barguil, S., and B. Wu, "YANG Data Models for 'Attachment Circuits'-as-a-Service (ACaaS)", Work in Progress, Internet-Draft, draft-boro-opsawg-teas-attachment-circuit-07, 10 July 2023, <https://datatracker.ietf.org/doc/html/draft-boro-opsawg-teas-attachment-circuit-07>.

[I-D.boro-opsawg-teas-common-ac]
Boucadair, M., Roberts, R., de Dios, O. G., Barguil, S., and B. Wu, "A Common YANG Data Model for Attachment Circuits", Work in Progress, Internet-Draft, draft-boro-opsawg-teas-common-ac-02, 3 May 2023, <https://datatracker.ietf.org/doc/html/draft-boro-opsawg-teas-common-ac-02>.

[I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for Virtual Network (VN) Operations", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-20, 14 October 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-actn-vn-yang-20>.

[I-D.ietf-teas-ietf-network-slices]
Farrel, A., Drake, J., Rokui, R., Homma, S., Makhijani, K., Contreras, L. M., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slices-25, 14 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slices-25>.

[RFC7640] Constantine, B. and R. Krishnan, "Traffic Management Benchmarking", RFC 7640, DOI 10.17487/RFC7640, September 2015, <https://www.rfc-editor.org/info/rfc7640>.

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

[RFC9408] Boucadair, M., Ed., Gonzalez de Dios, O., Barguil, S., Wu, Q., and V. Lopez, "A YANG Network Data Model for Service Attachment Points (SAPs)", RFC 9408, DOI 10.17487/RFC9408, June 2023, <https://www.rfc-editor.org/info/rfc9408>.

Commenté [BMI90]: This can be listed as normative

Commenté [BMI91]: This one also is to be listed as normative.

Appendix A. Augmentation Considerations

The NSSM defines the minimum attributes of slice services. In some scenarios, further extension, e.g., the definition of AC technology specific attributes and the "isolation" SLE characteristics are required.

For AC technology specific attributes, if the customer and provider need to agree, through configuration, on the technology parameter values, such as the protocol types and protocol parameters between

the PE and the CE. The following shows an example where BGP and static routing are augmented to the Network Slice Service model. The protocol types and definitions can reference [I-D.boro-opsawg-teas-common-ac].

```
augment /ietf-nss:network-slice-services/ietf-nss:slice-service/ietf-
nss:sdps\
  /ietf-nss:sdp/ietf-nss:sdp-peering/ietf-nss:protocols:
    +--rw bgp-attributes
    |  +--rw description?   string
    |  +--rw peer-as?       inet:as-number
    |  +--rw neighbor*      inet:ip-address
    +--rw static-attributes
       +--rw cascaded-lan-prefixes
          +--rw ip-lan-prefixes* [lan next-hop]
             +--rw lan         inet:ip-prefix
             +--rw next-hop    union
                   ...
```

Commenté [BMI92]: Please use use "--yang-line-length 69" to generate the diagram.

In some scenarios, for example, when multiple Sslice ~~services~~ Services share one or more ACs, independent AC services, defined in [I-D.boro-opsawg-teas-attachment-circuit], can be used.

For "isolation" SLE characteristics, the following identities can be defined.

```
  identity service-interference-isolation-dedicated {
    base service-isolation-type;
    description
      "Specify the requirement that the slice service is not impacted
   by the existence of other customers or services in the same
   network, which may be provided by the service provider using
   ~~dedicatd~~dedicated network resources, similar to a dedicated private
network.";
  }
```

Appendix B. Examples of Network Slice Services

B.1. Example-1: Two A2A Slice Services with different match approaches

The following example describes a simplified service configuration of two IETF Network ~~slice~~ Slice Service instances where the SDPs are the customer-facing ports on the PE:

Commenté [BMI93]: Indicate that the json example is about the body of the request.

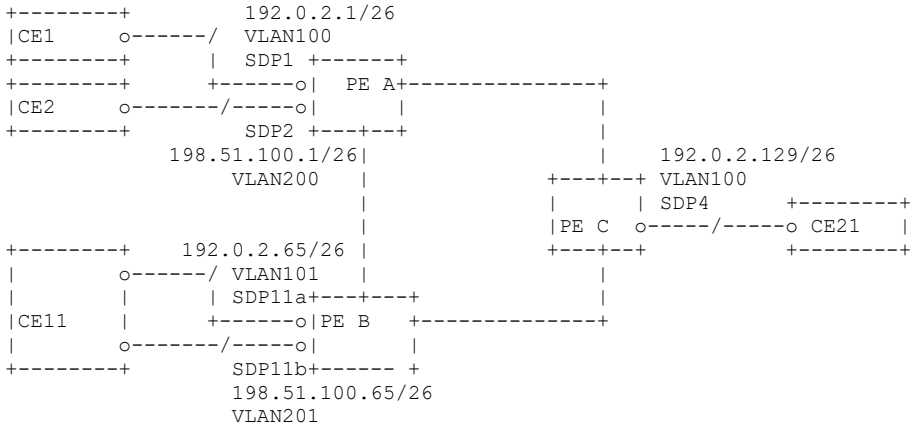
* IETF Network Slice 1 on SDP1, SDP11a, and SDP4, with an A2A connectivity type. This is a L3 slice service and using the uniform low latency "slo-sle-template" policy between all SDPs. These SDPs will also have AC eBGP peering sessions with unmanaged CE elements (not shown) using an AC augmentation model such as the one shown above.

* IETF Network Slice 2 on SDP2, SDP11b, with A2A connectivity type. This is a L3 slice service and using the uniform high bandwidth "slo-sle-template" policy between all SDPs.

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. In 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",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice1",
          "description": "example slice1",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          },
          "slo-sle-template": "low-latency-template",
          "status": {
```

Commenté [BMI94]: You may say that the template are known to the customer.

```
},
"sdps": {
  "sdp": [
    {
      "id": "1",
      "node-id": "PE-A",
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:service-any-match",
            "target-connection-group-id": "matrix1",
            "target-connectivity-construct-id": 1
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac1",
            "description": "AC1 connected to device 1",
            "ac-node-id": "PE-A",
            "ac-tp-id": "GigabitEthernet5/0/0/0.100",
            "ac-ipv4-address": "192.0.2.1",
            "ac-ipv4-prefix-length": 26,
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["100"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "3a",
      "node-id": "PE-B",
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:service-any-match",
            "target-connection-group-id": "matrix1",
            "target-connectivity-construct-id": 1
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac3a",
```

```
        "description": "AC3a connected to device 3",
        "ac-node-id": "PE-B",
        "ac-tp-id": "GigabitEthernet8/0/0/4.101",
        "ac-ipv4-address": "192.0.2.65",
        "ac-ipv4-prefix-length": 26,
        "ac-tags": {
          "ac-tags": [
            {
              "tag-type": "ietf-nss:vlan-id",
              "value": ["101"]
            }
          ]
        },
        "status": {
        }
      }
    ]
  },
  "status": {
  }
},
{
  "id": "4",
  "node-id": "PE-C",
  "service-match-criteria": {
    "match-criterion": [
      {
        "index": 1,
        "match-type": "ietf-nss:service-any-match",
        "target-connection-group-id": "matrix1",
        "target-connectivity-construct-id": 1
      }
    ]
  },
  "attachment-circuits": {
    "attachment-circuit": [
      {
        "id": "ac4",
        "description": "AC4 connected to device 4",
        "ac-node-id": "PE-C",
        "ac-tp-id": "GigabitEthernet4/0/0/3.100",
        "ac-ipv4-address": "192.0.2.129",
        "ac-ipv4-prefix-length": 26,
        "ac-tags": {
          "ac-tags": [
            {
              "tag-type": "ietf-nss:vlan-id",
              "value": ["100"]
            }
          ]
        },
        "status": {
        }
      }
    ]
  },
  "status": {
  }
```

```
      }
    ]
  },
  "connection-groups": {
    "connection-group": [
      {
        "id": "matrix1",
        "connectivity-type": "ietf-vpn-common:any-to-any",
        "connectivity-construct": [
          {
            "id": 1,
            "a2a-sdp": [
              {
                "sdp-id": "1"
              },
              {
                "sdp-id": "3a"
              },
              {
                "sdp-id": "4"
              }
            ],
            "status": {
            }
          }
        ]
      }
    ]
  }
},
{
  "id": "slice2",
  "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": [
      {
        "id": "2",
        "node-id": "PE-A",
        "attachment-circuits": {
          "attachment-circuit": [
            {
              "id": "ac2",
              "description": "AC2 connected to device 2",
              "ac-node-id": "PE-A",
              "ac-tp-id": "GigabitEthernet7/0/0/3.200",
              "ac-ipv4-address": "198.51.100.1",
              "ac-ipv4-prefix-length": 26,
```

```
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["100"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "3b",
      "node-id": "PE-B",
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac3b",
            "description": "AC3b connected to device 3",
            "ac-node-id": "PE-B",
            "ac-tp-id": "GigabitEthernet8/0/0/4.201",
            "ac-ipv4-address": "198.51.100.65",
            "ac-ipv4-prefix-length": 26,
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["201"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    }
  ]
},
"connection-groups": {
  "connection-group": [
    {
      "id": "matrix2",
      "connectivity-type": "ietf-vpn-common:any-to-any",
      "connectivity-construct": [
        {
          "id": 1,
          "a2a-sdp": [
            {
              "sdp-id": "2"
            },
```

```
                    {
                      "sdp-id": "3b"
                    }
                  ],
                  "status": {
                  }
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

B.2. Example-2: Two P2P slice services with different match approaches

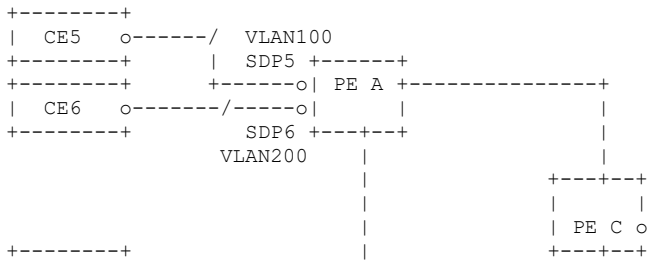
The following example describes a simplified service configuration of two IETF Network slice instances where the SDPs are the customer-facing ports on the PE:

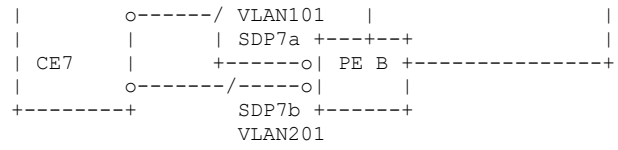
* IETF Network Slice 3 on SDP5 and SDP7a with P2P connectivity type. This is a L2 slice service and using the uniform low-latency "slo-sle-template" policies between the SDPs. A connectivity-group level slo-policy has been applied with a ~~delay~~ delay-based metric bound of 10ms which will apply to both connectivity-constructs.

* IETF Network Slice 4 on SDP6 and SDP7b, with P2P connectivity type. This is a L2 slice service and using the the high bandwidth "slo-sle-template" policies between the SDPs. Traffic from SDP6 and SDP7b is requesting a bandwidth 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",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice3",
          "description": "example slice3",
          "slo-sle-template": "low-latency-template",
          "status": {
          },
          "sdps": {
            "sdp": [
              {
                "id": "5",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-nss:service-any-match",
                      "target-connection-group-id": "matrix3"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "id": "ac5",
                      "description": "AC5 connected to device 5",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet5/0/0/1",
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "tag-type": "ietf-nss:vlan-id",
                            "value": ["100"]
                          }
                        ]
                      },
```

```
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "7a",
      "node-id": "PE-B",
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:service-any-match",
            "target-connection-group-id": "matrix3"
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac7a",
            "description": "AC7a connected to device 7",
            "ac-node-id": "PE-B",
            "ac-tp-id": "GigabitEthernet8/0/0/5",
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["200"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    }
  ]
},
"connection-groups": {
  "connection-group": [
    {
      "id": "matrix3",
      "connectivity-type": "ietf-nss:point-to-point",
      "service-slo-sle-policy": {
        "slo-policy": {
          "metric-bound": [
            {
              "metric-type": "ietf-nss:one-way-delay-maximum",
              "metric-unit": "milliseconds",
              "bound": "10"
            }
```

```
            ]
          }
        },
        "connectivity-construct": [
          {
            "id": 1,
            "p2p-sender-sdp": "5",
            "p2p-receiver-sdp": "7a",
            "status": {
            }
          },
          {
            "id": 2,
            "p2p-sender-sdp": "7a",
            "p2p-receiver-sdp": "5",
            "status": {
            }
          }
        ]
      }
    ]
  }
},
{
  "id": "slice4",
  "description": "example slice4",
  "slo-sle-template": "high-BW-template",
  "status": {
  },
  "sdps": {
    "sdp": [
      {
        "id": "6",
        "node-id": "PE-A",
        "attachment-circuits": {
          "attachment-circuit": [
            {
              "id": "ac6",
              "description": "AC6 connected to device 6",
              "ac-node-id": "PE-A",
              "ac-tp-id": "GigabitEthernet7/0/0/4",
              "ac-tags": {
                "ac-tags": [
                  {
                    "tag-type": "ietf-nss:vlan-id",
                    "value": ["101"]
                  }
                ]
              },
              "status": {
              }
            }
          ]
        },
        "status": {
        }
      },
      {
```

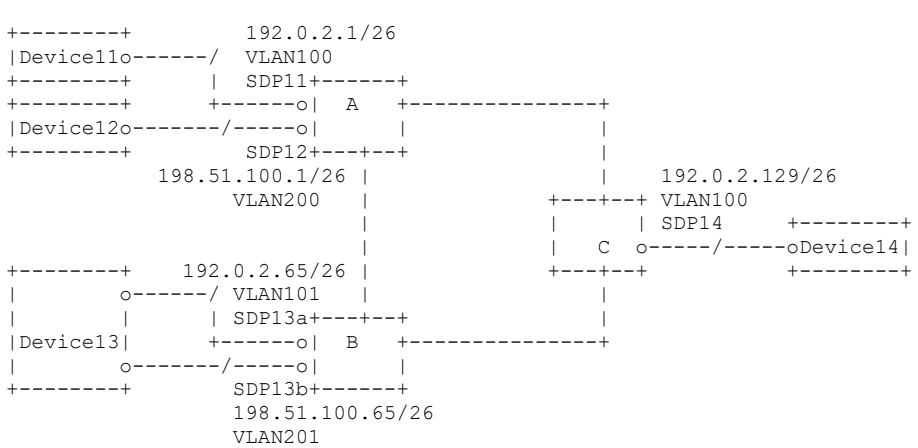
```
      "id": "7b",
      "node-id": "PE-B",
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac7b",
            "description": "AC7b connected to device 7",
            "ac-node-id": "PE-B",
            "ac-tp-id": "GigabitEthernet8/0/0/5",
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["201"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    }
  ]
},
"connection-groups": {
  "connection-group": [
    {
      "id": "matrix4",
      "connectivity-type": "ietf-nss:point-to-point",
      "connectivity-construct": [
        {
          "id": 1,
          "p2p-sender-sdp": "6",
          "p2p-receiver-sdp": "7b",
          "service-slo-sle-policy": {
            "slo-policy": {
              "metric-bound": [
                {
                  "metric-type": "ietf-nss:one-way-bandwidth",
                  "metric-unit": "Mbps",
                  "bound": "1000"
                }
              ]
            }
          },
          "status": {
          }
        },
        {
          "id": 2,
          "p2p-sender-sdp": "7b",
          "p2p-receiver-sdp": "6",
          "service-slo-sle-policy": {
            "slo-policy": {
              "metric-bound": [
```

```
                        {
                          "metric-type": ietf-nss:one-way-bandwidth",
                          "metric-unit": "Mbps",
                          "bound": "5000"
                         }
                       ]
                     }
                   },
                   "status": {
                   }
                 }
               ]
             }
           ]
         }
       }
     ]
   }
 }
}
```

B.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 SDPs, but using an explicit set of SLO policies with a latency metric of 10ms for hub to spoke traffic.




```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
```

```
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice5",
          "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": [
              {
                "id": "11",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-nss:service-any-match",
                      "target-connection-group-id": "matrix5",
                      "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "id": "ac11",
                      "description": "AC11 connected to device 11",
                      "ac-node-id": "PE-A",
                      "ac-tp-id": "GigabitEthernet5/0/0/2",
                      "ac-ipv4-address": "192.0.2.1",
                      "ac-ipv4-prefix-length": 26,
                      "ac-tags": {
                        "ac-tags": [
                          {
                            "tag-type": "ietf-nss:vlan-id",
                            "value": ["100"]
                          }
                        ]
```

```
                    },
                    "status": {
                    }
                  }
                ]
              },
              "status": {
              }
            },
            {
              "id": "12",
              "node-id": "PE-A",
              "service-match-criteria": {
                "match-criterion": [
                  {
                    "index": 1,
                    "match-type": "ietf-nss:service-any-match",
                    "target-connection-group-id": "matrix5",
                    "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                  }
                ]
              },
              "attachment-circuits": {
                "attachment-circuit": [
                  {
                    "id": "ac12",
                    "description": "AC12 connected to device 12",
                    "ac-node-id": "PE-A",
                    "ac-tp-id": "GigabitEthernet7/0/0/5",
                    "ac-ipv4-address": "198.51.100.1",
                    "ac-ipv4-prefix-length": 26,
                    "ac-tags": {
                      "ac-tags": [
                        {
                          "tag-type": "ietf-nss:vlan-id",
                          "value": ["200"]
                        }
                      ]
                    },
                    "status": {
                    }
                  }
                ]
              },
              "status": {
              }
            },
            {
              "id": "13a",
              "node-id": "PE-B",
              "service-match-criteria": {
                "match-criterion": [
                  {
                    "index": 1,
                    "match-type": "ietf-nss:service-any-match",
                    "target-connection-group-id": "matrix5",
```

```
                    "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                  }
                ]
              },
              "attachment-circuits": {
                "attachment-circuit": [
                  {
                    "id": "ac13a",
                    "description": "AC13a connected to device 13",
                    "ac-node-id": "PE-B",
                    "ac-tp-id": "GigabitEthernet8/0/0/6",
                    "ac-ipv4-address": "192.0.2.65",
                    "ac-ipv4-prefix-length": 26,
                    "ac-tags": {
                      "ac-tags": [
                        {
                          "tag-type": "ietf-nss:vlan-id",
                          "value": ["101"]
                        }
                      ]
                    },
                    "status": {
                    }
                  }
                ]
              },
              "status": {
              }
            },
            {
              "id": "13b",
              "node-id": "PE-B",
              "service-match-criteria": {
                "match-criterion": [
                  {
                    "index": 1,
                    "match-type": "ietf-nss:service-any-match",
                    "target-connection-group-id": "matrix5",
                    "connection-group-sdp-role": "ietf-vpn-
common:spoke-role"
                  }
                ]
              },
              "attachment-circuits": {
                "attachment-circuit": [
                  {
                    "id": "ac13b",
                    "description": "AC3b connected to device 13",
                    "ac-node-id": "PE-B",
                    "ac-tp-id": "GigabitEthernet8/0/0/4",
                    "ac-ipv4-address": "198.51.100.65",
                    "ac-ipv4-prefix-length": 26,
                    "ac-tags": {
                      "ac-tags": [
                        {
                          "tag-type": "ietf-nss:vlan-id",
                          "value": ["201"]
```

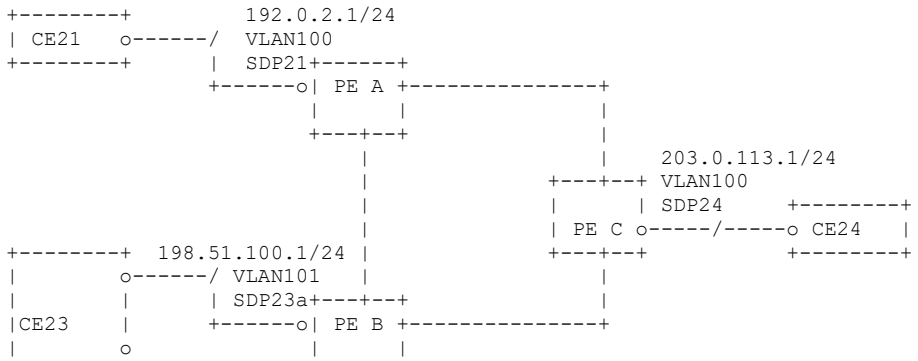
```
                        }
                      ]
                    },
                    "status": {
                    }
                  }
                ]
              },
              "status": {
              }
            },
            {
              "id": "14",
              "node-id": "PE-C",
              "service-match-criteria": {
                "match-criterion": [
                  {
                    "index": 1,
                    "match-type": "ietf-nss:service-any-match",
                    "target-connection-group-id": "matrix5",
                    "connection-group-sdp-role": "ietf-vpn-common:hub-
role"
                  }
                ]
              },
              "attachment-circuits": {
                "attachment-circuit": [
                  {
                    "id": "ac14",
                    "description": "AC14 connected to device 14",
                    "ac-node-id": "PE-C",
                    "ac-tp-id": "GigabitEthernet4/0/0/3",
                    "ac-ipv4-address": "192.0.2.129",
                    "ac-ipv4-prefix-length": 26,
                    "ac-tags": {
                      "ac-tags": [
                        {
                          "tag-type": "ietf-nss:vlan-id",
                          "value": ["100"]
                        }
                      ]
                    },
                    "status": {
                    }
                  }
                ]
              },
              "status": {
              }
            }
          ]
        },
        "connection-groups": {
          "connection-group": [
            {
              "id": "matrix5",
              "connectivity-type": "ietf-vpn-common:hub-spoke",
              "connectivity-construct": [
```

```
                  {
                    "id": 1,
                    "p2mp-sender-sdp": "14",
                    "p2mp-receiver-sdp": ["11", "12", "13a", "13b"],
                    "service-slo-sle-policy": {
                      "slo-policy": {
                        "metric-bound": [
                          {
                            "metric-type": "ietf-nss:one-way-delay-
maximum",
                            "metric-unit": "milliseconds",
                            "bound": "10"
                          }
                        ]
                      }
                    },
                    "status": {
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

B.4. Example-4: An A2A 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 A2A 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",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          },
          {
            "id": "standard-template",
            "description": "take the standard forwarding path"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice6",
          "description": "example slice6",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          },
          "slo-sle-template": "standard-template",
          "status": {
          },
          "sdps": {
            "sdp": [
              {
                "id": "21",
                "node-id": "PE-A",
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-nss:_service-dscp-match",
                      "value": ["EF"],
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 2
                    },
                    {
                      "index": 2,
                      "match-type": "ietf-nss:_service-any-match",
                      "target-connection-group-id": "matrix6",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
```

```
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac21",
            "description": "AC21 connected to device 21",
            "ac-node-id": "PE-A",
            "ac-tp-id": "GigabitEthernet5/0/0/0",
            "ac-ipv4-address": "192.0.2.1",
            "ac-ipv4-prefix-length": 24,
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["100"]
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "23a",
      "node-id": "PE-B",
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:_service-dscp-match",
            "value": ["EF"],
            "target-connection-group-id": "matrix6",
            "target-connectivity-construct-id": 2
          },
          {
            "index": 2,
            "match-type": "ietf-nss:_service-any-match",
            "target-connection-group-id": "matrix6",
            "target-connectivity-construct-id": 1
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac23a",
            "description": "AC23a connected to device 23",
            "ac-node-id": "PE-B",
            "ac-tp-id": "GigabitEthernet8/0/0/4",
            "ac-ipv4-address": "198.51.100.1",
            "ac-ipv4-prefix-length": 24,
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
```

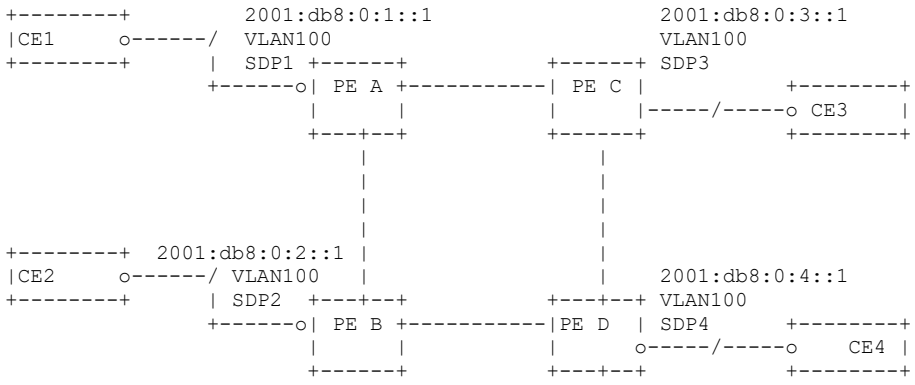
```
                "value": ["101"]
              }
            ]
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  },
  {
    "id": "24",
    "node-id": "PE-C",
    "service-match-criteria": {
      "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-nss:service-dscp-match",
          "value": ["EF"],
          "target-connection-group-id": "matrix6",
          "target-connectivity-construct-id": 2
        },
        {
          "index": 2,
          "match-type": "ietf-nss:_service-any-match",
          "target-connection-group-id": "matrix6",
          "target-connectivity-construct-id": 1
        }
      ]
    },
    "attachment-circuits": {
      "attachment-circuit": [
        {
          "id": "ac24",
          "description": "AC24 connected to device 24",
          "ac-node-id": "PE-C",
          "ac-tp-id": "GigabitEthernet4/0/0/3",
          "ac-ipv4-address": "203.0.113.1",
          "ac-ipv4-prefix-length": 24,
          "ac-tags": {
            "ac-tags": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": ["100"]
              }
            ]
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  }
]
```

```
          },
          "connection-groups": {
            "connection-group": [
              {
                "id": "matrix6",
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "21"
                      },
                      {
                        "sdp-id": "23a"
                      },
                      {
                        "sdp-id": "24",
                        "slo-sle-template": "low-latency-template"
                      }
                    ],
                    "status": {
                    }
                  },
                  {
                    "id": 2,
                    "a2a-sdp": [
                      {
                        "sdp-id": "21"
                      },
                      {
                        "sdp-id": "23a"
                      },
                      {
                        "sdp-id": "24"
                      }
                    ],
                    "status": {
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```


B.5. Example-5: An A2A Network Slice Service with SLO Precedence Policies

The following examples describes a simplified service configuration of an IETF Network slice instance "slice-7" with four SDPs: SDP1, SDP2, SDP3 and SDP4 with A2A connectivity type. All SDPs are designated as customer-facing ports on the PE.

The service is realized using a single A2A connectivity construct, and a low-bandwidth "slo-sle-template" policy applied to SDP4 and SDP3, while a high-bandwidth "slo-sle-template" policy applied to SDP1 and SDP2. Notice that the "slo-sle-templates" at the ~~connecitivty~~connectivity ~~—~~ construct level takes precedence ~~to~~ over the one specified at the group level.




```
   {
      "data": {
        "ietf-network-slice-service:network-slice-services": {
          "slo-sle-templates": {
            "slo-sle-template": [
              {
                "id": "high-BW-template",
                "description": "take the highest BW forwarding path"
              },
              {
                "id": "low-BW-template",
                "description": "lowest BW forwarding behavior"
              }
            ]
          },
          "slice-service": [
            {
              "id": "slice-7",
              "description": "Foo",
              "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": [
    {
      "id": "SDP1",
      "description": "Central Office 1 at location PE-A",
      "node-id": "PE-A",
      "sdp-ip-address": ["2001:db8:0:1::1"],
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:_service-vlan-match",
            "value": ["100"],
            "target-connection-group-id": "matrix1"
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "AC-SDP1",
            "description": "Device 1 to PE-A",
            "ac-node-id": "PE-A",
            "ac-tp-id": "GigabitEthernet1/0/0/0",
            "ac-ipv6-address": "2001:db8:0:1::1",
            "ac-ipv6-prefix-length": 64,
            "ac-tags": {
              "ac-tags": [
                {
                  "tag-type": "ietf-nss:vlan-id",
                  "value": ["100"]
                }
              ]
            },
            "incoming-qos-policy": {
              "qos-policy-name": "QosQoS-Gold",
              "rate-limits": {
                "cir": "1000000",
                "cbs": "1000",
                "pir": "5000000",
                "pbs": "1000"
              }
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "SDP2",
      "description": "Central Office 2 at location PE-B",
      "node-id": "PE-B",
      "sdp-ip-address": ["2001:db8:0:2::1"],
      "service-match-criteria": {
        "match-criterion": [
```

```
        {
          "index": 1,
          "match-type": "ietf-nss:_service-vlan-match",
          "value": ["100"],
          "target-connection-group-id": "matrix1"
        }
      ]
    },
    "attachment-circuits": {
      "attachment-circuit": [
        {
          "id": "AC-SDP2",
          "description": "Device 2 to PE-B",
          "ac-node-id": "PE-B",
          "ac-tp-id": "GigabitEthernet2/0/0/0",
          "ac-ipv6-address": "2001:db8:0:2::1",
          "ac-ipv6-prefix-length": 64,
          "ac-tags": {
            "ac-tags": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": ["100"]
              }
            ]
          },
          "incoming-qos-policy": {
            "qos-policy-name": "QoSs-Gold",
            "rate-limits": {
              "cir": "1000000",
              "cbs": "1000",
              "pir": "5000000",
              "pbs": "1000"
            }
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  },
  {
    "id": "SDP3",
    "description": "Remote Office 1 at location PE-C",
    "node-id": "PE-C",
    "sdp-ip-address": ["2001:db8:0:3::1"],
    "service-match-criteria": {
      "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-nss:_service-vlan-match",
          "value": ["100"],
          "target-connection-group-id": "matrix1"
        }
      ]
    },
    "attachment-circuits": {
```

```
      "attachment-circuit": [
        {
          "id": "AC-SDP3",
          "description": "Device 3 to PE-C",
          "ac-node-id": "PE-C",
          "ac-tp-id": "GigabitEthernet3/0/0/0",
          "ac-ipv6-address": "2001:db8:0:3::1",
          "ac-ipv6-prefix-length": 64,
          "ac-tags": {
            "ac-tags": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": ["100"]
              }
            ]
          },
          "incoming-qos-policy": {
            "qos-policy-name": "QosQoS-Gold",
            "rate-limits": {
              "cir": "1000000",
              "cbs": "1000",
              "pir": "5000000",
              "pbs": "1000"
            }
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  },
  {
    "id": "SDP4",
    "description": "Remote Office 2 at location PE-D",
    "node-id": "PE-D",
    "sdp-ip-address": ["2001:db8:0:4::1"],
    "service-match-criteria": {
      "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-nss:_service-vlan-match",
          "value": ["100"],
          "target-connection-group-id": "matrix1"
        }
      ]
    },
    "attachment-circuits": {
      "attachment-circuit": [
        {
          "id": "AC-SDP4",
          "description": "Device 4 to PE-D",
          "ac-node-id": "PE-A",
          "ac-tp-id": "GigabitEthernet4/0/0/0",
          "ac-ipv6-address": "2001:db8:0:4::1",
          "ac-ipv6-prefix-length": 64,
          "ac-tags": {
```

```
                    "ac-tags": [
                      {
                        "tag-type": "ietf-nss:vlan-id",
                        "value": ["100"]
                      }
                    ]
                  },
                  "incoming-qos-policy": {
                    "qos-policy-name": "QosQoS-Gold",
                    "rate-limits": {
                      "cir": "1000000",
                      "cbs": "1000",
                      "pir": "5000000",
                      "pbs": "1000"
                    }
                  },
                  "status": {
                  }
                }
              ]
            },
            "status": {
            }
          }
        ]
      },
      "connection-groups": {
        "connection-group": [
          {
            "id": "matrix1",
            "slo-sle-template": "low-BW-template",
            "connectivity-construct": [
              {
                "id": 1,
                "a2a-sdp": [
                  {
                    "sdp-id": "SDP1",
                    "slo-sle-template": "high-BW-template"
                  },
                  {
                    "sdp-id": "SDP2",
                    "slo-sle-template": "high-BW-template"
                  },
                  {
                    "sdp-id": "SDP3"
                  },
                  {
                    "sdp-id": "SDP4"
                  }
                ],
                "status": {
                }
              }
            ]
          }
        ]
      }
    }
```

```
        ]
      }
    }
  }
```

B.6. Example-6: SDP at CE, L3 A2A Slice Service

The following example describes a simplified service configuration of one IETF Network slice instances where the SDPs are located at the PE-facing ports on the CE:

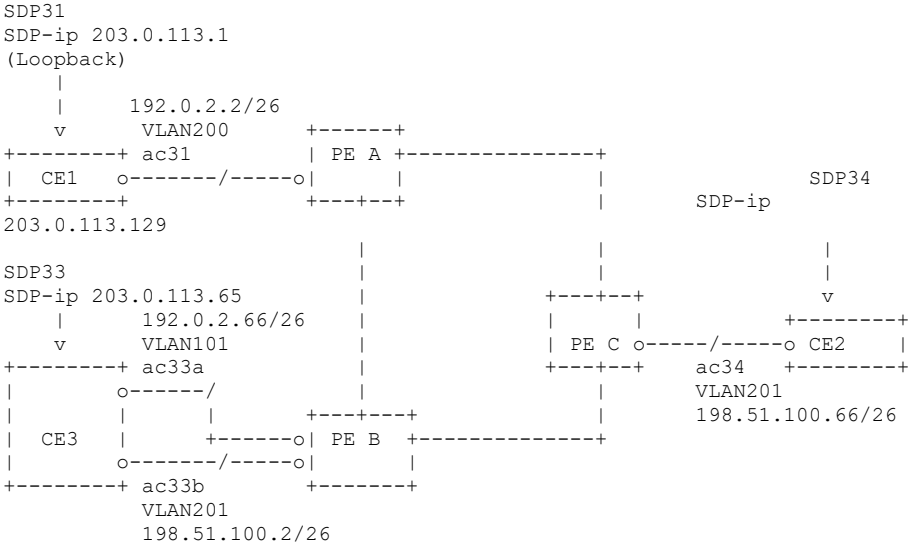
* IETF Network Slice 8 with SDP31 on CE Device1, SDP33 (with two ACs) on Device 3 and SDP34 on Device 4, with an A2A connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs.

* This example also introduces the optional attribute of "sdp-ip". In this example it could be a loopback on the device. How this "sdp-ip" is used by the NSC is out-of-scope here, but an example could be it is the management interface of the device. The SDP and AC details are from the perspective of the CE in this example. How the CE ACs are mapped to the PE ACs are up to the NSC implementation and out-of-scope in this example.

  SDP31 ac-id=ac31, node-id=Device1, interface: GigabitEthernet0 vlan 100

  SDP33 ac-id=ac33a, node-id=Device3, interface: GigabitEthernet0 vlan 101

  SDP33 ac-id=ac33b, node-id=Device3, interface: GigabitEthernet1 vlan 201
  SDP34 ac-id=ac34, node-id=Device4, interface: GigabitEthernet3 vlan 100



```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice8",
          "description": "slice-8",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          },
          "slo-sle-template": "low-latency-template",
          "status": {
          },
          "sdps": {
            "sdp": [
              {
                "id": "31",
                "node-id": "Device-1",
                "sdp-ip-address": ["203.0.113.1"],
                "service-match-criteria": {
                  "match-criterion": [
                    {
                      "index": 1,
                      "match-type": "ietf-nss:service-any-match",
                      "target-connection-group-id": "matrix1",
                      "target-connectivity-construct-id": 1
                    }
                  ]
                },
                "attachment-circuits": {
                  "attachment-circuit": [
                    {
                      "id": "ac31",
                      "description": "AC1 connected to PE-A",
                      "ac-node-id": "Device-1",
                      "ac-tp-id": "GigabitEthernet0",
                      "ac-ipv4-address": "192.0.2.2",
                      "ac-ipv4-prefix-length": 26,
                      "ac-tags": {
                        "ac-tags": [
```

```
            {
              "tag-type": "ietf-nss:vlan-id",
              "value": ["100"]
            }
          ]
        },
        "status": {
        }
      }
    ]
  },
  "status": {
  }
},
{
  "id": "33",
  "node-id": "Device-3",
  "sdp-ip-address": ["203.0.113.65"],
  "service-match-criteria": {
    "match-criterion": [
      {
        "index": 1,
        "match-type": "ietf-nss:service-any-match",
        "target-connection-group-id": "matrix1",
        "target-connectivity-construct-id": 1
      }
    ]
  },
  "attachment-circuits": {
    "attachment-circuit": [
      {
        "id": "ac33a",
        "description": "AC33a connected to PE-B",
        "ac-node-id": "Device-3",
        "ac-tp-id": "GigabitEthernet0",
        "ac-ipv4-address": "192.0.2.66",
        "ac-ipv4-prefix-length": 26,
        "ac-tags": {
          "ac-tags": [
            {
              "tag-type": "ietf-nss:vlan-id",
              "value": ["101"]
            }
          ]
        },
        "status": {
        }
      },
      {
        "id": "ac33b",
        "description": "AC33b connected to PE-B",
        "ac-node-id": "Device-3",
        "ac-tp-id": "GigabitEthernet1",
        "ac-ipv4-address": "198.51.100.2",
        "ac-ipv4-prefix-length": 26,
        "ac-tags": {
          "ac-tags": [
            {
```

```
                "tag-type": "ietf-nss:vlan-id",
                "value": ["201"]
              }
            ]
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  },
  {
    "id": "34",
    "node-id": "Device-4",
    "sdp-ip-address": ["203.0.113.129"],
    "service-match-criteria": {
      "match-criterion": [
        {
          "index": 1,
          "match-type": "ietf-nss:service-any-match",
          "target-connection-group-id": "matrix1",
          "target-connectivity-construct-id": 1
        }
      ]
    },
    "attachment-circuits": {
      "attachment-circuit": [
        {
          "id": "ac34",
          "description": "AC34 connected to PE-C",
          "ac-node-id": "Device-4",
          "ac-tp-id": "GigabitEthernet3",
          "ac-ipv4-address": "198.51.100.66",
          "ac-ipv4-prefix-length": 26,
          "ac-tags": {
            "ac-tags": [
              {
                "tag-type": "ietf-nss:vlan-id",
                "value": ["100"]
              }
            ]
          },
          "status": {
          }
        }
      ]
    },
    "status": {
    }
  }
 ]
},
"connection-groups": {
  "connection-group": [
    {
      "id": "matrix1",
```

```
                "connectivity-type": "ietf-vpn-common:any-to-any",
                "connectivity-construct": [
                  {
                    "id": 1,
                    "a2a-sdp": [
                      {
                        "sdp-id": "31"
                      },
                      {
                        "sdp-id": "33"
                      },
                      {
                        "sdp-id": "34"
                      }
                    ],
                    "status": {
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  }
}
```

B.7. Example-7: SDP at CE, L3 A2A Slice Service with Network Abstraction

The following example describes a simplified service configuration of one IETF Network slice instances where the SDPs are located at the PE-facing ports on the CE.

In this example it is assumed that the NSC already has circuit binding details between the CE and PE which were previously assigned (method is out-of-scope) or the NSC has mechanisms to determine this mapping. While the NSC capabilities are out-of-scope of this document, the NSC may use the CE device name, "sdp-id", "sdp-ip", "ac-id" or the "peer-sap-id" to complete this AC circuit binding.

We are introducing the "peer-sap-id" in this example, which in this case, is an operator provided identifier that the slice requester can use for the NSC to identify the service attachment point (saps) in an abstracted way. How the NSC uses the "peer-sap-id" is out of scope of this document, but a possible implementation would be that the NSC was previously provisioned with a "peer-sap-id" to PE device/interface/VLAN mapping table. Alternatively, the NSC can request this mapping from an external database.

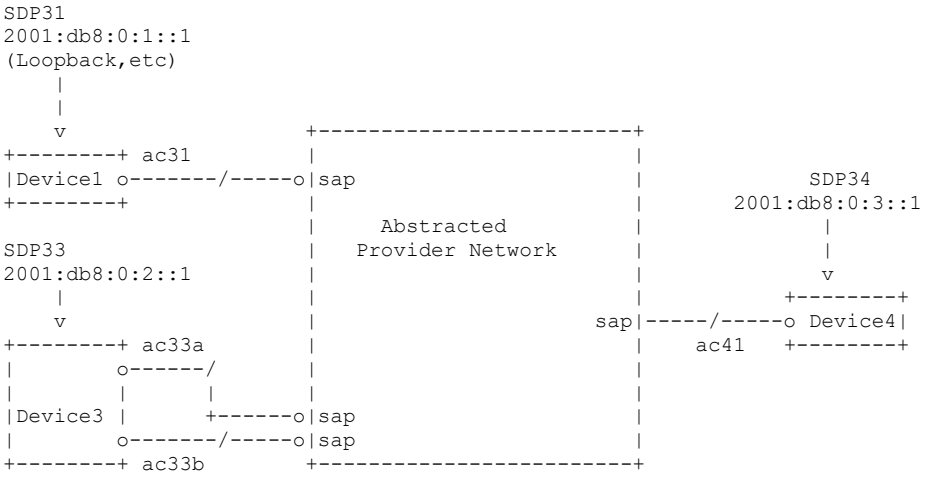
* IETF Network Slice 9 with SDP31 on CPE Device1, SDP33 (with two ACs) on Device 3 and SDP34 on Device 4, with an A2A connectivity type. This is a L3 slice service and using the uniform low-latency slo-sle-template policy between all SDPs.

  SDP31 ac-id=ac31, node-id=Device1, peer-sap-id= foo.com-circuitID-12345

SDP33 ac-id=ac33a, node-id=Device3, peer-sap-id=foo.com-circuitID-67890

SDP33 ac-id=ac33b, node-id=Device3, peer-sap-id=foo.com-circuitID-54321ABC

SDP34 ac-id=ac34, node-id=Device4, peer-sap-id=foo.com-circuitID-9876




```
{
  "data": {
    "ietf-network-slice-service:network-slice-services": {
      "slo-sle-templates": {
        "slo-sle-template": [
          {
            "id": "high-BW-template",
            "description": "take the highest BW forwarding path"
          },
          {
            "id": "low-latency-template",
            "description": "lowest possible latency forwarding behavior"
          }
        ]
      },
      "slice-service": [
        {
          "id": "slice-9",
          "description": "example slice7",
          "service-tags": {
            "tag-type": [
              {
                "tag-type": "ietf-nss:service-tag-service",
                "value": ["L3"]
              }
            ]
          },
```

```
"slo-sle-template": "low-latency-template",
"status": {
},
"sdps": {
  "sdp": [
    {
      "id": "31",
      "node-id": "Device-1",
      "sdp-ip-address": ["2001:db8:0:1::1"],
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:service-any-match",
            "target-connection-group-id": "matrix1"
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac31",
            "sdp-peering": {
              "peer-sap-id": "foo.com-circuitID-12345"
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "33",
      "node-id": "Device-3",
      "sdp-ip-address": ["2001:db8:0:2::1"],
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:service-any-match",
            "target-connection-group-id": "matrix1",
            "target-connectivity-construct-id": 1
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac33a",
            "sdp-peering": {
              "peer-sap-id": "foo.com-circuitID-67890"
            },
            "status": {
            }
          },
          {
```

```
            "id": "ac33b",
            "sdp-peering": {
              "peer-sap-id": "foo.com-circuitID-54321ABC"
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    },
    {
      "id": "34",
      "node-id": "Device-4",
      "sdp-ip-address": ["2001:db8:0:3::1"],
      "service-match-criteria": {
        "match-criterion": [
          {
            "index": 1,
            "match-type": "ietf-nss:_service-any-match",
            "target-connection-group-id": "matrix1"
          }
        ]
      },
      "attachment-circuits": {
        "attachment-circuit": [
          {
            "id": "ac34",
            "sdp-peering": {
              "peer-sap-id": "foo.com-circuitID-9876"
            },
            "status": {
            }
          }
        ]
      },
      "status": {
      }
    }
  ]
},
"connection-groups": {
  "connection-group": [
    {
      "id": "matrix1",
      "connectivity-type": "ietf-vpn-common:any-to-any",
      "connectivity-construct": [
        {
          "id": 1,
          "a2a-sdp": [
            {
              "sdp-id": "31"
            },
            {
              "sdp-id": "33"
            },
            {
```

```
                    "sdp-id": "34"
                  }
                ],
                "status": {
                }
              }
            ]
          }
        ]
      }
    }
  ]
}
}
```

Appendix C. Complete Model Tree Structure

Commenté [BMI95]: Add a note to the RFC editor to remove the appendix

```
module: ietf-network-slice-service
  +--rw network-slice-services
     +--rw slo-sle-templates
     |  +--rw slo-sle-template* [id]
     |     +--rw id               string
     |     +--rw description?     string
     |     +--rw template-ref?    leafref
     |     +--rw slo-policy
     |     |  +--rw metric-bound* [metric-type]
     |     |  |  +--rw metric-type          identityref
     |     |  |  +--rw metric-unit          string
     |     |  |  +--rw value-description?   string
     |     |  |  +--rw percentile-value?    percentile
     |     |  |  +--rw bound?               uint64
     |     |  +--rw availability?   identityref
     |     |  +--rw mtu?            uint16
     |     +--rw sle-policy
     |        +--rw security*              identityref
     |        +--rw isolation*             identityref
     |        +--rw max-occupancy-level?   uint8
     |        +--rw steering-constraints
     |           +--rw path-constraints
     |           +--rw service-function
     +--rw slice-service* [id]
        +--rw id                              string
        +--rw description?                    string
        +--rw service-tags
        |  +--rw tag-type* [tag-type]
        |     +--rw tag-type    identityref
        |     +--rw value*      string
        +--rw (slo-sle-policy)?

        |  +--:(standard)
        |  |  +--rw slo-sle-template?         leafref
        |  +--:(custom)
        |     +--rw service-slo-sle-policy
        |        +--rw description?   string
        |        +--rw slo-policy
        |        |  +--rw metric-bound* [metric-type]
        |        |  |  +--rw metric-type          identityref
```

```
|        |  |  +--rw metric-unit          string
|        |  |  +--rw value-description?   string
|        |  |  +--rw percentile-value?    percentile
|        |  |  +--rw bound?               uint64
|        |  +--rw availability?   identityref
|        |  +--rw mtu?            uint16
|        +--rw sle-policy
|           +--rw security*               identityref
|           +--rw isolation*              identityref
|           +--rw max-occupancy-level?    uint8
|           +--rw steering-constraints
|              +--rw path-constraints
|              +--rw service-function
+--rw compute-only?                   empty
+--rw status
|  +--rw admin-status
|  |  +--rw status?        identityref
|  |  +--rw last-change?   yang:date-and-time
|  +--ro oper-status
|     +--ro status?        identityref
|     +--ro last-change?   yang:date-and-time
+--rw sdps
|  +--rw sdp* [id]
|     +--rw id                          string
|     +--rw description?                string
|     +--rw location
|     |  +--rw altitude?    int64
|     |  +--rw latitude?    decimal64
|     |  +--rw longitude?   decimal64
|     +--rw node-id?                    string
|     +--rw sdp-ip-address*             inet:ip-address
|     +--rw tp-ref?                     leafref
|     +--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 incoming-qos-policy
|     |  +--rw qos-policy-name?   string
|     |  +--rw rate-limits
|     |     +--rw cir?   uint64
|     |     +--rw cbs?   uint64
|     |     +--rw eir?   uint64
|     |     +--rw ebs?   uint64
|     |     +--rw pir?   uint64
|     |     +--rw pbs?   uint64
|     +--rw outgoing-qos-policy
|     |  +--rw qos-policy-name?   string
|     |  +--rw rate-limits
|     |     +--rw cir?   uint64
|     |     +--rw cbs?   uint64
```

```
|      |     +--rw eir?   uint64
|      |     +--rw ebs?   uint64
|      |     +--rw pir?   uint64
|      |     +--rw pbs?   uint64
|      +--rw sdp-peering
|      |  +--rw peer-sap-id*   string
|      |  +--rw protocols
|      +--rw ac-svc-name*              string
|      +--rw attachment-circuits
|      |  +--rw attachment-circuit* [id]
|      |     +--rw id                        string
|      |     +--rw description?              string
|      |     +--rw ac-svc-name?              string
|      |     +--rw ac-node-id?               string
|      |     +--rw ac-tp-id?                 string
|      |     +--rw ac-ipv4-address?
|      |     |       inet:ipv4-address
|      |     +--rw ac-ipv4-prefix-length?   uint8
|      |     +--rw ac-ipv6-address?
|      |     |       inet:ipv6-address
|      |     +--rw ac-ipv6-prefix-length?   uint8
|      |     +--rw mtu?                      uint16
|      |     +--rw ac-tags
|      |     |  +--rw ac-tags* [tag-type]
|      |     |     +--rw tag-type    identityref
|      |     |     +--rw value*      string
|      |     +--rw incoming-qos-policy
|      |     |  +--rw qos-policy-name?   string
|      |     |  +--rw rate-limits
|      |     |     +--rw cir?   uint64
|      |     |     +--rw cbs?   uint64
|      |     |     +--rw eir?   uint64
|      |     |     +--rw ebs?   uint64
|      |     |     +--rw pir?   uint64
|      |     |     +--rw pbs?   uint64
|      |     +--rw outgoing-qos-policy
|      |     |  +--rw qos-policy-name?   string
|      |     |  +--rw rate-limits
|      |     |     +--rw cir?   uint64
|      |     |     +--rw cbs?   uint64
|      |     |     +--rw eir?   uint64
|      |     |     +--rw ebs?   uint64
|      |     |     +--rw pir?   uint64
|      |     |     +--rw pbs?   uint64
|      |     +--rw sdp-peering
|      |     |  +--rw peer-sap-id?   string
|      |     |  +--rw protocols
|      |     +--rw status
|      |        +--rw admin-status
|      |        |  +--rw status?        identityref
|      |        |  +--rw last-change?   yang:date-and-time
|      |        +--ro oper-status
|      |           +--ro status?        identityref
|      |           +--ro last-change?   yang:date-and-time
|      +--rw status
|      |  +--rw admin-status
|      |  |  +--rw status?        identityref
|      |  |  +--rw last-change?   yang:date-and-time
```

```
|     |  +--ro oper-status
|     |     +--ro status?        identityref
|     |     +--ro last-change?   yang:date-and-time
|     +--ro sdp-monitoring
|        +--ro incoming-bw-value?     uint64
|        +--ro incoming-bw-percent    decimal64
|        +--ro outgoing-bw-value?     uint64
|        +--ro outgoing-bw-percent    decimal64
+--rw connection-groups
|  +--rw connection-group* [id]
|     +--rw id                                  string
|     +--rw connectivity-type?
|     |       identityref
|     +--rw (slo-sle-policy)?
|     |  +--:(standard)
|     |  |  +--rw slo-sle-template?            leafref
|     |  +--:(custom)
|     |     +--rw service-slo-sle-policy
|     |        +--rw description?   string
|     |        +--rw slo-policy
|     |        |  +--rw metric-bound* [metric-type]
|     |        |  |  +--rw metric-type
|     |        |  |  |       identityref
|     |        |  |  +--rw metric-unit          string
|     |        |  |  +--rw value-description?   string
|     |        |  |  +--rw percentile-value?
|     |        |  |  |       percentile
|     |        |  |  +--rw bound?               uint64
|     |        |  +--rw availability?   identityref
|     |        |  +--rw mtu?            uint16
|     |        +--rw sle-policy
|     |           +--rw security*
|     |           |       identityref
|     |           +--rw isolation*
|     |           |       identityref
|     |           +--rw max-occupancy-level?    uint8
|     |           +--rw steering-constraints
|     |              +--rw path-constraints
|     |              +--rw service-function
|     +--rw service-slo-sle-policy-override?
|     |       identityref
|     +--rw connectivity-construct* [id]
|     |  +--rw id
|     |  |       uint32
|     |  +--rw (type)?
|     |  |  +--:(p2p)
|     |  |  |  +--rw p2p-sender-sdp?
|     |  |  |  |       -> ../../../../sdps/sdp/id
|     |  |  |  +--rw p2p-receiver-sdp?
|     |  |  |          -> ../../../../sdps/sdp/id
|     |  |  +--:(p2mp)
|     |  |  |  +--rw p2mp-sender-sdp?
|     |  |  |  |       -> ../../../../sdps/sdp/id
|     |  |  |  +--rw p2mp-receiver-sdp*
|     |  |  |          -> ../../../../sdps/sdp/id
|     |  |  +--:(a2a)
|     |  |     +--rw a2a-sdp* [sdp-id]
|     |  |        +--rw sdp-id
```


-> ../../../../../sdps/sdp/id
+--rw (slo-sle-policy)?
+--:(standard)
+--rw slo-sle-template? leafref
+--:(custom)
+--rw service-slo-sle-policy
+--rw description? string
+--rw slo-policy
+--rw metric-bound*
[metric-type]
+--rw metric-type
identityref
+--rw metric-unit
string
+--rw value-description?
string
+--rw percentile-value?
percentile
+--rw bound?
uint64
+--rw availability?
identityref
+--rw mtu?
uint16
+--rw sle-policy
+--rw security*
identityref
+--rw isolation*
identityref
+--rw max-occupancy-level?
uint8
+--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 description? string
+--rw slo-policy
+--rw metric-bound* [metric-type]
+--rw metric-type
identityref
+--rw metric-unit string
+--rw value-description? string
+--rw percentile-value?
percentile
+--rw bound? uint64
+--rw availability? identityref
+--rw mtu? uint16
+--rw sle-policy
+--rw security*
identityref
+--rw isolation*
identityref
+--rw max-occupancy-level? uint8
+--rw steering-constraints

```
|     |  |              +--rw path-constraints
|     |  |              +--rw service-function
|     |  +--rw service-slo-sle-policy-override?
|     |  |       identityref
|     |  +--rw status
|     |  |  +--rw admin-status
|     |  |  |  +--rw status?        identityref
|     |  |  |  +--rw last-change?   yang:date-and-time
|     |  |  +--ro oper-status
|     |  |     +--ro status?        identityref
|     |  |     +--ro last-change?   yang:date-and-time
|     |  +--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
|        +--ro two-way-max-delay?         uint32
|        +--ro two-way-delay-variation?   uint32
|        +--ro two-way-packet-loss?       decimal64
+--rw custom-topology
   +--rw network-ref?
           -> /nw:networks/network/network-id
```

Appendix D. Comparison with the Design Choice of 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.

Commenté [BMI96]: How this relates to AC/SAP?

* 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 Service connection requirements. However, the Network Slice SLOs and Network Slice SDPs 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.

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

Reza Rokui
Ciena
Email: rrokui@ciena.com
Tarek Saad
Cisco Systems, Inc
Email: tsaad@cisco.com

John Mullooly
Cisco Systems, Inc
Email: jmullool@cisco.com